



Co-funded by the European Commission



D5.8: OKKAM Entity Management: Challenges, Solutions and Experiences

Document Number	D5.8
Document Title	OKKAM Entity Management: Challenges, Solutions and Experiences
Version	1.0
Status	Final
Work Package	WP 5
Deliverable Type	Report
Contractual Date of Delivery	June 2010
Actual Date of Delivery	July 2010
Responsible Unit	UniTN
Contributors	ALL
Keyword List	
Dissemination level	PU

Abstract

This deliverable summarizes the challenges faced and lessons learned from the OKKAM project. It collects together general experiences as well as experiences from selected areas. Frankly reporting on the experiences made – not only the positive ones, this report is meant as a help for future players in the domain of entity identifier management and entity-centric technology as well as for actors involved in the setup of sustainable technology infrastructures as the OKKAM Entity Name System (ENS).

Document History

Version	Date	Status	Author (Company)	Description
0.1	June 2010	draft	Claudia Niederée	Definition of document structure
0.2	June 2010	draft	Claudia Niederée	First set of lessons learned for Section 3.2
0.3	June 2010	draft	Yannis Velegrakis, Zoltan Miklos, Claudia Niederée	Further lessons learned added to section 3.2; further text for section 3.2 added;
0.4	July 2010	draft	Claudia Niederée, Paolo Bouquet	First version of introduction (section 1) added, first version of section 2 added
0.5	July 2010	draft	John O’Flaherty, Daniele Cordioli, Adrian Mocan	First version of section 3.1 and 3.5, and section 3.4
0.6	July 2010	draft	George Giannakopoulos, Paolo Bouquet	First version of section 3.3 added, Revision of section 2 added
0.7	July 2010	draft	Adrian Mocan	Revision of section 3.4 added;
1.0	July 2010		Paolo Bouquet, Claudia Niederée	Final revision

Contributors

- Paolo Bouquet (UNITN)
- Daniele Cordioli (EXPERT SYSTEM)
- John O’Flaherty (MAC)
- George Giannakopoulos (UNITN)
- Zoltan Miklos (EPFL)
- Adrian Mocan (SAP)
- Claudia Niederée (L3S)
- Yannis Velegrakis (UNITN)

Executive Summary

This report summarizes the experiences made and lessons learned in the OKKAM project in the area of identifier management and entity-centric technology. It is a summary report on the results from the OKKAM project. However, the report does not focus on the research and technology results that have been achieved. It concentrates on extracting and summarizing those aspects of the OKKAM results that might be of special interest for other parties involved in similar endeavours in the context of entity-centric technology. Thus, the report focuses on challenges of a more general nature that have been identified and the lessons learned from them and the solutions chosen in OKKAM in this context.

The report is split into four parts: After an Introduction it summarizes the general experience in the area of conceptual and social concerns from the OKKAM project. In the third part selected technology issues are considered in more detail. For each of them identified challenges, OKKAM solutions as well as lessons learned are discussed. Finally there is a short concluding part.

Table of Contents

1	INTRODUCTION.....	6
1.1	PURPOSE OF THIS DOCUMENT.....	6
1.2	STRUCTURE OF THE DOCUMENT.....	6
2	GENERAL EXPERIENCES.....	8
3	EXPERIENCES IN SELECTED AREAS	12
3.1	SETTING UP AND POPULATING THE ENTITY REPOSITORY	12
3.1.1	<i>Introduction and Context</i>	12
3.1.2	<i>Identified Challenges</i>	13
3.1.3	<i>OKKAM Solutions</i>	14
3.1.4	<i>Lessons Learned</i>	14
3.2	ENTITY SEARCH AND MATCHING.....	15
3.2.1	<i>Introduction and Context</i>	15
3.2.2	<i>Identified Challenges</i>	15
3.2.3	<i>OKKAM Solutions</i>	16
3.2.4	<i>Lessons Learned</i>	17
3.3	MANAGING THE ENTITY LIFECYCLE	19
3.3.1	<i>Introduction and Context</i>	19
3.3.2	<i>Identified Challenges</i>	19
3.3.3	<i>OKKAM Solutions</i>	20
3.3.4	<i>Lessons Learned</i>	22
3.4	BUILDING ENTITY-CENTRIC TOOLS AND APPLICATIONS	23
3.4.1	<i>Introduction and Context</i>	23
3.4.2	<i>Identified Challenges</i>	24
3.4.3	<i>OKKAM Solutions</i>	25
3.4.4	<i>Lessons Learned</i>	27
3.5	OPERATING ENTITY IDENTIFIER MANAGEMENT AS A GLOBAL PUBLIC INFRASTRUCTURE.....	28
3.5.1	<i>Introduction and Context</i>	28
3.5.2	<i>Identified Challenges</i>	29
3.5.3	<i>OKKAM Solutions</i>	29
3.5.4	<i>Lessons Learned</i>	32
4	CONCLUSIONS	34
5	REFERENCES.....	35

1 Introduction

1.1 Purpose of this document

Creating a stable, flexible and sustainable solution for entity identifier management in the Web, as it has been defined as the purpose of the OKKAM project, is a challenging endeavour. Putting this plan into practise – as it has been done in the 30 month of the OKKAM project – did not only result in the OKKAM ENS, which is now operational, but also created a variety of insights for the team involved in the OKKAM project.

The purpose of this document is to collect the challenges identified and the lessons learned during the OKKAM project, which we expect to be useful for other Research and Development teams working in the area of entity identifier management and related areas.

The OKKAM team has decided

- to keep this document concise,
- to avoid technical details that can be found in the other deliverables and in the scientific publications that resulted from the RTD work in the OKKAM project, and
- to frankly report on lessons learned and experiences made, in order to maximize the usefulness for the reader

The document is meant for research teams as well as for developers that work in the area of entity-centric technology, such as entity identifier management, entity search, entity-centric search etc. We also expect it to be useful – at least in parts - for research teams in related areas that aim to set up a large integrated, operational and sustainable system as a result of an RTD project.

1.2 Structure of the Document

The document is split into four sections. After shortly introducing the document content in this section, section 2 and 3 form the main part of the document, discussing the challenges and lessons learned, while section 4 closes the document with some conclusions.

In more detail, Section 2 of the document summarizes the general experiences made during the project in the area of entity identifier management. It focuses on conceptual and social concerns, leaving more scientific and technical issues for discussion in section 3.

For section 3, six central areas/issues in dealing with entity (identifier) management have been picked and are discussed in more detail, also including technical and scientific aspects. They are listed here together with a short reason for their selection and a pointer to the respective section:

- Setting up and populating the entity repository (see section 3.1): selected because of the importance of a well-populated entity repository for the acceptance and adoption of the entity-centric approach suggested by OKKAM.
- Entity Search and Matching (see section 3.2): selected because of the central role of the entity search functionality for the overall OKKAM entity identifier management service (Entity Name System, ENS)
- Managing the entity lifecycle (see section 3.3): selected because of the role of sound foundations and adequate lifecycle management for maintainability and the secure long-term operation of the OKKAM ENS.

- Building Entity-centric Tools and Applications (see section 3.4): Selected since building adequate useable and useful tools on top of the ENS and any other entity management solution is crucial to make it attractive for adoption;
- Operating Entity Identifier Management as a global public Infrastructure (see section 3.5): selected because of the expected general interest for the lessons learned in this area.

For each of the six selected areas:

- We start with a short introduction giving the reader some context.
- We continue with a list of challenges that have been identified while working on the issue during the OKKAM project. This includes challenges identified in the initial design phase, and challenges that have been only identified while gathering experiences with the first prototypes of the OKKAM ENS and the applications build on top of it and while gathering experiences from interactions with other communities.
- We summarize the solutions that have been developed in the OKKAM project to deal with the identified challenges
- We complete each of the sections with a list of lessons learned that have been collected from the OKKAM partners.

2 General Experiences

The general objective of the OKKAM project was (and still is) very ambitious, as creating a global naming system for the Web is a challenge which is not only tough from the scientific and technical standpoint, but is also likely to raise some conceptual and social concerns. In this section, we are going to focus more on this second type of issues, as the first type is addressed in detail in the remaining sections of the document.

As a starting point, we'd like to discuss two very broad issues which accompanied the entire development of an entity management system in OKKAM.

First, we found that it is not easy for people – even for people with a technical background or for researchers – to fully grasp the difference between an Entity Name System for the Web (like the one we built in the OKKAM project) and a public, global knowledge base on entities (like e.g. Dbpedia or Freebase). In our experience, this was one of the key obstacles in the acceptance of what we did, as people don't seem to like the idea that someone may collect information about them (or about other relevant entities) independently from their control. Of course, this is not the goal of the ENS (at least as it was designed by the OKKAM consortium), but the lack of a full understanding of the difference created a distorted perception that we found very difficult to address. This perception is sometimes combined with the feeling that this is going to be a modern version of the “big brother” on the web, and of course this is not what most people want.

Second, we mention another issue which was raised several times and often led to partial misunderstandings in external communication: the relationship between **global identifiers** for entities and **identification as authentication**. OKKAM IDs are public by definition and they are meant for the greatest publicity and distribution; a username or account name is not, as it is part of a system for protecting information, not for interlinking it. Therefore, the fact that they uniquely identify an entity on the web does not mean that they can or should be used as a way for uniquely authenticate a user on a system. This question was often asked in connection with OpenID: can people use OKKAM as an OpenID server? It is by now quite clear that for many users identification and authentication are not clearly distinguished. The OKKAM consortium took special care in not creating explicit links with services like OpenID, but the ambiguity still remains.

When the limits and functions of the ENS are clarified, we found that most people do understand that in principle a single naming system for the Web would make many operations much simpler and faster. This is true not only for researchers, but also for business people, as the advantage of having a single “key” to fetch and merge information about a single entity is quite obvious in many practical situations.

A very concrete example of why the current situation is far from ideal is web data mash-up through Sig.ma (<http://sig.ma/>):

- when Sig.ma is used to create mash-ups by **keyword**, the result has very good recall, but very poor precision (lots and lots of irrelevant sources are loaded because of homonymy or string similarity assumptions). The result is that for entities with a high potential for ambiguity the mash-up is really messy;
- when Sig.ma is used to create a mash-up by (linked data) **URIs**, the result has a very poor recall, but 100% precision (typically, only one source is retrieved, which is precisely the source where the URI was firstly introduced and used), as very few links between different URIs for the same resource are created and made available.

Quite obviously, having a single ID for the same resource everywhere would solve the problem of low precision with keywords, and at the same time the problem of poor recall with local URIs. However, even though the problem is evident to everyone, we must honestly recognize that there is not a general agreement on a single method for addressing this highly unsatisfactory situation. The solutions proposed by the Semantic Web community so far are mainly two (the third would be sticking to keyword search, but in this case the added value with respect to standard search engines would be doubtful):

1. The **Linked Data** approach: ask content providers to store identity statements (the well-known `owl:sameAs` triples) together with their data to make explicit the fact that the URI they use for an entity (e.g. for Venice) has the same reference as the URI used for Venice in GeoNames or in Dbpedia. This way, systems like Sig.ma could use these identity statements to improve recall.
2. The **OKKAM** approach: making possible for content providers to ground their data to a common (shared) identifier (the OKKAM id) for the same entity, so that systems like Sig.ma can find any data source about an entity with a simple index look-up operation.

Again, most people (especially from the business community) seem to understand why in principle the second solution would be easier and more effective. However, we had to face a number of recurrent arguments which may prevent its general adoption. Here we report the most popular ones as a list of attention points for anybody else willing to work on entity identifiers management on the Web:

1. dependency on a **centralized** third-party service: in the Web research community there seems to be a general concern about any dependency on centralized solutions for the Web. This for two different reasons: on the one hand, it might introduce a single point of failure; on the other hand, it seems to infringe the general vision of the Web as a totally decentralized system where anyone can publish content and data in a fractal way. The OKKAM's Entity Name System is perceived as a source of centralization, so people are reluctant to adopt it.
2. **Trust and Provenance**: another category of concerns has to do with the idea that a URI for an entity is not just a string, but somehow encodes some notion of ownership and provenance, generally based on the DNS authority. In other words, a URI like <http://sws.geonames.org/3164603/> for Venice is not just the identifier for the Italian city in a dataset (GeoNames), but somehow brings some level of trust as it has been published within the domain name of GeoNames, where only authorized people can deal with data. Based on this, for example, users might decide to trust what is said about <http://sws.geonames.org/3164603/> more than what is said about <http://dbpedia.org/resource/Venice>, based on some ranking of trust between GeoNames and Dbpedia. Using everywhere the OKKAM id for Venice would inhibit this kind of mechanism, and again people seem to be reluctant.
3. **Dereferenciation**: one of the expected advantages for using HTTP URIs as identifiers for entities on the Web is that users and application can resolve the address and get information about the entity itself. This is one of the key principles of Linked Data. Of course, if a user clicks on the two URIs for Venice we just used above as an example, they get different information: the GeoNames triples in the first case, the Dbpedia triples in the second. Even though OKKAM ids are, indeed, HTTP URIs, using them in different datasets would make this mechanism of multiple dereferenciation impossible, as Venice would be identified

through the same URI (its OKKAM id) in all datasets, making just impossible to know what are the different triple sets about Venice.

4. **Ownership:** for entities that have a commercial value (e.g. products), there is also another concern, namely that users would find very heterogeneous information about a product attached to the same identifier (the OKKAM id), in a mix of official and unofficial information which would make it difficult for users to tell it apart. Why FIAT, for example, might want to attach its official information about the new FIAT500 to the same identifier that competitors (or unhappy customers) would use for complaining about the same car?

At different degrees, all these arguments touch real problems, and the OKKAM Consortium has tried to address them in various ways. This led to several small, but significant, shifts in the way the OKKAM message was delivered through time in external communication. In particular:

- **Role of OKKAM identifiers:** in the OKKAM proposal and in the first phase of the project, the underlying assumption was that OKKAM ids should become “*the*” single identifier for an entity on the web. In other words, the idea was that OKKAM ids should replace any other pre-existing URI. This idea proved not only to be unfeasible, but conceptually wrong. Unfeasible, because people are very unlikely to throw away their ID schemes to replace them with OKKAM ids. Conceptually wrong, as the role of local URIs (like the Dbpedia URIs) and the role of OKKAM ids are different: the first ones are suitable for navigating across graphs using RDF browsers; the second are suitable for fixing reference. This led to a vision of the ENS as a “switchboard”, namely a service which may enable content providers to make their content accessible by mapping their local URIs to an OKKAM id. Technically, this was implemented as a collection of services for “alternativeIDs” in the profile of an OKKAM entity, moving away from the idea that the ENS should be thought of as an authority for standard identifiers.
- **Identity and reference:** at the beginning of the project, we conceived of OKKAM ids as a means for creating a single identity for an entity. However, this is not exactly what an entity naming system should really do, as an identity is a viewpoint on an entity and as such different identities require different identifiers. In the conceptual model which was developed in Y2 (and presented at ASWC2009), the major conceptual step was to present OKKAM ids as tools for fixing reference, and reference as the “intersection” across different identities attached to the same object (the names thing, and not what is said about it). This idea proved to be very challenging also from a philosophical standpoint, but it seems that it is much more palatable for content providers, as they do not see OKKAM ids as potential competitors for accessing data anymore, but rather as a potential ally. This is the reason why, for example, the triples from OKKAM profiles are not used on the data mash-ups produced by Sig.ma: an OKKAM profile is for recognizing an entity, not for describing an entity.
- **Intranet vs. Internet:** it was an initial assumption that the success of OKKAM would be on the side of user generated content on the Web. This led, for example, to investing a lot of effort in the creation of a large number of so-called ENS-empowered tools, namely standard tools for data and content production (like Protégé for OWL or MS Word for text processing), based on the assumption that single users would have started to create their own data and content by including OKKAM ids in it. However, this bottom-up approach seems to be unfeasible, as the only trigger for single users to do it would be the possibility of connecting their OKKAMized content with other existing OKKAMized content, which currently is not available in large amount on the web. Therefore, the OKKAM consortium

decided to devote more effort in smaller networks (e.g. institutional portals, digital libraries, Intranets, etc.) where the positive effects of global naming across independent and heterogeneous datasets and systems is more tangible. As an example, we can cite the Tax Project with Trentino Riscossioni, where the OKKAMization of about 60 independent databases had the effect of making the job of tax inspectors much faster and easier than before. In general, it seems that the availability of large and popular datasets with OKKAM ids may be the trigger for individual users to OKKAMize their content, as they have a perception of an immediate advantage.

To sum up, we can make two general observations. On the one hand, some of the concerns above are stronger in the (Semantic) Web community than in the business community. The centralization of the ENS, the use of DNS as an authority for trust, the use of entity identifiers as a way for navigating across graphs do not seem to be real problems, provided that OKKAM can offer practical solutions to business needs, such as flexible data integration, semantic search, smart authoring. As an example, we can rely here the experience with some of the OKKAM partners (SAP, Elsevier, ANSA), but also some of the pilot projects which we have developed in the final part of the project (the Tax Agency in Trentino, the City of Manor in Texas, the AGRIS digital library with FAO, the Innovation Management portal of the Flemish Government, the portal of the University of Trento).

On the other hand, with the progress of the project, we witnessed an increasing acceptance also in the research community. The fast development of initiatives like Open Government Data and Linked Data have shown that there is a need for different solutions, including the alignment of identifiers in ways which can be different from the use of `owl:sameAs` statements. As an example for this, we can mention the contacts we had with projects like LarKC, NoTube, SmartProducts, but also the participation of OKKAM representatives in the W3C Incubator and then Working Group on RDF2RDB, where valuable use cases for OKKAM have been included in the official documents.

In the next sections, we discuss more detailed and technical aspects of our experience, in the hope that they might help other people and teams that may undertake the enterprise of developing an entity management system for the Web.

3 Experiences in Selected Areas

3.1 Setting up and Populating the Entity Repository

Every person working in network-based systems, among them the Semantic Web, is aware of the problem of identifying and referring entities. This issue derives from the difficulty of ensuring that the same entity is consistently assigned the same name (identifier) in all the local nodes of the system, across different data structures, formats, and applications. This is recognized as one of the most serious obstacles towards the creation of a global, integrated information space at the top of many decentralized systems.

3.1.1 Introduction and Context

The overall goal of the OKKAM project was to enable and bootstrap the Web of Entities, and one of the most important steps was to create the initial population of the repository. It is clear that in order to identify an entity in a unique way, we need to recognize it on the text, and afterwards compare the candidates with the entities inside the OKKAM repository. The creation of a good repository and its maintenance was a fundamental step inside the Okkamization process. In this scenario, we have selected four main areas that we have investigated in our work.

Sources identification and Import from existing repository of entities

This first area was very important in order to create a substantial base population for the OKKAM repository. A priority activity was to identify good sources and convert them into a common format; these sources could have been information on the web (like web pages, blogs, forum etc...), entities collected in local repositories, information stored in personal computers and so on.

Identification and selection of data collections that can be used as sources for the initial population of the OKKAM repository was another important task that could be represented by the following sub-activities:

- Sources identification for good entity coverage without overlapping
- Entities identification from relevant existing entity
- Definition of general info about this entity
- Comprehension and recognition not only of entities, but also of attributes characterizing them

Definition of Basic OKKAMizers Set.

Definition of the set of OKKAMizers to be built based on the requirements coming from the previous topics.

Creation of initial population

After the previous stages, the natural consequence was to plan and perform a bootstrapping process using the developed OKKAMizers and to import components in order to create a significant population of entities;

Creation and integration of OKKAMizer services

The OKKAM project will develop a set of OKKAMizers for frequently used document formats as well as for central types of entities that are important for the OKKAM applications, such as persons,

organizations and products. The list of OKKAMizers that we have built during the project was decided according to the requirement analysis. OKKAMization manages various popular information and knowledge representation formats such as MS Word, RDF, PDF, HTML, Relational databases, etc.

3.1.2 Identified Challenges

In this context, the major challenge was to develop methods and techniques able to satisfy the main area of work. This means to face a lot of challenges involving not only the methodological approach but also different kind of technologies to be used.

Analyzing and searching ideal sources to enrich OKKAM repository

Semantic web contains different kind of information in different formats, not all this information are relevant for our goals. The sources we searched were sources containing a good number of entities with related information qualifying them. The kind of entities to consider in our analysis should not be underevaluated: we need entities with a spread distribution on the web, entities that we are able to find in different kind of documents.

Create the infrastructure

After the sources identification, it was fundamental to design an infrastructure able to guarantee the initial and continuative populations of the repository. This means that what we need is an architecture analyzing different kind of sources; these might be web pages, forums, blogs, document repositories, etc. Where required these content collections are crawled, harvested, and stored in a repository.

The main processing step is the parsing and analysis of the content object for entity extraction, the disambiguation of the entities via the entity repository and the entity creation, if not yet created.

Increasing the amount of entities in the OKKAM repository

Once created the first entities population, it became relevant to increase this repository. In order to do this we identify the following sub-activities:

- To evaluate if there are sources with relevant entities to use
- To verify if the candidate entities are already in the repository
- To check new kind of entities to insert
- To map different entities, coming from different sources, in the same unique entity

Searching for new methods and algorithms to extract entities

The sources used to increase the number of entities could be structured or not structured. Probably if we are looking structured data we could find different records with information related to the entity, while we need a different approach if we are working with unstructured data: for each entity we have to recognize their distinctive features from plain text, using semantic methods and innovative algorithms.

From the perspective of **integrating** the various components, validating them and committing them into the operational service, the main challenges were:

1. Defining and agreeing a standard approach to test, validate and benchmark the ENS on each incremental upgrade of a component, without affecting the contents of the Entity Repository.

2. Populating the Entity Repository while incrementally developing and upgrading the overall system and physical storage was challenging to ensure forward and backward compatibility of the Entity records, deciding if an entity is new or not, if a merge or split is required and avoiding duplication.
3. Finding and eliminating the duplicate Entities became very time-consuming when the number of entities got into the millions.

3.1.3 OKKAM Solutions

Approaching this challenge, and implementing a solutions for each of them allow us to maintain a pool of interesting entities, unambiguously identified and associated with a minimum of information. In this way OKKAM will not become “another entities repository” but a very strategic access point to identify entities. Let’s imagine to have a great number of entities with different attributes linking directly the same entity in other repositories: it’s not far from becoming reality.

To test, validate and benchmark the ENS on each incremental upgrade, a standard set of typical entities were agreed (the Golden Set). This was used in conjunction with a strong procedure on each test bed, to test, validate and benchmark the ENS on each incremental upgrade of a component, even before it was committed to the operational service. The use of the www.okkamdev.org gForge distributed development platform proved to be very useful in this.

Populating the Entity Repository while incrementally developing and upgrading the overall system and storage architecture was addressed by defining and settling on the OKKAM Entity Model early in the project (Entity Model V3 as used in ENS V1)¹.

Deciding if an entity is new or not, if a merge or split is required and avoiding duplication proved to be the core research and innovation of the project to generate a-priori unique global entity identifiers. A set of tools was developed to assist with this², but as the repository grew, the time to undertake these tasks also increased. So in the ENS V3 the storage was split into integrated online (for fast service response) and offline (for heavy maintenance and cleansing processes)³.

3.1.4 Lessons Learned

- By using the OKKAM ENS, each entity is associated with a unique identifier and a profile, containing only as much information as needed to recognize the entity. A very important aspect of OKKAM in this context is represented by the ability to identify the entities, the related information in unstructured data; to create an infrastructure that the web citizens can use to recognize and create new entities; to evaluate/ increase the entities qualifier; to allow the enrichment of unstructured data with OKKAMid.
- Populating a repository while developing and evolving the system that is creating and maintaining it, is not easy! However with strong management of a formal release process and planning it can be successfully achieved, as was done in OKKAM.

¹ See D5.2-Integration Prototype “OKKAM Infrastructure V1”, Nov08

² Particularly the ENS Administration Console, see D5.6 (Integration Prototype OKKAM Infrastructure V3, July 2010

³ see D5.6 (Integration Prototype OKKAM Infrastructure V3, July 2010

3.2 Entity Search and Matching

3.2.1 Introduction and Context

The entity search and matching functionality is a central functionality in identifier management for entities as it is provided by the OKKAM ENS. In the core, entity search performs the following service: Given a description of an entity (an “entity request”), it returns an identifier for this entity and (if existent) alternative identifiers. The idea here is that this functionality enables the re-use of existing entity identifiers (returned by the system) instead of inventing (generating) own identifiers for already known and described entities. For this purpose, discriminative descriptions and identifiers of entities already known to the system are managed by the system in a repository.

For performing entity search, entity matching is required, i.e. comparing different entity descriptions or profiles with each other and deciding, which of them – most probably - describe the same entity. Conceptually, entity matching compares the description of the entity given by the ENS user with the entity profiles that exist in the entity repository of the entity identifier management system, in our case the OKKAM Entity Repository.

3.2.2 Identified Challenges

For entity matching - also known as de-duplication, entity linkage, entity resolution, and duplicate detection - a rich body of research and methods exist, which are a good starting point for implementing the entity matching functionality for entity search. However, due to the variety in the posed entity requests and the size and heterogeneity of the entity repository the design and implementation of the entity search functionality imposes a set of additional challenges. The most important of these challenges are summarized below:

- Designed as a global, general service the ENS has to face a wide variety of queries coming from different sources (different applications + human users), addressing different types of entities, being in different formats (from simple keyword to structured queries) and conveying different levels of knowledge about the entity (e.g. only the name, name + properties, name + properties + context information). This imposes challenges for the entity matching (and for the entity ranking as described below) for finding the best matching entity in the repository in all the cases.
- During the project it has shown that the query load is different from the one envisioned in the beginning of the project and that it also changes if new entity-centric applications are considered as users of the ENS. This raises the challenge to find a flexible solution for entity matching that can be adapted to changing requirements. Furthermore, adequate measures are required for evaluating the quality of the entity search functionality implemented.
- The repository of entities, in which entity search has to operate in, is not homogeneous. It contains entities of different types (e.g. persons, locations, organization). Furthermore, it is impossible to impose uniform schemata even within single entity types, in order not to hinder adoption in domains with different conceptualizations of or perspectives on entities. This clearly imposes challenges to entity search, since it has to operate in a very heterogeneous environment.

- The entity repository is clearly more useful, if most of the entities a user is looking for are already in the repository. Thus the entity repository needs a considerable ground population (see also section 3.1) and is furthermore growing with use. It is thus to be expected that entity search has to operate in settings with at least 10s of Millions rather 100s of Millions of entities or even more. The combination of the repository characteristics (size and heterogeneity) with the request characteristics above imposes major challenges for the entity search functionality, which has to carefully balance efficient processing, not missing entities that already exist in the repository and really finding the searched entity with high precision in the described context.
- Entity search is not restricted to one entity at a time scenarios. In some cases it is necessary to perform entity resolution/matching as a bulk operation, where a large set of entity descriptions are given at a time for finding the matching entities in the repository. This raises challenges such as how to deal with cases of ambiguity, in which cases and under which conditions new entities are inserted into the repository, and of efficient processing for very large data sets to be processed.
- An additional challenge that came up during the project – mainly from the interaction with possible application cases – is the need to deal with alternative already existing identifiers for an entity. Here the challenge is to serve as a type of translator or mediator between different existing identifiers, e.g. from different identifier system as an additional service of entity search.
- The size of the entity repository as well as the type of requests makes it in many cases impossible to just return the one entity, the user of the ENS was thinking about (e.g. in case of underspecified queries and ambiguities). For this case it is important to create a good ranking function that is able to push the most probable matching candidates in a high position of the returned ordered entity list – hopefully returning the desired entity in the top three or five, if not at the first rank.

3.2.3 OKKAM Solutions

For addressing the above challenges an entity matching framework has been developed in the OKKAM project and adapted during the project to comply with newly identified challenges. The solutions in this area are described in detail in the deliverables D6.1, D3.1 and in various publications created as results of the work in the OKKAM project. Here we just give a short summary of the solution developed in the area of entity search and matching.

Matching is implemented as a two phase process, where in a first phase a promising set of matching candidates is identified from the large entity repository relying on extended IR technology. In the second phase the restricted candidate list is analysed in more detail for identifying the most promising matching candidates for the considered entity requests and as a result a ranked list of entities is returned. The matching functionality has been implemented as a matching framework, which can host an extensible set of different matching modules. For request processing it is possible to select a matching module (e.g. based on the entity type or the request type) and is even possible to use more than one matching module and to combine their results. Through its easy extensibility with further matching modules the matching framework enables the required adaptability of the entity search functionality as discussed above. For the entity requests a special request language has been developed that is tailored to this purpose. For dealing with the wide variety of possible requests, which the ENS has to deal with, an deep analysis of the cases (e.g. under-specification and over-specification) has been performed and iterative approach has been taken for the development

of matching functionality, which tuned the initially developed functionality based on the results of systematic testing and an in depth analysis of the critical cases. Furthermore, much effort has been invested into the ranking functionality, for ensuring that the “right” entity is ranked into a high position in the result list. This included for example the consideration of attributes that are “important” for discriminating entities.

For dealing with matching as a bulk process a special component has been implemented that can deal efficiently with finding entity identifiers for large sets of entity descriptions.

3.2.4 Lessons Learned

The implementation and evaluation of the ENS and the matching functionality, which went through several iterations, was a complex and valuable process that brought new insights for the requirements and design of an ENS and led to some lessons learned. The most important lessons are listed here as a hopefully useful input to teams confronted with similar tasks:

- A system such as ENS should expect entities that are very heterogeneous, which imposes major challenges to the matching functionality. In addition to schema matching issues that appear mainly due to the absence of uniform schema information, it is necessary to deal with **under-specification** and with what we call **over-specification** in a systematic way. Under-specification occurs if the user (which can be an application or a human user) does not give enough information about the entity that we need to disambiguate (e.g. entity extraction just delivering the entity name). This gets more challenging in a global context. Over-specification results from the discrepancy between what is the user's knowledge about an entity and what the system knows about the entity, i.e., the user may state characteristics of this entity that are not known to the system, but the system should still be able to retrieve the corresponding entity.
- **Information Retrieval technology**, such as Lucene, can become a first step in system that aim addressing entity search and matching, even for problems that have a clear flavour of more structured search, such as entity identifier search in the ENS. We experienced this by inspecting the entity search results that are achieved by the storage layer of our matching framework.
- Even in a heterogeneous setting, where a close schema binding is missing, it is worth to take a close look on what we call **"important" attributes**. This is a mixture of estimating the selectivity of an attribute and of considering the attributes users prefer for describing an entity of a specific type (e.g., ``name" for people). Several experiments in the context of the ENS have shown that using the knowledge about the "importance" of attributes in the ranking and matching of the results can considerably improve the effectiveness.
- Designing a **ranking function** for entity profiles, which is an important part of the entity search functionality, is a challenging task, because of the heterogeneous nature of the profiles. We had to reconsider several times, what kind of ranking function we apply. Attribute importance, as mentioned above, proofed to be very important. Our initial strategy of not considering term frequency for ranking turned out to be problematic. It looked plausible, that the frequency of a term in a free text field of the profile should not influence the ranking. However, for some type of entities this strategy resulted in sub-optimal results. We found that the most effective way to define entity ranking is to define a specific ranking for each different entity type.

- Aspects such as schema **heterogeneity and multilingualism** should be considered more in depth early in the project. They have been underestimated in the first phase of the project and have proven to be difficult to "add" in a later phase after other design decisions have been made and implemented. In general, handling attribute name mappings, as they are used for dealing with schema heterogeneity in the current solution, is a challenging task in the targeted settings. The system would also benefit from a mechanism for automatically detecting and adapting mappings, e.g. according to user feedback, existing data, etc.
- **Regular and systematic evaluation** of the performance of the system in terms of quality and speed is crucial for the effective improvement of the entity matching and search functionality. This also includes the collection and analysis of critical cases, which gives important insights for the improvement and fine-tuning of the system.
- The creation of an **early prototype** of the ENS and especially of the entity search and matching functionality, as it has been done in the OKKAM project, gives many additional insights for system improvements and enables a deeper understanding of the requirement and is, thus, recommended for the creation of this type of systems.
- **Uncertainty** may come at different forms at different places and for different uses. Uncertainty may exist at the attribute level reflecting the real world situation in which not all the characteristics of an entity are 100% sure, there may exist different opinions about whether an attribute is valid or not, or there may be different situations in which an attribute is valid or not. Uncertainty may exist also on the relationships between the different entities. One of the main issues in entity management is the entity identification, i.e., finding whether two entities represent the same real world object or not. However, even this question turns out to be hard, since in many situations it is not possible for someone to give a definite answer given the information that is available. There are even situations in which from one point of view two entities represent the same real world object, and from another point of view, they do not. Finally, uncertainty also exists in query answering. Since queries may be underspecified, it may be hard to determine a single definite answer to the query. Thus, an answer is a set of possible entities, each one coming with a certain degree of confidence. We have learned that for systems such as OKKAM, the matching techniques should take uncertainty fully into consideration (by considering it as a first class object). They should start from the assumption that data may be by default incomplete and the queries underspecified. They should also exploit the fact that some auxiliary information about duplication of entities may exist. As such the results returned should always be probabilistic with the probabilities based on this information.
- It is not clear how to **evaluate an entity matching systems**. Entity systems evaluation is fundamentally different from query engines. In query engines for structured data such as databases, an answer to a query is typically deterministic and fully specified by the conditions of the query. In an entity matching system in which queries may be underspecified, or the information stored in the entity system may be incomplete, it is not clear what a correct answer is. Only an expert user can verify whether an answer is correct or not. This reminisces the situations of the information retrieval systems. However, it is yet different since the data is not a document that can be analyzed and through context more information be retrieved. Instead, it is partially structured information. We have learned that in order to evaluate the efficiency and effectiveness of a system such as OKKAM, an advanced evaluation mechanism is needed. In particular, we had to create a large collection of user queries that reflect the kind of questions that the users pose on such systems in real

applications scenario. This means that if our system performs well for these queries, there is a high possibility that it will also perform well the majority of the practical situations. However, this collection is not systematic and we have learned that it is not enough. Poor performance for queries in this collection cannot actually identify the exact problem in which the matching technique underperforms. For that reason we need to have a tool that systematically generates data and respective queries. Each set of data and queries should be designed to test a specific matching situation. When the matching system fails in such a situation, it is easy to identify the problem and how to fix it. Finally, the tool should be capable of generating data of different sizes, allowing that way to test how well the system solutions scale up.

- Different matching solutions perform better to different situations. No single technique can be used globally. Furthermore, different techniques take into consideration different specifications and auxiliary information which means that for the same query different result sets may be returned, that are typically conflicting. **Combining** these results into one single unified answer set is a challenging task. In OKKAM we have learned that such a combination should be based on how well each matcher has performed in similar situations in the past for better results.

3.3 Managing the Entity Lifecycle

3.3.1 Introduction and Context

Within the OKKAM lifetime, we examined the special requirements of lifecycle management for entities in the context of an entity management system for the semantic web. We studied the requirements with respect to creating and modifying these entities, as well as to managing their evolution over time. We tackled the issues arising from the access control models needed for the management of a large, distributed repository of entities. Finally, we devised ways to improve system performance, based on the analysis of system usage.

3.3.2 Identified Challenges

The main challenges faced in the management of Entity Lifecycle were:

- Conceptual model for the entities: how can one define and identify entities under the assumption of unique identity, as a complement to existing practices of the Semantic Web?
- Representation of entities: how can one represent entities in a flexible way, to allow for minimally restricted content in entity descriptions, while supporting metadata-rich content?
- Access control of entities: how can one support the entity repository operations in terms of security in a large-scale, open and distributed environment, keeping flexibility together with efficient access control?
- Maintaining entity profiles: what is a good set of strategies to manage the maintenance of a large set of entity information, where updates from an open community can significantly change and even degrade information over time?

Entity resolution and ranking: this challenge has been elaborated on in Section **Error! Reference source not found.**

3.3.3 OKKAM Solutions

3.3.3.1 *Conceptual Model*

For the conceptual model we introduced and utilized the notion of rigid identification or **identification by direct reference** in the Semantic Web. The solution is described in the **OKKAM conceptual model**, where we make a core distinction between web resources and non-web resources. Web resources are computational objects that are accessible on the Web by dereferencing a URI. In other words, web resources are entities that exist on the Web. On the other hand, non-web resources are real world entities that cannot be accessed on the Web, e.g., the city of Vienna. It is common in the Semantic Web literature to distinguish two identifying functions for http URIs in response to the distinction between web resources and non-web resources:

- I. Identifying by access, which holds between URIs and web resources;
- II. Identifying by description, which holds between URIs and non-web resources.

The OKKAM conceptual model introduces a novel type of http URIs – OkkamIDs – with a novel identifying function:

- III. Identifying by direct reference.

While the function of Semantic URIs is to identify non-web resources giving descriptions of them and thereby communicating information about them, OkkamIDs identify non-web resources without spreading information about them. The distinction between OkkamIDs and semantic URIs is not based on a difference in the architecture of the web. Indeed, OkkamIDs, too, when dereferenced, redirect to computational objects – OKKAM profiles – containing information about non-web resources (such profiles are stored in the OKKAM repository). Instead, the distinction is grounded on pragmatics and concerns the way clients make use of the information contained in those profiles. Such information is not used to describe the non-web resources but to fix the reference of other semantic URIs and to express the fact the two or more semantic URIs identify one and the same non-web resource giving different descriptions of it. OkkamIDs function as fundamental tools for linking data and collecting information about non-web resources. In fact, if one knows the OkkamID of a non-web resource, then one can use it to query the Semantic Web and collect all the semantic URIs of that resource with their associated information. In other words, the OkkamID of a non-web resource is a sort of gateway to the semantic URIs and the associated information about that non-web resource.

3.3.3.2 *Representation*

In order to efficiently represent and handle entities and corresponding information (profiles), in an open domain, the OKKAM system is designed to store free-form information on entities. The representation of the entities in the system is flexible in order to accommodate the requirements of all the different domains, where it may be used. Therefore, OKKAM uses **no fixed schema** for representing entities. Note that in OKKAM we are merely interested in assigning and managing unique ids to entities, which means that we do not need to represent all the known information about an entity, but rather the minimum amount of data that can help us discriminate this entity from all the rest.

A set of attributes described by **free-text attribute key-value pairs** is used to describe an entity. Each attribute is backed by a **set of meta-data**, which allows additional control over access rights and other meta-information. Proposed attributes, also termed **default attributes**, are used to facilitate non-restrictively a minimum set of homogeneous information per entity, based on its semantic type (e.g., person, location, artefact).

3.3.3.3 Access Control

In the setting of large-scale identification of entities, a set of different actions on the data and a large number of users of various roles is given. The challenge is made more difficult by the fact that we need to be efficient in a highly distributed environment. Furthermore, there is a need to facilitate both software agents and human in the most transparent way possible.

Entity management and entity lifecycle management functionalities are provided to users of OKKAM by means of WS APIs. There are different types of clients consuming those APIs: third-party institutions and organizations external to OKKAM that wish to adopt OKKAM APIs into their information systems, and external to ENS Web front-ends that provide intuitive and easy to use UI to end-users for managing ENS entities and life cycle. In the latter case, end users interact with ENS indirectly by means of front-end interfaces, which in turn interact with ENS APIs on behalf of end users. OKKAM consortium has provided several front-ends to facilitate ENS adoption and usage by end users, such as the administration console front-end and the entity creator front-end (also called ENS web toolkit).

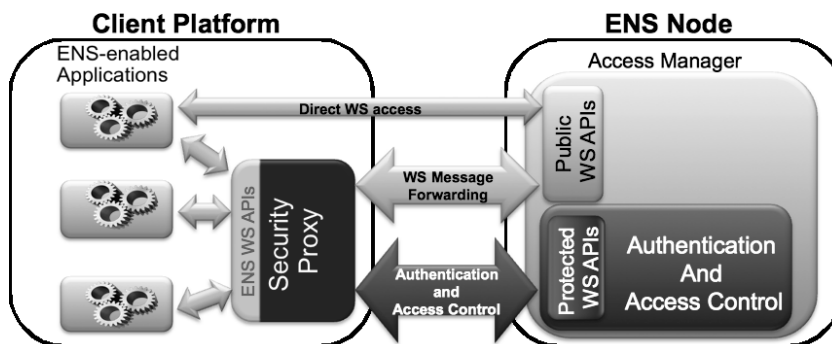


Figure 1: API-level Proxy-based Communications

A main cornerstone of the security architecture is the design of **security proxy** to facilitate easy adoption of ENS functionality into existing information systems. **Error! Reference source not found.** shows the designed API-level proxy-based communications. It has been developed a user-side security proxy component enabling secure and trusted interactions with ENS APIs. The security proxy hides the complexity of adopted Web services security standards used for confidentiality and data integrity of message exchange with ENS APIs, and abstracts certificate management and access control model implementation from the application level.

Effective access control enforcement is achieved by means of **bilateral user-to-ENS trust establishment process** allowing users and the ENS build confidence in each other to proceed with a service access.

The access control specification is based on a **semantic access control model** that defines in a scalable and flexible way semantic abstraction of policy specification from policy applicability to resources. The model defines access policies requirements bound to specific semantic properties, and defines a separate policy applicability specification that, during run-time operations, determines which semantic policies apply to what ENS repository entities taking into account both, the semantic properties of the access polices and the semantics of ENS entities. In that way, a large amount of ENS repository entities could qualify to relatively few access policies specifications and, as consequence, facilitate scalable access control management.

3.3.3.4 Maintaining Entity Profiles

The maintenance of entity profiles required a multi-aspect approach. We achieved to provide processes and tools for:

- Accessing and updating information through APIs and Web-UIs. This was supported by the **Entity Management Kit**, where the user can create, update, split, merge, and delete entities based on one's access rights. Individual steps of the process, allow for interoperability with other systems (through **alternative IDs**) and linking to existing entities, using **OKKAM ID references for attribute values**.

Another solution for the maintenance problem is related to **online resolution of entities**. This is the case of checking whether a candidate new entity exists in the repository. The user is informed in real-time on whether there is a near-duplicate entity in the repository. Furthermore, providing **feedback to the user** during the creation of an entity and guiding her **to maximize the quality** of entity information she inserts, supports the overall repository maintenance.

- Informing users on changes of entity data, based on their preferences. This was supported by the **adaptive subscription mechanism**, which allows users to subscribe to entities according to their interests and then get reports on changes in the profile information of these entities. In order to facilitate the users, the subscription system sorts the changes sent to each user by how interesting she finds the change. The system learns what is interesting per user, based on a feedback process and a machine learning backbone. The aim of the adaptive subscription is to optimize the information flow to the users of the open community, so that they can react quickly to correct errors in the entity profiles of their interest.
- Automatic detection of **aging in information**, through a set of algorithms that model when an attribute-value pair can be removed as redundant information.
- **Bulk resolution** of entities. This process allows minimal noise to be inserted into the repository during mass imports of entities (bulk OKKAMization). It also supports the verification of the quality of the repository as a whole at any given moment in time.

3.3.4 Lessons Learned

The main lessons learned in the domain of Entity Lifecycle were the following.

- The **early-adopted holistic view** on the problem of entity representation, allowing an open set of attribute key-value pairs in entity descriptions proved critical in the next steps.
- The **support of meta-data** in the representation of the attributes allowed flexibility when the need appeared, e.g., in the case of the private OKKAM nodes, where additional levels of information visibility needed to be supported.
- The **reuse of components and good practices** between different WPs – e.g., the use of the matching techniques in bulk resolution - helped maximize the profit of existing research and facilitated the flow of information between parties.
- The adoption of a **transparent approach for the security layer** helps increase adoption of the system and easily supports extensions to the core system.

- The **conceptual positioning of OKKAM** in the current Semantic Web landscape, even though non-trivial, proved very important as a motivation factor for the use of the system by 3rd parties, prolonging the lifecycle of the whole system as an infrastructure.
- Daring, through focused study, to offer **novelty, always pays off at the end**. Individual efforts in sub-problems of the Lifecycle Management, as well as the new view OKKAM offered as a base hub of rigid identifiers, managed to combine effectively research with active application.

3.4 Building Entity-centric Tools and Applications

Knowledge Management is one of the most important tasks in any organization - essentially all applications access data or produce data in a form or another. In recent years, the unstructured information sources such as documentation, user generated content and even the World Wide Web have started to play an increasing role in the development of the new business applications. In particular, business domains such as Business Intelligence, Customer Relationship Management, Help Desk Solutions, Call Center Applications, Product Information Management, are only some of the areas that require support for the management and understanding of unstructured data.

In this section, we discuss the building of entity-centric tools and applications from two main perspectives: first, from the perspective of the enterprise applications that can actually benefit from an entity centric approach and secondly, from the tool support and integration needs that can make this approach a reality.

3.4.1 Introduction and Context⁴

Out of these areas mentioned above, we have selected three classes of applications (namely, *Enterprise Search*, *Supplier Relationship Management* and *Analytics over Unstructured Community Data*) that we have investigated in our work and for which an entity centric approach in general and OKKAM in particular presents clear advantages.

Enterprise Search

Enterprise Search (ES) is different from searching a digital library or the Web in many aspects [1]. One of the major differences is that the corpora to be searched are highly specific and centred around a set of Enterprise Entities (EE), relevant to the company's business, such as products, business processes, purchase orders, etc. These entities are often captured in structured knowledge bases in the form of enterprise ontologies or dictionaries [3]. Using the knowledge about EEs for enterprise search promises to capture more of the semantics within documents and within queries and thus to improve the search results. But this promise is impeded by numerous problems. In particular, it is difficult to identify EEs in text due to syntactic heterogeneity and ambiguity. Specialized systems need to be developed in order to improve the identification and disambiguation of EEs, which in turn, would help to improve the quality of the search results. The OKKAM ENS together with the appropriate okkamizers and matching modules, represent examples of such systems.

Supplier Relationship Management

⁴ Please note that some of the discussions in this and the following sections are excerpts or summaries of more detailed analyses provided in D9.2 [2].

Currently, valuable business information is still stored and exchanged as unstructured data in various forms, such as fax or emails. For example, documents exchanged between business partners capture information on transactions between them like purchasing or invoicing. A major challenge is to recognize entities of interest like products or companies. Named Entity Recognition (NER) is the task of identifying real-world entities in unstructured data. So far, many approaches have been developed, focusing on common types of entities such as persons, organizations, etc. (see [4] for a recent survey).

Analytics over Unstructured Community Data

Benefits of text analysis are widely recognized in many areas, such as marketing (user satisfaction, user demands measurements), public relations (brand and product positioning), and software management (finding most common bugs, new usage scenarios of software, etc.). Business Intelligence (BI) over unstructured text has been under intense scrutiny both in industry and research. Recent work in this field includes automatic integration of unstructured text into BI systems, model recognition, and probabilistic databases to handle uncertainty of Information Extraction (IE) results.

The OKKAM empowered tools are come to support the creation of new OKKAMized content and have the role of enabling an entity centric approach across the classes of applications mentioned above. The OKKAM empowered tools represent not only tools but also a typical three tier architecture, an architecture in which the user interfaces, business logic, computer data storage and data access are developed and maintained as independent modules, on separate platforms.

3.4.2 Identified Challenges

In this context, one of the major challenges is to develop methods and techniques able to connect the corporate structured data with unstructured data. Since in general, the structured data has been designed to capture information about real world entities, the natural approach is to attempt to recognize and extract the occurrences of the very same entities in text. Nevertheless, this is not at all a trivial task: if the entities defined and used in a database or a business warehouse are homogenous and follow an original design, in the case of text documents entities can occur in various forms, referred through the use of various names, abbreviations or identifiers.

Enterprise Search

Evidently, user queries as well as documents deal with products or concepts of the company [5], i.e., enterprise concepts or entities. A natural way to improve the response quality is to first identify all EEs in a query and match them to the EEs in documents. However, usually neither queries nor documents mention concepts in the same way as they are described in a knowledge base. Users often use ad-hoc abbreviations, drop tokens in multi-token concepts, misspell tokens and intersperse their queries with error messages, trace snippets or code examples. In our work we have been especially focusing on answering queries that are short text messages (> 5 tokens) rather than only keywords, which comprise 10% to 15% of the distinct queries, i.e. 600,000 per week, in our use case scenario.

Supplier Relationship Management

Obviously, any support for automatic processing by for example Named Entity Recognition (NER) would speed up the process, reduce human errors and thus save costs. We have investigated three cases within the SRM domain where automation could be applied: entity pattern recognition, context pattern recognition and entity type pre-selection [2].

Analytics over Unstructured Community Data

The data preparation process for BI over unstructured text is still error prone and involves significant manual effort to reach sufficient data quality for performing the actual analysis (e.g., removing false positives). One of the main reasons is the variation in the quality of Information Extraction task, which is used to bring the unstructured text into a BI engine. IE achieves an accuracy of 90-98% in identifying simple entities and just 50-60% for complex entities (e.g., relations between entities) [6]. Moreover, according to a recent on IE [7], it is almost impossible to guarantee 100% accuracy in a real world setup.

Issues that make IE a hard problem are, to name just a few, human language complexity, high ambiguity of words, insufficient domain knowledge, and the presence of typos and misspellings in the text. All these issues lead to a complex IE development process, which involves extraction, evaluation, learning, and tuning of the IE execution plan. Therefore we need tools and methodologies that help an IE developer to discover errors of IE and extend the knowledge about the text corpus (e.g., learn new regular expression or entity patterns, new relations between entities, etc.).

Another important challenge is the design and development of the right OKKAM empowered tools for supporting the creation of new OKKAMized content. In general, this will be achieved by designing and implementing plug-ins for popular content creation tools: these plug-ins will interact with the OKKAM infrastructure for getting and manipulating entity identifiers and profiles. In this scenario, we have selected four main areas that we have investigated in our work:

- *Identification of tools that could be equipped with OKKAM plug-ins.* The challenge is to identify a tool mix that shows a good usage and spread. These tools cover different areas and domains, web and desktop applications and have to be a mixture of open sources and commercial tools. In particular in this time period where the document editing world is moving from the client to the online cloud, one of the main challenge is to create tools usable for desktop editing tools (Microsoft Office, Openoffice) but also online editing tools (wiki, blogs, email ...). Working on what spins around the web, from blogging to creating online contents to spread information and putting identifiers into the documents is a challenge involving several communities in the IT world.
- *Creating a distributed maintainable architecture.* Covering a wide user base and enabling caching systems will be possible through a scalable architecture. The creation of such an architecture is one of the most important challenge in OKKAM.
- *Making easy to integrate third-party tools.* An easy way to integrate the plug-in and a good usability are needed to increase the distribution of the different OKKAM empowered tools.

3.4.3 OKKAM Solutions

In order to address the quality issues introduced by the Information Extraction, the first step is to maintain a pool of entities of interest, unambiguously identified and associated with a minimum of information necessary to discriminate between them when no appropriate identifier is known.

Enterprise Search

OKKAM represents a natural fit in this landscape. It allows the creation of profiles containing accurate descriptions of the entities of interest, attached to a persistent unique identifier. Furthermore, the alternative identifiers that have been developed for the same entities in the context of the existing applications can be attached to the very same profile as alternative identifiers.

OKKAM can be used to maintain both a private and a public view on the entities of interest. Details such as planned releases, intermediary versions, etc. can be maintained internally in a private node,

while the public description of products and software components can be made available to the public node. The usage of the public node for this purpose represents in fact an opportunity to advertise and encourage business partners or developers to use the correct information for referring to certain aspects and details of the products in the company's portfolio.

In addition, used in conjunction with a public node, an OKKAM internal node can be used to forwards search request to the outside world and to retrieve further documents related to the user query. Besides enabling the access to more information and relevant resources, this approach exposes the users and developers to opinions, problems and solutions that might have not been known inside of a closed ecosystem.

Supplier Relationship Management

As it can be seen in this scenario, it is crucial to be able to identify, ideally in an automatic fashion, the entities of interest and more importantly, to be able to seamlessly share them across applications and even across companies. While in certain cases pattern recognition can be used to recognize certain occurrences (as those for examples of the invoice identifiers), in order to provide support for proper disambiguation, contextual information has to be retrieved first.

In the SRM case, the business documents refer to the products (e.g. through their identifiers) that represent the scope of that business transaction. These products will be subject of further references by other business documents, so as a consequence information about them should be created and managed in a centralized, coherent and consistent fashion. By adopting the ENS approach as proposed by OKKAM, a unique profile together with the relevant metadata can be maintained for each product. The alternative identifiers used internally by various applications inside the company, can be gathered and maintained around a unique, persistent identifier associated with this profile.

Regarding the inter-corporate interoperation, the internal OKKAM nodes that maintain profiles containing sensitive information on the entities of interest (e.g. products) can be synchronized with the OKKAM public node. That is, non-confidential information can be provided for the public profiles of the same products that are maintained internally in the private node and more importantly, the same identifier (or one of those specified as alternative identifiers) can be used in the public profile. In this way, having the public OKKMA node as a bridge, identifiers understandable by external business parties can be used, without affecting the internal, legacy applications.

Analytics over Unstructured Community Data

By extending the BI capabilities in such way that they cover not only the structured data but unstructured text as well, enables the business analyst to gain new insights in to areas such as marketing, public relations, software management, etc. Entities play a central role as they need to be extracted from documents, correlated with each other or with the already existing facts in order to derive statistics based on their occurrences in various contexts. That is, being able to uniquely identify the same entities in different contexts represents a major challenge. By using OKKAM and OKKAMized content outside the corporate boundaries (e.g. on the Web), it becomes possible to extend the analysis, on-demand over external resources. That is, if external Web pages are annotated with OKKAM identifiers and in parallel, the same identifiers are also used to annotate internal documents or the corporate structured data, an aggregated analysis over all these heterogeneous sources can be performed. This means being able not only to connect text analysis with classical BI but also to step beyond the corporate boundaries and learn how a company like SAP is perceived or reviewed and maybe even more important, how SAP products perform in respect with their competitors.

To address the challenges related to the development of the OKKAM empowered tools, a 3-tier architecture has been developed in the OKKAM project and adapted during the project to comply with newly identified challenges. The OKKAM empowered tools architecture is a typical three tier architecture. That is, this is a client-server architecture, with three main layer:

- **OKKAMcore Layer.** This layer contains the ENS services, which are available via Web Services. The ENS stores unique identifiers and offers services to applications in order to retrieve them. The information is then passed back to the logic layer for processing, and then eventually back to the user.
- **Business Logic Layer.** This layer contains the Application services API. In particular, it coordinates the application, processes commands, and makes logical decisions and evaluations. The Application Services API automatically extracts entities from documents and Web resources; it also moves and processes data between the two surrounding layers. The Application Services API allows the following operations starting from a document:
 - Recognizing the named entities like People, Companies, Location, and other
 - Recognizing the OKKAM entities
 - Generating the OKKAM core queries extracting the named entities
 - Enriching the document with the OKKAMid
 - Enriching the document with the RDFa
- **Presentation Layer.** This is the topmost level of the application. The presentation tier displays information related to such services for each tool, in commercials and open source environments. The first and second OKKAM empowered tool suite contains different plugins for different tools. All this tools are described in detail in the deliverables D7.1, D7.2.

3.4.4 Lessons Learned

By analyzing the classes of scenarios described above, it is clear that the so-called enterprise entities represent pivot enablers when information integration, data aggregation or analytical tasks need to be performed. The continuously growing importance of the unstructured information sources however, has significantly changed the way the enterprise entities need to be managed and handled. Documents refer to the same entities as those defined in the corporate structured data but in a much fuzzier way; applications resort more and more to the unstructured data in order to augment, complete or improve the functionality they offer. In order to reduce the costs associated with operations such as entity extraction, entity disambiguation, etc. and to enable the interoperation among the applications concerned with the same enterprise entities, a consistent, centralized view on these entities has to be maintained. Such a view has to be flexible enough in order to fully match various application contexts without need for change in the business logics of these applications. That is, this view should include only essential information sufficient for the identification and recognition of these entities.

By using the OKKAM ENS, such a view on the enterprise entities can be created and maintained. Each entity is associated with a unique identifier and a profile, containing only as much information as needed to recognize the entity when the identifier is not known. In fact, at a closer analysis, one can recognize that this profile represents nothing else than the metadata of that particular entity.

A very important aspect of OKKAM in this context is represented by its public deployment. One of the major pain-points in managing the entire ecosystem around the company's products and solutions, is making sure that business partners and customers have access to (and use) up-to-date information about the company products. By using a public OKKAM node, the company has one more instrument to proactively disseminate correct information about its solutions and use it as support in solving enterprise information integration issues.

In respect with the different OKKAM Empowered tools, their implementation and evaluation was a complex process that led to some important lessons and experiences. The most important lessons are listed here:

- A great interoperability and modularity between the different architecture layers are needed in order to have a good scalability
- The tools usability depends not only on the different approaches, the Cognitive walkthrough and the Heuristic Evaluation, but also it is strictly related to the feedback collected from the user community
- The distribution of software is strictly related the distribution and commercial approach used in the project;
- OET are the tip of the iceberg OKKAM, a problem during the use of OKKAM empowered Tools involve directly the entire project because could be a problem in a different level of the architecture.

3.5 Operating Entity Identifier Management as a global public Infrastructure

3.5.1 Introduction and Context

The OKKAM WP5 (Integration, Technical Management and Roll Out) was focused on production and delivery of the operational global ENS public infrastructure and service. The integration management approach⁵ involved a quantifiable overall development methodology to encompass developer guidelines and release cycles, with a high emphasis on software quality measures (testing, performance etc). Throughout the project, this methodology was actively used, reviewed and evolved to formally manage and control activities towards a coordinated and effective global public infrastructure and service, to realise the project goals and full potential of the technical achievements of the other WPs and PCAs.

The OKKAM development and production environment was designed to be open, agile, distributed and involve continuous integration, with a software development approach that was also open, focused, distributed and agile. This was motivated by:

- the ambitious nature of the complex OKKAM system,
- the short timescales between the Integration Prototypes (which provided the ongoing direction and focus),
- the heterogeneous distributed development teams of the various partners
- the need to deliver the system on time and within budget.

⁵ As defined in OKKAM D5.1(OKKAM Development and Deployment Guidelines), April 2008.

The aim was to ensure that OKKAM would be developed to the highest standards in a very integrated and parallel way throughout the project and not just at the end, with coordinated and consistent design and development of the various components and services within the OKKAM infrastructure, especially with respect to mutual interfaces;

OKKAM used a collaborative distributed agile development approach as used for Open Source software development, but within the project consortium and on the project's own collaborative development management gForge platform at www.okkamdev.org.

3.5.2 Identified Challenges

Production and operation of the Entity Identifier Management as a global public Infrastructure involved the engineering management and integration of many distributed research and development teams, from academic, research and commercial partners, and outputs from many very different Workpackages and Project Competence Areas (PCAs), in the large OKKAM Integrating Project. So ensuring focus on production of a service, while encouraging engineering and research excellence and managing the distributed development from such varied and heterogeneous development teams over the relatively short 30 month duration of the project, proved to be quite a challenge.

Specific identified challenges included:

1. Ensuring that all of the various and distributed WPs and PCA's had a clear vision of the overall ENS Service.
2. Putting in place strong procedures to ensure engineering excellence, but not too heavy to stifle the research interest and excellence, while progressing towards an operational service and public infrastructure. How to balance research pursuits with engineering excellence.
3. Day to day management of the system and the process of committing new updated components, while ensure 24x7 availability of the public infrastructure and service.
4. Planning the evolving user requirements, system features and service releases to which the various teams would agree and deliver.
5. Managing and providing the OKKAM Server platform as it grew bigger and ever more complex.
6. How to ensure that new components that were being added to the system did not crash the system and stop the public service.
7. How to accommodate open source requirements for a public infrastructure, along with proprietary software for commercially exploitable applications.
8. Deciding when and how to make the public infrastructure open source: too soon and we lose the community. Too late and they have passed us by.
9. How to maintain, evolve and grow the public platform using open source generic components (such as Hbase and Voldermort), whose quality varied with new releases over the life of the project.

3.5.3 OKKAM Solutions

1. Integration Prototypes and Demonstration Subprojects (ISPs)

Integration and engineering of the ENS global public infrastructure and service in the large disparate OKKAM project was organised into 3 OKKAM Integration Prototypes and Integration and Demonstration Subprojects (ISPs) and subsequent Project Reviews, as follows:

- OKKAM Infrastructure V1 – Nov08, (Review 1 – Feb09)

- OKKAM Infrastructure V2 – Nov09, (Review 2 – Mar10)
- OKKAM Infrastructure V3 – Jun10, (Review 3 – Jul10)

These structured the project development and integration, fostered systematic evolution towards the targeted OKKAM infrastructure and ensured integration of the contributions from the different WPs and PCAs into the infrastructure and the applications built upon it.

This OKKAM ISP approach allowed the project to

- foster systematic evolution of project results
- ensure horizontal integration of RTD contributions
- result in integration prototypes

and were the main drivers during the project life-cycle, causing all work to converge 3 times during the 30 months of the project. This worked very well.

2. Feature-Release Matrix and Verification Plan

To address the challenges listed above, the engineering procedures of the project (which were defined in D5.1) were reviewed and evolved throughout the project, in the light of experience and what did (and did not) work. For instance, in light of feedback from the First Review (Feb 09), a critical assessment of the OKKAM software development process was performed. As a result stronger management-driven planning of further releases was established and stronger management structures for monitoring the progress towards the releases was put in place.

The more formal release planning was documented with the identification of pending requirements in a Feature Matrix for scheduled releases. In particular, the functional specification of the ENS was refined and prioritised to a list of 12 Requirements and 6 realistic Performance Targets. These set the basis of the critical success factors for the ENS service. The ENS Feature-Release Matrix was developed from the agreed list of user requirements, and a Roadmap of ENS Releases beyond the end of the project was circulated and agreed by all Partners. This was then used to develop a draft Verification Plan for each release of the ENS.

The establishment of these planning tools led to more transparency for the partners on which functionality was expected at which time, which enabled them to prioritise their work in line with the overall project goals.

3. Technical Task Force

On the organizational level, a Task Force for coordinating the software development, integration and deployment process was set up under the lead of the Technical Director, who formed a small, virtual team to overcome institutional boundaries, to lower communication overhead and to speed-up reaction times. This group had the ability to take ad-hoc decisions and the skill to implement changes, and was in charge of organizing the integration tasks in the completion of each new release of ENS for the production platform. The Task Force kicked-off with an internal meeting in to begin the process by determining what would be technically feasible within available timescales, agreeing who would do what, and the procedures to be adopted to ensure success (in the style of the

“coding retreats” described in the DoW⁶). The Technical Task Force then worked intensively and collaboratively during the remainder of the project, on development of the ENS up to V3 with numerous virtual meetings (using Marratech⁷, which proved to be very useful). The team members interacted in a fast and effective way to ensure on-time development of OKKAM public infrastructure and ENS service towards the targeted features, testing, integration and deployment of the components for the ENS, as well as ensuring adequate preparation of the components before integration (e.g. documentation, provision of unit tests, component testing).

4. Requirements Management

As the focus of WP5 was on production of the public global infrastructure and ENS operational service, an ENS Architecture document was iteratively refined and circulated amongst the Partners. In addition, a Requirements Management System was set up to systematically revisit requirements (driven by the needs of the application partners) and for monitoring progress towards fulfilling those requirements, which also contributed to improving the Software Engineering process in OKKAM. Partners were asked to include in their requirements not only the requirement description itself, but also a description on how fulfilment of the requirement could be measured. This enabled a more thorough verification process to be implemented.

5. Test Beds, nightly system builds and code reviews

To optimize the integration process, two cluster environments were set up which enabled updates, fixes and new releases of the ENS to be staged on a testing environment, and to be deployed at the operational site after successful verification. Due to the high complexity of the ENS system, nightly builds were implemented using the Hudson build manager, to reveal any sub-project commits that may contain errors. These nightly builds proved to be very effective and useful. In addition, code reviews were performed to ensure that the exchangeability of ENS components, and that contracts between components were not violated.

6. Distributed Development Management Platform and Forge

The OKKAM software was collaboratively developed using the <http://www.okkamdev.org> project development website⁸. The required projects OKKAM-core-test-suite and ENS are hosted there and are freely available to download to any developer with an OKKAM login and password.

The OKKAM Community Portal⁹ proved to be very useful. Documentation for external Developers is being actively written and made available on the OKKAM Community Portal, which also includes the Public ENS Issue Tracker.

7. Open Source and Dual Licensing Model

⁶ DoW section B.1.4.2.4

⁷ www.marratech.com

⁸ The www.okkamdev.org collaborative development site is described in D5.1 (OKKAM Development and Deployment Guidelines), Mar08 and discussed in D5.2 (Integration Prototype OKKAM Infrastructure V1), Dec08.

⁹ At <http://community.okkam.org/>

The ENS OKKAM public server public infrastructure components and ENS Software Libraries were released under a dual hybrid open/proprietary license model designed to meet the development and distribution needs of both commercial distributors (such as OEMs, ISVs and VARs) and open source projects (GPLv2) based on the successful MySQL licensing model.

3.5.4 Lessons Learned

Engineering Technical Management should include formal and structured processes of software production, versioning, quality assurance, with a strong management-driven planning and monitoring of progress towards further releases of the service.

Integration and Technical Management of a public infrastructure and service, while encouraging innovative research and engineering excellence based on inputs from a number of heterogeneous and distributed development teams is not easy and takes a lot of time ! Projects should allocate at least 15% of their effort to this vital task, if useful and usable infrastructures and services are to result from RTD projects like OKKAM.

Structuring the project work plan into 3 Integration Subproject Prototypes were the main focus and integration drivers during the project life-cycle and did cause all work to converge 3 times during the project. This is a model that could be followed in other similarly complex and diverse projects.

Use of the Community Portal and <http://gforge.okkamdev.org> distributed development management platform proved to be very efficient and useful, and provide a Forge platform to complete the system's Open Source Software release process.

A viable Hybrid Dual Licensing model, with GPL v2 for the public infrastructure and proprietary licensing for value-added applications, based on the proven MySQL model is possible for a global public service such as ENS.

While planned Integration and Development procedures have to be put in place, these should be constantly used, reviewed and improved throughout the life of a project such as OKKAM.

The following were found to be particularly effective:

- Technical Task Force working collaboratively and effectively, with light weekly reviews and progress reports.
 - To coordinate the software development and deployment process.
 - A small, virtual team to overcome institutional boundaries,
 - Lower communication overhead, able to ensure on-time development of the service towards targeted features, testing, integration and deployment of the public infrastructure components,
 - Ensuring adequate preparation of components before integration (e.g. documentation, provision of unit tests, component testing).
- Optimised integration process with separate Test and Operational Cluster environments and clear commit procedures.
 - With updates, fixes and new service releases first validated on the Test Bed and then deployed at the Operational Site
 - Nightly builds using an automated build manager, to reveal any sub-project commits that may contain errors are very useful.
- Test Bed systems maintained at various sites for local component testing and quality standards validation

- Code reviews to ensure the exchangeability of all system components and that contracts between components are not violated.
- A formal service release planning process is required. Starting from identification of requirements using a Requirements Management System (driven by the needs of Application Partners) and defining a Feature Release Matrix for scheduled releases to focus developments in individual WPs
 - A service Feature-Release Matrix and Roadmap should be developed and agreed to include a Formal Release Planning Process of
 - Requirements
 - Features
 - Verification

4 Conclusions

All in all, the conclusion we can draw from this experience is that entity management on the web is definitely needed, but the exact form in which this may happen is not completely defined. OKKAM had the merit of bringing this issue to the attention of the overall community at a time when the problem was not very clearly perceived, and has produced some tools and services which will play a role in the future entity management. In particular, we'd like to mention the Entity-centric Semantic Search Engine Sig.ma, which implements a completely different paradigm of data navigation on the Web of Data, in which the basic elements are not nodes in a graph, but entities. And we can also mention the family of ENS-empowered tools, which show that an entity-centric approach to content and data authoring is feasible and can be effective (see e.g. the ANSA experimental portal of enriched news). But we can't predict if the ENS will become the solution for entity management on the web, or if a different solution will finally prevail. For sure, today the OKKAM technologies and tool suite are the most advanced solution in this domain, and the creation of an OKKAM public trust and an OKKAM spin-off company (see “D14.6 - Dissemination and Exploitation Plan Update” and “D14.8 - Sustainability plan”) is an attempt to test the proposed solution in the real world.

5 References

- [1] A. Z. Broder and A. C. Ciccolo. Towards the next generation of enterprise search technology. *IBM Syst. J.*, 43(3):451–454, 2004.
- [2] A. Mocan, W. Barczynski, F. Brauer, and G. Hackenbroich: D9.2 Lessons learned for Organizational Knowledge Management. *OKKAM Deliverable*. December, 2009
- [3] J. L. G. Dietz. *Enterprise Ontology: Theory and Methodology*. Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2006.
- [4] W. Barczynski, F. Brauer, A. Mocan, M. Schramm and J. Froemberg: BI Style Relation Discovery between Entities in Text. *2nd International Workshop on New Trends in Information Integration (NTII 2010)*, 1-6 March, 2010, Long Beach, California, USA.
- [5] A. Löser, W. M. Barczynski, and F. Brauer. What’s the Intention behind Your Query? A few Observations from a Large Developer Community. In Proc. *IRSW 2008*.
- [6] R. Feldman. Tutorial: Information extraction, theory and practice. 2006.
- [7] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3): 261-377, 2008.