FALCON

D3.6: REVISED L3DATA FEDERATION PLATFORM RELEASE

David Lewis, Alfredo Maldonado, Brian Walshe, Kaniz Fatema, Anton Gerdelan, Arturo Calvo

Distribution: Public Report

Federated Active Linguistic data CuratiON (FALCON)

FP7-ICT-2013-SME-DCA

Project no: 610879

Document Information

Deliverable number:	D3.6
Deliverable title:	Revised L3Data Federation Platform Release
Dissemination level:	PU
Contractual date of delivery:	31st March 2015
Actual date of delivery:	30th June 2015
Author(s):	David Lewis, Alfredo Maldonado, Brian Walshe, Kaniz Fatema, Anton Gerdelan, Arturo Calvo
Participants:	TCD
Internal Reviewer:	DCU
Workpackage:	WP3
Task Responsible:	TCD
Workpackage Leader:	Interverbum

Revision History

Revision	Date	Author	Organization	Description
1	To 15/5/2015	David Lewis, Alfredo Maldonado, Brian Walshe, Kaniz Fatema, Anton Gerdelan, Arturo Calvo	TCD	redesign of L3data schema and data processing components
2	To 6/6/2015	D.Lewis	TCD	Document design
3	12/6/2015	Christophe DeBruyne	TCD	Proof read and suggested corrections
4	20/6/2015	D.Lewis	TCD	Corrections and formatting

Contents

1. Executive Summary

This deliverable presents the revised design of the L3Data platform components of the FALCON system. This consists of an L3Data Server component (L3Data Svr) component and an L3Data Management (L3Data Mgr) component. L3Data Svr that offers open RESTful interfaces to store and retrieve L3Data in the form of tables and table metadata conforming to the Comma Separated Value of the Web standard begin developed at the W3C, which is turn has been profiled to support the requirements for the FALCON system based on other Linked Data vocabularies. The L3data Mgr component coordinated some of the integrations between XTM translation management, the Text Analytics and Machine Translation components via sharing of L3Data on the L3Data Svr component. This deliverable outlines the novel aspects of the design, including the structure of the data, and outlines the interactions of component, which includes references to interactions defined in existing components. Finally the L3Data system is described in alignment to the Linguistic Linked Data Reference Framework developed in the LIDER project.

2. Introduction

The L3Data Federation Platform implements the L3Data schema and system architecture and API defined by FALCON Deliverable D2.2: Initial L3Data Schema and Architecture. The L3Data store is implemented using existing Linked Data stores such as Sesame[1] and Apache Jena[2] (with Jena being the configuration used in the original release). The implementation allows localisation project state to be recorded using the W3C Provenance RDF vocabulary[3], with specialisations taken from the CNGL Global Intelligent Content vocabulary as defined in D2.2. In this initial release the project state is captured from the XTM Cloud translation management tools in the form of a Translation Interoperability Package Protocol unit, and in particular the XLIFF1.2 file it contains. This is performed by a component called the Logger, which interacts with XTM Cloud using an open RESTful API. A further API is offered giving access to SPARQL query functionality over the project state. With this initial release the L3Data Federation Platform offers querying of, access to and recording of L3Data to other WP3 components, primarily the translation management tool set centred on XTM Cloud. The APIs offered also enable the development of further web-based tools to support monitoring of workflow and different Language Translation (LT) component and human worker performance through visualisation of provenance queries conducted over federated L3Data stores, as well as to generate reuse audit reports.

This document provides an overview of the platform, as already specified in D2.2. This initial release will be made available via the FALCON GitHub repository[4]. While this document gives an overview of this release, those wishing to explore the platform in more detail are referred to the GitHub for the latest code and documentation.

3. Revised L3Data Platform Overview

[1] http://rdf4j.org/
[2] https://jena.apache.org/
[3] http://www.w3.org/TR/prov-overview/
[4] https://github.com/CNGL-repo/Falcon

The W3C standards for creating, managing, interlinking and search open data of the web have matured to the level that they can fully support open, massively multilingual language resources that integrate semantic knowledge, lexical knowledge, corpora and online content and data sets of all types. Extensive suites of proven open source tools exist and there is an expanding migration of language resources to this technological platform, particularly in lexical-conceptual resources.

The three core principles of Linked Data as an open data platform are:

1. Linked Data ensures that data and services form a linked ecosystem rather than a set of fragmented and non-interoperable datasets and services. A growing set of standardised linked data vocabularies ensure convergence
2. Semantic Technologies conformant to web standards such as RDF and OWL offer powerful APIs such as SPARQL for search and RESTful services to publish, update and manipulate linked data on the web.
3. De-centralization is key in that the implementation of the architecture is Web-based and does not rely on any central node, service or particular providers of a cloud. In particular, this should prevent any vendor lock-in and dependencies on particular agents.

The Revised L3Data Federation Platform has be developed to reflect the experiences garnered from:

- Experience in applying Linked Data in various application areas, including language applications;
- The implementation of the initial version of the platform; a revised model of how L3Data can be integrated into localisation workflow and in particular the workflows enabled by the integration of the SME partner tools in FALCON;
- Investigation and exploration of emerging best practices and standards in the area of open data on the web and linguistic Linked Data.

For the latter, developments in open data on the web have tracked through the W3C Data Activity[5] and in particular the development of specifications in two active Working Groups: CSV on the Web[6] and Open Annotation[7] and one active Community Group: Open Digital Rights Language[8].

[5] http://www.w3.org/2013/data/
[6] http://www.w3.org/2013/csvw/wiki/Main_Page
[7] http://www.w3.org/annotation/
[8] https://www.w3.org/community/odrl/

Developments in Linguistic Linked Data are tracked through participation in the W3C Community Groups for Linked Data for Language Technology[9], Best Practices in Linguistic Linked Data[10] and Ontological-Conceptual Vocabulary[11].

The approach taken builds on DataID best practice for publishing linked data sets based on open vocabularies [Brummer]. This provides best practice in the use of Linked Data vocabularies for capturing the metadata of data sets published on the web. It proposed the use of the Data Catalogue vocabulary[12] (DCAT), combined with the Provenance Ontology[13] (PROV-O) and the VoID vocabulary[14] for describing RDF data sets. DCAT is use to record information about authorship, version, different distributions of the data set and the licenses under which those are published. It recommends capturing licensing information in a machine-readable form using the Open Digital Right Language (ODRL)[15]. The use of PROV-O allows the dataset of be modelled as an Entity object with provenance relationship to other Entities and too objects representing the Agents involved in producing the Entity and the Activities by which the Entity was produced. Modelling Entity, Agent and Activities as first class objects allows more detailed recording of provenance information than is possible using the Dublin Core Term[16] attributes used by DCAT. In particular PROV-O allows the modelling of relationships between these objects using attribute specified in PROV-O and also the specification of additional, application-specific attributes as required.

The PROV-O vocabulary was also applied to recording logs from multilingual content processing in the Global Intelligent Content (GIC) model[17] published by CNGL. This was aligned more specifically to the use cases in FALCON, as it addressed the provenance records from content processing chains consisting of heterogeneous language technology-based and manual content processing steps. The GIC model therefore represented multilingual language data, rather than data sets in general as in DataID. It therefore specialised PROV-O entity and activity classes to represent classes that would be relevant to multilingual content processing. These included classes and attributes taken from other vocabularies in this domain. Entity subclasses were subsumed from classes from the Natural Language Processing Interchange Format (NIF) vocabulary[18] for Word, Phrase and Sentence were included. Activity subclasses representing text annotation, translation and quality assurance were defined in the GIC vocabulary. Further, the Linked Data vocabulary resulting

[9] https://www.w3.org/community/ld4lt/
[10] https://www.w3.org/community/bpmlod/
[11] https://www.w3.org/community/ontolex/
[12] http://www.w3.org/TR/vocab-dcat/
[13] http://www.w3.org/TR/prov-o/
[14] http://www.w3.org/TR/void/
[15] http://www.w3.org/ns/odrl/2/ODRL21
[16] http://dublincore.org/documents/dcmi-terms/
[17] https://github.com/CNGL-repo/GIC
[18] http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html#

from the RDF bindings of metadata defined in the Internationalization Tag Set 2.0[19]. These were used to annotate sentences with metadata resulting from text analytics (e.g. links to entities in a knowledge base or lexicon), terminology and machine translation.

Experience in annotating language resources with open vocabularies as linked data was also gained from a collaboration between the Linked Data for Language Technology W3C Community Group and the META-SHARE community which had developed an XML metadata schema for language resources, which was used across a distributed set of bespoke nodes supporting the open registration, search and indexing of language resources under both open and restricted licenses. This collaboration resulted in an RDF vocabulary that is able to capture the data previously specified in the META-SHARE XML vocabulary. By making use of open data vocabularies for this metadata, most notably DCAT, it became easier to register, index and search META-SHARE language resources using RDF tooling rather than the dedicated XML-aware servers used previously. However, this only applied to the metadata of published language resources. The modelling of language resources in RDF was not attempted and remains an open challenge.

The revised model for L3Data was informed by this related work in it use of:

- DCAT metadata for cataloguing individual data sets used;
- PROV-O for annotating the resources produced by different components of the system workflow in terms of the activity conducted by that component in generating these resources and the software, human and organisational agents responsible for those activities;
- ODRL for recording the possession of rights over individual resources by the parties (corresponding to PROV-O agents) and the usage right offered to other the component.

The revised L3Data model however moves beyond the state of the art on the use of linked and open data in several important respects that are outlined in the following subsections.

Data Flow Model

PROV-O does not offer a complete record of the sequence of activities that operate on the entities of which it captures the provenance. It does enable the derivation of one entity from another to be recorded, or the fact that an activity that generated an entity made use of another entity in that process. The vocabulary does provide a link to a plan object that can capture such a sequence, but does not specify a vocabulary for this. A vocabulary has been defined for specifying simple activity data flow definitions that can be used to annotate provenance records. This consists of Step objects that can be specified with sequential dependencies within a specific Plan object. It can also contain data flow Variable objects that can be defined as the input or outputs of Steps. Attributes can be added to

[19] http://www.w3.org/TR/its20/

PROV-O records linking Activity instances to Steps in a Plan and Entity instances to Variables in a Plan.

For the FALCON showcase system the following Steps and Variables have been defined based on the process flow definition defined in [D2.2].

Steps	Step Definition
extractSource	The process of extraction and segmentation of translatable source content from the documents to be translated by the translation project executing the Plan
suggestTerms	The process of Automated Term Extraction (ATE) of suggested terms from the translatable source content.
validateSuggestedTerms	The process of human validation of term suggestions, identifying them as valid or invalid for the project.
buildMtEngine	The process of building an initial statistical machine translation (SMT) based on parallel text available to the project.
machineTranslateSource	The process of machine translating the project source.
optimisePeOrder	The process of optimising the order in which source content segments are machine translated and post-edited in order to optimise the capture of term validation or the generation of post-edits for MT retraining
guidedPostedit	The process of machine translating source segments and post-editing and capturing terms from those segments.
reTrainMt	The process of retraining an SMT engine based on post edits generated during the execution of a project.
completeProject	The process of completing a project.
publishData	The process of selecting data from the project to publish to a specified audience with specific usage rights.
retrainAte	The process of retaining an ATE engine for a client based on the validation of previously suggested

	terms for that client.
Variables	**Variable Definition**
sourceContent	The content submitted by a client for processing a translation project executing according to this plan.
segmentedSource	The translatable parts of the source content segmented for translation.
suggestedTerms	Terms suggested automated term extraction performed on the segmented content.
suggestedTermTranslations	Suggested translation terms of derived from automated search of a language resource.
validatedSuggestedTerms	The human validation of suggested terms.
validatedSuggestedTermTranslations	The human validation of suggested term suggestions.
mtEngineRef	The reference to a specific MT engine instance.
initialMtOfProject	The output of translating the segmented source content of a project.
optimisedPeOrder	A suggestion of the order in which the segments of a project should be post-edited.
posteditedTargetSegments	The translation of source segments resulting from human post-editing of suggested machine translations
targetTermsCapturedinPe	The target language translation of terms provided by post-editors.
publishedParallelText	The parallel text published at the end of a project.
publishedTerminology	The terminology published at the end of the project.
publishedSegmentToTermAnnotation	The published annotations between published parallel text and published terms form the project.
ateEngineRef	The reference to an ATE engine instance.

Table 1: Steps and Variables defined for the plan operated by the FALCON System

Open Annotation

There is a need to define clear usage rights declarations for the resources produced by components operated by different parties in the workflow. Some of these resources involve generating annotated relationships between input resources, e.g. between source segments and terms, or links between input resources and output resources, e.g. alignment links

between source segments and generated target language segments. As these annotation resources may have different usage rights than the content-based resources, i.e. segments and terms, these annotations are stored separately in order to enable rights declarations to be applied separately and to ease therefore the management of the lifecycle of these resources as these are used downstream in the workflow and later shared or published.

For this purpose the emerging standard being developed by the W3C Open Annotation working group[20] is adopted. This models an annotation as an object with one parameter pointing to the resource being annotated (referred to as the 'body') and the resource with which it is being annotated (referred to as the 'target'). Compared to solutions that embed the annotation in the body resource, this approach allows the annotation object to be annotated with metadata recording attributes of the annotation process, e.g. the provenance of the annotation itself. This also allows the downstream processing of the annotation to be handled separately from the body and target of the annotation.

The open annotation object is used where either the annotation or body and target originate from component operated by different organisational actors in the process value chain. For the purpose of this model, the most fine-grained configuration of actors is assumed, namely:

a. Client organisation commissioning the translation project;
b. The language service provider conducting the translation;
c. The Publisher of public parallel text used to train the machine translation engine used in the project;
d. The Publisher of public terminology resources used in the project;
e. The Language Technology service provider offering the Automated Text Extraction service used in the project;
f. The Language Technology service provider offering automated entity linking services used in the project to suggest term definitions and term translations. In this project Babelfy[21] and BableNet[22] are used in conjunction to provide this service;
g. The Language Technology service provider offering the re-trainable statistical machine translation engines used in the project;
h. The Translators commissioned to post-edit content for the project.

The open annotation model is then applied in the following annotation, with the differing organisations responsible for the body and target resources indexed by the letters above.

Body Resource	**Target Resource**	**Additional Annotation Properties**
Source Segment	Suggest Term (e)	Frequency score indicating the importance of the

[20] http://www.w3.org/annotation/
[21] http://babelfy.org/
[22] http://babelnet.org/

(a)		term to this segment and the source content as a whole.
Suggested Term (e)	Suggested Term Definition (f)	Reference to concept providing the definition; Confidence score in the accuracy of the suggested definition.
Suggested Term (e)	Suggested Term Translation (f)	Reference to lexical entry offing the suggested translation; Confidence score in the accuracy of the suggested translation.
Suggested Term Definition (f)	Validation of Suggested Term Definition (h)	Positive or negative validation of suggested term definition
Suggested Term Translation (f)	Validation of Suggested Term Definition (h)	Positive or negative validation of suggested term translation
Source Segment (a)	Term Frequency Score (b)	n/a
Source Segment (a)	Post-editing Priority Score (b)	n/a
Source Segment (a)	Suggested Segment Translation (g)	Confidence score for machine translation; time taken to translate segment; words in the source segment not known to the SMT engine; terms in the source segment for which the translation is taken from terminology
Suggested Segment Translation (g)	Post-edited Translation (h)	The time in milliseconds taken by the post-editor to generate the post-edited text
Suggested Segment Translation (g)	Validated Term Translation (h)	n/a

Table 2: Summary of open annotations objects used

Tabular Data

The analysis of the requirement for the FACLON showcase system indicated that the main resources being manipulated and passed between component could be adequately modelled as tabular data, specifically tables of segments and their translation and table of suggested and validated term. Further, the requirements did not reveal use cases that required ad hoc searches across the data produced from different components was a pre-requisite. Instead, the data exchanged between the LT component would input and process the whole output of previous components. Therefore there was no direct requirement to maintain data in an RDF format. This was compounded by assessments of the initial L3Data

model [D3.2] indicating that the use the NIF format to exchange translation project data between component required a seven-fold overhead in the volume of data exchanged. As language technology component would load resources, typically source content, parallel text or lists of terms, and process sequentially internally, there was no loss of functionality from forgoing an external search capability on the resources output from different components.

For this purpose the emerging W3C standard, CVS on the Web (CSVW)[23] is adopted. This allows tabular data to be store as web resources in the popular Comma Separated Value (CSV) format[24], which is accessible to developers across a wide range of systems types. Anecdotally, this format was found much easier to deal with by both the localisation tool developers and the NLP processing component developers represented in the FALCON consortium. While the localisation tools, such as XTM, TermWeb and Easyling used in FALCON, may support XML-based interoperability formats such as XLIFF, TMX and TBX, but support for data export/import in CSV is widely supported due to its near ubiquitous support is a wide range of tools, including spread-sheet applications. Tabular data, in particular tab separated tables, is the standard method for exchanging multilingual textual data between NLP components. NLP processing libraries are well optimised to support tabular data and NLP developers are comfortable with its use. Meanwhile, support for XML formats, while common in localisation tools due to standardisation effort for translation memory exchange (TMX) and terminology exchange (TBX), but is less common in NLP solutions. NLP integration platforms such as UIMA and PANACEA offer solutions that work once the NLP component has been ported to that platform. Such solutions are less appropriate for interoperability in federated solutions where parties may have differing software architectures and many different data-exchange partnerships and so can rarely be confined to specific platforms. While NIF offers an RDF-based solution for exchange of NLP data between tools, it suffers from both the complexity and low skills based in using RDF. These drawbacks therefore make the use of tabular data attractive in the context of the FALCON system.

The CSVW specifications address some of the deficiencies of existing the CSV format. Specifically it provides a standard metadata format, termed the CSV metadata vocabulary[25], that captures metadata associated with a table and a schema capturing the metadata for individual columns, including data type metadata, identification of primary key columns and reference to foreign keys in other CVSW conformant tables. This metadata can be captured in JSON or RDF. JSON, similar to CSV itself, is understood by tool and NLP developers and supported in a wide variety of programming languages (natively or via libraries), so this is the format selected for use in FALCON. However, it is also possible to map this into RDF so

[23] http://www.w3.org/2013/csvw/wiki/Main_Page
[24] http://www.w3.org/2013/csvw/wiki/Main_Page
[25] http://www.w3.org/TR/2015/WD-tabular-metadata-20150416/

that the metadata records can be aggregated into an RDF triple store and therefore subject to complex searches using the SPARQL query language.

The following principles were applied to the design of L3Data in CSVW format:

- Each table was typed as a PROV-O Entity and where applicable the table metadata referenced the Variable on the process Plan to which the Entity corresponded.
- Each table was typed as a DCAT Dataset, with the metadata including a Dublin Core title, description and language attribute.
- Each table was typed as a DCAT Distribution, with metadata for referencing a rights file in ODRL.
- The table delimiter was set to the tab character ("U+0009") rather than the default comma, to facilitate the inclusion of columns representing text.
- The metadata for each table included a PROV-O 'wasGeneratedBy' attribute indicating the activity that generated the resource in the table, a reference to the step from the process Plan to which that
- Table had a dedicate column representing the unique key for each row, thereby facilitating conversion of the tabular data to a graphical, RDF-based version.
- Separate Tables were allocated to data from different sources, namely source segments, suggested terms, suggested term definitions, suggested term translations, validate terms, validated term definitions and validated term translations.
- Separate tables were used for each open annotation relationship identified between other resources.

3.1. Web Service Integration

Figure 1 presents a message sequence diagram showing the interactions between the L3Data Mgr and Svr components and the other components in the systems. It is focussed on the assembly of an optimised guided post-editing order, and includes reference to two further message sequences already defined in deliverable D3.7.

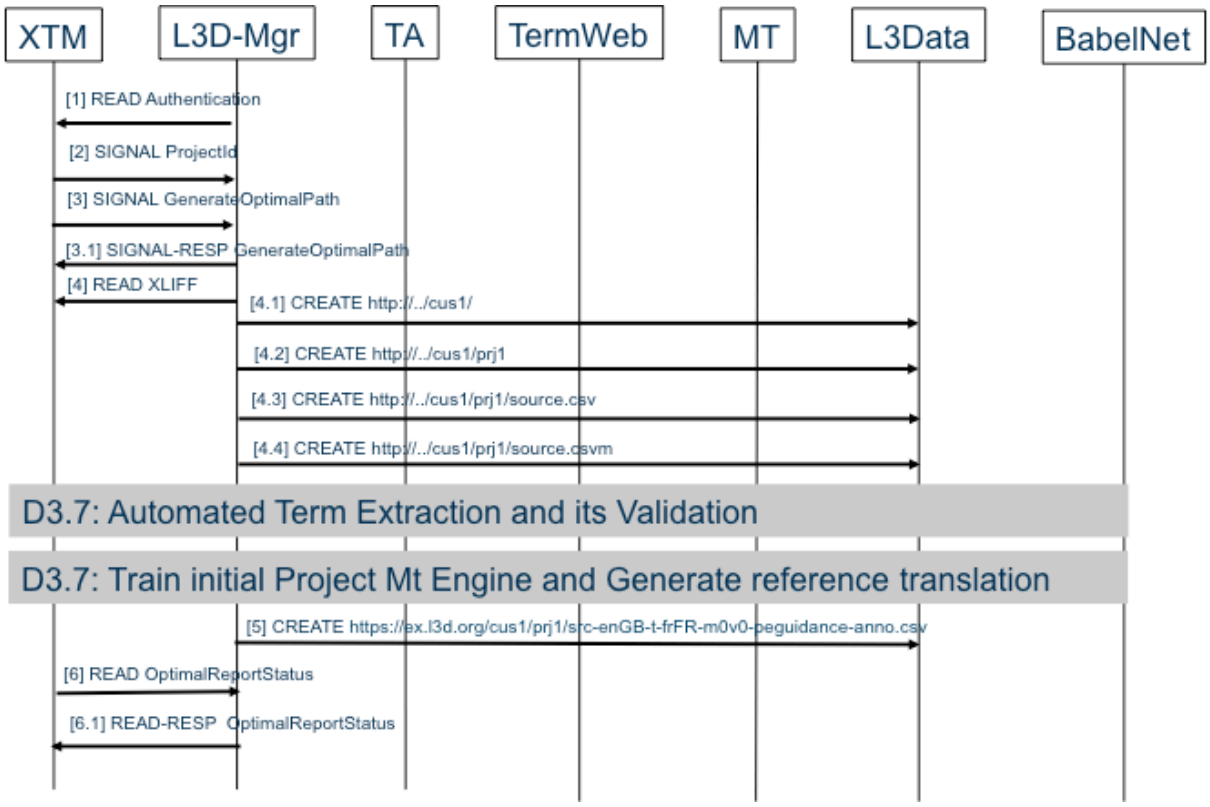

Figure 1: Message Sequence Diagram for L3Data components

ID	Message	Implementation
1	READ Authentication	Use Open Id API to access Open ID server in XTM and retrieve authentication key
2	SIGNAL ProjectID	XTM provide project metadata in term of end customer ID and project ID HTTP-POST-Request: ex.l3d.org/mgr/signal ?signal=project-metadata?cust=cus1?proj=prj1
3	SIGNAL GenerateOptimalPath	XTM requests that the optimal guided post-editing path is generated, providing link to source in XLIFF format HTTP-POST-REQUEST: ex.l3d.org/mgr/signal ?signal=generateOptimalPath {“xliffpath”:”http://falcon.xtm-intl.com/generatedfiles/47204825623956454353 4/file.xlf”, “engineID”:”engine0001”
3.1	SIGNAL-REPONSE GenerateOptimalPath	L3DMgr responds with ID to be polled for status of the optimal path generation HTTP-POST-RESPONSE: {“id”:”123456”}
4	READ XLIFF file	L3Data Mgr fetches the XLIFF file HTTP-GET-REQUEST: http://falcon.xtm-intl.com/generatedfiles/47204825623956454353 4/file.xlf
4.1	CREATE customer	L3Data Mgr creates end customer specific path in L3Data Svr if not already present HTTP-PUT-REQUEST: https://ex.l3d.org/cus1/

4.2	CREATE project	L3Data creates project specific path under end customer in L3Data Svr HTTP-PUT-REQUEST: https://ex.l3d.org/cus1/prj1
4.3 4.4	CREATE source segments	L3Data Mgr log the source segment: HTTP-PUT-Request: https://ex.l3d.org/cus1/prj1/source.csvm HTTP-PUT-Request: https://ex.l3d.org/cus1/prj1/source.csv CSV table with columns: • Source segment identifier • Source segment
5	CREATE post-editing priority	L3Data Mrg logs the result of the guided PE optimisation order to the L3Data Svr: HTTP-PUT-Request: https://ex.l3d.org/cus1/prj1/src-enGB-t-frFR-m0v0-peguidance-anno.csv HTTP-PUT-Request: https://ex.l3d.org/cus1/prj1/src-enGB-t-frFR-m0v0-peguidance-anno.csv CSV table with columns: • ID of annotation of a segment with post-editing priority • Reference to source segment • Score indicating the priority attached to post-editing the segment
6	READ OptimalReportStatus	XTM polls L3Data Mgr to check if the optimal guided post-edit order is available HTTP-GET-REQUEST: {“id”:”123456”}
6.1	READ-RESP OptimalReportStatus	XTM polls L3Data Mgr for results of guided PE optimisation, and if ready it returns the URI from where a text file with optimised order of segment IDs: HTTP-GET-RESPONSE: {“OptimalFilePathUri”:”http://ex.l3d.org/cus1/prj1/optimised-pe-order.txt”}

Table 3: L3Data component interactions

The full functionality of the L3Data Mgr and Svr component produces a set of CSVW files each with associated metadata files. The schema of these tables will be presented in deliverable D2.3.

4. SUMMARY

The L3Data component is summarised here as a profile of the linguistic linked data reference architecture suggested by the LIDER project.

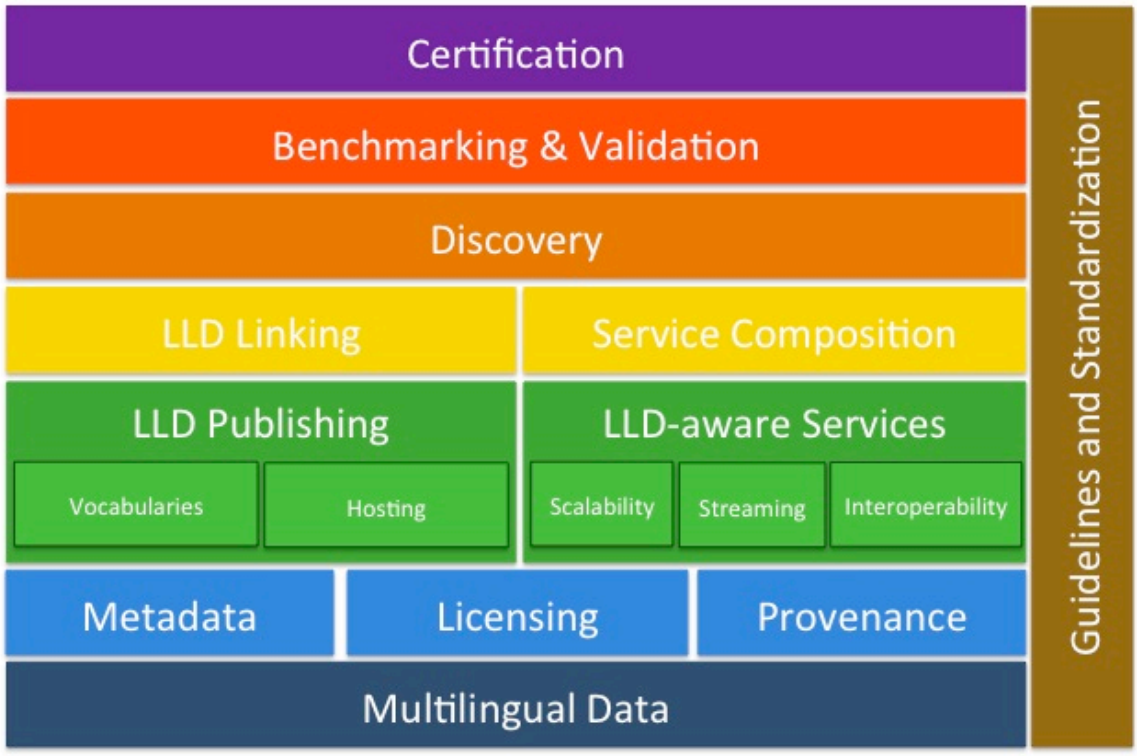

Figure 2: Overview of Linguistic Linked Data Reference Architecture from the LIDER Project

The reference architecture is structured as described in the table below, with the corresponding implementation in the FALCON system described alongside it.

LIDER Linguistic Linked Data Reference Architecture Elements	**Implementation of L3Data Architecture of the FALCON system**
Multilingual data, in all forms, modality and media types form substrate of the platform. Mappings to existing common data format such as XML vocabularies, JSON and Comma Separated Values ensure the major benefits of the platform are gained without costly transformation of existing data	Multilingual data consists I. Parallel text imported and generated within the project, the latter by both by machine and human translation activity. II. Terminology, consisting of term, their concept definitions and translation into project target languages CSV format is used for storing and publishing both, though import from TMX for parallel data and TBX for terminology is supported
Metadata: providing basic information about the dataset (author, language, structure), etc.	Metadata is associated with individual tables of multilingual data or open annotation data. In this category it consists of the DCAT attribute of title and description.
Licensing: specifying the terms and conditions of use of linguistic resources should be	Licensing information for all tables is recorded as reference from each table to an ORDL file, using the

specified. This includes the description of copyright information and any other rights-related restriction (e.g. privacy and data protection of personal data and commercial paywall access control where needed).	dct:license attribute in the table metadata. The right information addressed cover copyright and right provided by the EU database directive.
Provenance: describing the origin and processing history of data, which is key to assessing its usability in a specific task	All L3Data logged is a provenance Entity and associated Activity and Agent from the PROV-O vocabulary.
The Linguistic Linked Data (LLD) specific layers provide the integration point with the European Platforms for Language Technology Services and comprises of the following two layers:	
• **Linguistic Linked Data Publishing** consists of guidelines, best practices and standards describing how different types of resources (lexica, corpora, terminologies, lexico-semantic resources) should be transformed into RDF and how they should be published on the Linguistic Linked Open Data (LLOD) cloud. This layer also comprises of concrete tools and frameworks supporting transformation and publication as well as recommendation on use of common vocabularies and data hosting.	L3Data is published as tabular data according to the CSVW draft recommendation, with metadata recorded as JSON-LD according to the CSVW metadata format extended with other vocabularies described.
• **Linguistic Linked Data Linking**: This layer comprises of guidelines, best practices, tools and frameworks to supporting linking of resources as well as concrete tools and frameworks to support semi-automatic linking of resources, including managing links between resources with different access terms and conditions.	L3Data can be linked to other resrouces, either by referencing CSVW and metadata files or individual cell, row or column fragments of the CSV files using CSV fragment identifiers. The metadata and data can be mapped into an RDF representation based on the mapping of the JSON-LD vocabularies used into RDF. Within the L3Data model, linking between elements (i.e. tables) is performed using the Open Annotation model, where links are modelled as first class objects connecting annotated (target) and annotating (body) data.
For **LLD Services**, the following two layers are included:	
• LLOD-aware Services: Addresses	The L3Data Svr component offers a simple RESTful

content and knowledge analytics and processing services that can consume and produce Linguistic Linked Data. This includes:	interface for creating and reading L3Data in the form of CSVW files. This therefore represents a reusable form of LLOD-aware services that could be used with different configurations of process and by other components developed in other tool chains. In the FALCON system, this service is used to store and retrieve specific L3Data by the L3Data Mgr, the Text Analytics and the Machine Translation component. These components, however, do not themselves offer services that could be described LLOD-aware, instead they offer service designed to integrate with the commercial tools in the FALCON system.
○ **Scalability:** LLOD-aware services should be able to scale to processing large amounts of data. This requires a non-centralized architecture in which services can cache results and pass intermediate results to other services instead of relying on a client to coordinate and interact with all services implementing a complex workflow.	L3Data potentially consumes large volumes of parallel text for training machine translation engines, and can also generated large volumes of parallel text data as translation job are processes. This is addressed though the use of CSVW format for storing and publishing data to result in a more compact serialised publication format that is more compact than, for example, NIF, TMX or TBX, while still offering the referencing of data nodes needed for it to act as linked data. Though the implementation of the L3Data Svr component consists of a single HTTP server, the structure of the data allows the data to be readily distributed over different servers operated by the different actors in the value chain addressed in FALCON, namely the client, the language service provider, the translators subcontracted to the language service providers and the language technology service providers offering text analytics and machine translation services.
○ **Streaming:** The architecture relies on streaming principles to support the implementation of services that can process data in a stream fashion, thus reducing overhead of creating and closing connections, supporting real time analytics.	L3Data does not process stream data directly. The use of LT in the broader FALCON system does lend itself to the translation and terminology management of a stream of content update to a web site, though this would be operated as a series of translation projects for the same client, but with automated term extraction and machine translation components being progressively improved through retraining across the series of projects.
○ **Interoperability** focussed on use of common vocabularies to describe data inputs and	L3Data is based nearly entirely on existing open data vocabularies, either W3C recommendations such as PROV-O, DCAT, ITS2.0[26] or others such as

[26] http://www.w3.org/TR/its20/

output of services.	CSVW, Open Annotation, ODRL and P-PLAN[27] that are in the public domain and offered as open vocabularies.
• **Service Composition** for the chaining of single LLD-aware services to implement, monitor and optimise more complex workflows combining NLP, data management and human elements.	The L3Data architecture adopts a single core process flow that integrates the components involved. It does not therefore support explicit service composition in the formation of processing chains. However the chain implemented is captured explicitly using P-PLAN, so as to provide a service composition framework for analysing the provenance data that L3Data represents. This means that L3Data resulting from different process chains using similar components could be similarly modelled with ease. Generalisations of the process steps and entities are under analysis in the CNGL Centre for Global Intelligent Content, alongside analysis of other systems integrating LT components.
Discovery of LLD data set implemented by an arbitrary number of services that index and aggregate datasets and services and expose a standardized API that supports querying a repository to find datasets that meet certain criteria. The tools in the discovery layer should support SPARQL but provide Linked Data that is both understandable and searchable by both humans and machines.	While the L3Data CSVW format can be mapped into RDF and therefore subject to queries using SPARQL, the processed defined for the FALCON system does not rely on such a search. It is intended that a mapping from CSVW metadata format to data resource aggregation services such a DataHub[28] and LingHub[29] will enable published L3Data to publicly indexed and discovered.
Benchmarking and Validation to support the comparison of datasets and services to allow potential users to choose the service or dataset that best meets their needs using a common set of tools and quality definitions.	Currently the L3Data system offers access to portions of the EU's Directorate General of Translation's (DG-T)[30] ranslation Memory or EurVoc Terminology[31] data set as a benchmark and for use in new projects.
Certification to allow independent services and agents to assign quality labels or certificates to datasets if they meet specified conditions.	No certification is associated with L3Data at this point.
Guidelines & Standardisation is orthogonal to the above-mentioned layers and emphasizes that standardisation and promotion of uptake	The deliverables of the FALCON project will provide guidelines for the use of existing standards employed in the L3Data Schema. Feedback based

[27] http://www.opmw.org/model/p-plan/
[28] http://datahub.io/
[29] http://linghub.lider-project.eu/
[30] https://open-data.europa.eu/en/data/dataset/dgt-translation-memory
[31] https://open-data.europa.eu/en/data/dataset/eurovoc

by appropriate community initiatives (community groups, working groups) is crucial to ensure wide acceptance, implementation and use of LLD Platform.	on this implementation experience will be fed back to the active standardisation community as appropriate.

Table 4: Comparison of L3Data to the LIDER Linguistic Linked Data Reference Architecture

References

[Brummer] Martin Brümmer, Ciro Baron, Ivan Ermilov, Markus Freudenberg, Dimitris Kontokostas, and Sebastian Hellmann. 2014. DataID: towards semantically rich metadata for complex datasets. In Proceedings of the 10th International Conference on Semantic Systems (SEM '14), Harald Sack, Agata Filipowska, Jens Lehmann, and Sebastian Hellmann (Eds.). ACM, New York, NY, USA, 84-91. DOI=10.1145/2660517.2660538 http://doi.acm.org/10.1145/2660517.2660538

[D3.2] FALCON Deliverable D3.2: Initial L3Data Federation Platform Release

[Lewis] Dave Lewis, Rob Brennan, Leroy Finn, Dominic Jones, Alan Meehan, Declan O´Sullivan, Sebastian Hellmann, and Felix Sasaki, Global Intelligent Content: Active Curation of Language Resources using Linked Data, in The 9th edition of the Language Resources and Evaluation Conference, 26-31 May, Reykjavik, Iceland