LetsMT!

Platform for Online Sharing of Training Data and Building User Tailored MT

www.letsmt.eu/

Project no. 250456

Deliverable D1.2
Functional Specification

Version No. 1.0
31/08/2010

Document Information

Deliverable number:	D1.2
Deliverable title:	Functional Specification
Due date of deliverable according to DoW:	M6 – August, 2010
Actual submission date of deliverable:	August 31, 2010
Main Author(s):	Raivis Skadiņš (TILDE), Kārlis Goba (TILDE), Māris Puriņš (TILDE), Valters Šics (TILDE), Jörg Tiedemann (UUP), David Filip (MOR), Željko Agić (FFZG), Thomas Dohmen (SEM)
Participants:	
Reviewer	Philipp Koehn (UEDIN)
Workpackage:	WP1
Workpackage title:	LetsMT! platform and infrastructure
Workpackage leader:	TILDE
Dissemination Level:	PU
Version:	1.0
Keywords:	Machine translation, functional specification, user story, use case, LetsMT!, LetsMT

History of Versions

Version	Date	Status	Name of the Author (Partner)	Contributions	Description/Approval Level

EXECUTIVE SUMMARY

This document describes the functional requirements, vision of logical and infrastructure design of the LetsMT! system.

Table of Contents

Introduction

This document describes the functional requirements, and the vision of logical and infrastructure design of the LetsMT! system.

Preface

This document is based on the results of the previous project activities, e.g. deliverable D1.1 "Report on Requirements Analysis", D2.1 "Specification of data formats allowed" and the project's Annex I - "Description of Work" document.

Objectives

The aim of the Functional Specification document is to describe the function of the LetsMT! system and the high-level system design necessary to support further development and implementation of the new software.

The document will describe the functional and non-functional requirements for the LetsMT! system in detail sufficient for system architects, developers, and testers to proceed with design and development activities.

This document consists of two major parts. The first part (Chapter I) contains the conceptual design and user stories/requirements for the system. The second part (Chapter II) describes the logical and infrastructure design of the system.

The described functionality is within the scope and expected to be within the budget of the LetsMT! project, however due to the innovative nature of the project, certain functional or non-functional requirements may be modified, removed from or added later to the project scope. The user interface described in this document is conceptual and may differ from the final implementation.

1 Conceptual Design

This chapter is mostly based on the project deliverable D1.1 “Requirements analysis” findings and the experience of the LetsMT! Consortium members.

1.1 System overview

In recent years statistical machine translation (SMT) has become a major breakthrough in machine translation (MT) development providing a cost effective and fast way to build MT systems. This development was particularly facilitated by the open-source corpus alignment tool GIZA++ and the MT training and decoding tool Moses. Another factor for facilitating the development of MT for many languages was the EU translation corpus and other parallel data available on the Internet. The EuroMatrix project has demonstrated how open source tools and publicly available data can be used to generate SMT systems for all language pairs of the official EU languages.

However, these achievements do not fulfil all expectations regarding the application of available SMT methods. The quality of an SMT system largely depends on the size of training data. Obviously, the majority of parallel data is in widely-used languages (e.g. English, German and some others). As a result, SMT systems for these languages are of much better quality compared to systems for under-resourced languages, i.e. languages with scarce linguistic resources. Current systems are built on the data accessible on the web, but it is just a fraction of all parallel texts. The majority of valuable parallel texts still reside in the local systems of different corporations, public and private institutions, and desktops of individual users.

Another obstacle preventing wider use of MT is its general nature. Although free web translators provide reasonable quality for many language pairs, they perform poorly for domain and user-specific texts. Current free systems cannot be adjusted for particular terminology and style requirements. Large international corporations contract MT companies like Language Weaver to adapt translation systems for their particular needs. However, this costly process is not accessible to smaller companies or the majority of public institutions. This prevents a large segment of the EU population from using existing MT solutions to get access to online information.

Specifically regarding application in the localization and translation industry, a huge number of parallel texts in a variety of industry formats have been accumulated, but the application of this data does not fully utilize the benefits of the modern MT technology. At the same time, this industry is experiencing growing pressure to increase efficiency and performance, especially due to the fact that the volume of texts to be translated grows at a higher rate than the availability of human translation, and translation results are expected in real-time. At present, the integration of MT in localization services is in its early stages, and the cost of developing specialized MT solutions is prohibitive to most players in the localization and translation industry. The quality of the generic MT offerings provided for free is too low to reap any efficiency gains in the professional localization industry setting. The same problem is faced by online information providers. They provide information mostly in the larger languages because the cost of human translation into smaller languages is prohibitively high and the quality of existing MT solutions for smaller languages is unsatisfactory.

To fully exploit the huge potential of existing open SMT technologies and the huge potential of user-provided content, we propose to build an innovative online platform for data sharing and MT building. The LetsMT! project will extend the use of existing state-of-the-art SMT methods enabling users to participate in data collection and MT customization to increase the quality, scope, and language coverage of MT. Currently LetsMT! is creating a cloud-based platform that gathers public and user-provided MT training data and generates multiple MT systems by combining and prioritizing this data.

LetsMT! services for translating texts will be available in several ways: through the web portal, through a widget provided for free inclusion in a web-page, through browser plug-ins, and through integration in computer-assisted translation (CAT) tools and different online and offline applications. Localisation and translation industry business and translation professionals will be able to use the LetsMT! platform for uploading their parallel corpora in the LetsMT! website, building custom SMT solutions from the specified collections of training data, and accessing these solutions in their productivity environments (typically, various CAT tools).

1.1.1 System User Groups

LetsMT! system is targeted to support specific organizations and user groups needs for statistical machine translation. These user groups and their respective needs to be addressed by LetsMT! are summarized in the table below.

User group	Main needs
Individual Internet users	• Free MT for under-resourced languages or specific domain • Access to multilingual news in business and finance
Localization and translation industry companies	• Increase in productivity of translation work through application of translation tools and MT • MT adapted to terminology and stylistic requirements of the particular project
Web developers	• Provide access to websites for multilingual user community using widgets
University education and research community	• Easy-to-use infrastructure for training and research on SMT • Parallel corpus data, especially for under-resourced languages and domains • Easy to use infrastructure for experiments in SMT training on different data • Parallel corpus sharing platform
Translation automation solution developers and providers	• Integration of new MT directions into translation automation tools and solutions

Table 1. Main needs of LetsMT! user groups

1.1.2 User Characteristics

During interviews, conducted in order to gather user base requirements, it was discovered that different types of users have different requirements for the LetsMT! platform. A summary of the different user types within the different user groups is outlined in the table below.

User type	Description
Individual Internet users	
Anonymous Internet User	A person browsing the Web; who wants to test SMT or hopes to receive better machine translation than provided by other publicly available MT systems like Google Translate. This user does not want to make an effort, just use the service to either translate a desired text or to compare with other SMT engines. This user is not expected to contribute corpora
Business news reader	Wants to read international financial news in "smaller" native EU languages. Foreign language skill is limited; therefore LetsMT! is used to gist English news in local language and vice versa. Requirements for translation quality are not very high, however it is expected that terminology and the essence of news will be translated correctly and understandably. This user is not expected to contribute corpora.
MT enthusiast	A person who is aware of various available MT tools and technologies. Attraction to LetsMT! is based on the possibility to directly influence the SMT engine and review the SMT system training and translation logs. This MT enthusiast is ready to contribute corpora in order to achieve better SMT results. Receiving praise for contributing or belonging to some elite group would be considered a bonus. Might submit poor quality comparable corpora.
Localization and translation industry company	
Translator	Needs to use SMT because of organization's workflow. May be very skeptical to MT results. Resistant to change. Wants to use SMT seamlessly integrated in daily routines. CAT tool integration is preferred. Very simple and quick on-line tool could be acceptable that supports file formats used in translation projects.
Translation Project Manager	Wants to reduce translation project cost by reducing time translators spend on initial translation. Would be ready to contribute high-quality corpora/translation memories in order to improve SMT system quality. Needs to have good control and quality measurements of trained SMT systems.

User type	Description
	Different clients have different domains, vocabularies, tools, translation styles. Thus a single SMT system does not provide the necessary quality of translation and post-editing of machine translated text takes the same or even longer time. Will need lots of specific SMT systems for each project type or customer. Would use only a few SMT systems simultaneously, but needs lots of historical data to build SMT systems on-demand whenever specific project starts.
Localization Company Manager	Have high quality corpora which are most likely protected by IPR or confidentiality agreements. Highly aware of IPR and the need to protect organization-specific knowledge (competitive advantage). SMT usage in current translation workflow and tools must be cost and time efficient. Currently maintenance and even evaluation of SMT feasibility is very expensive due to lack of knowledge of technologies involved and infrastructure requirements.
Web developers	
Web developer	Web developer who needs to create multi-lingual sites, but does not have necessary resources to provide high quality translation. Commonly will use LetsMT! for prototyping of web sites in different languages. If the quality of the translation is accepted by the client in general, the web developer could provide minor fixes and improvements of translation.
University education and research community	
Researcher	Member of educational and research organization or translation and localization industry organization investigating options available in SMT field. Most interested in quality of translation and possibilities to control SMT system training. Has some parallel and monolingual corpora available and is interested in improvements in translation quality after corpora submission. This user will want to tweak every possible SMT system option and compare results of these tweaks. Could be a decision maker or contribute to decision making about use of LetsMT! services.
Research organization leader	Owns large amounts of mono and bi-lingual corpora. Is interested in theoretical aspects of results of SMT systems and how different input data influence quality of translations. Research organizations most likely will have many different SMT systems and will re-train SMT systems frequently.

User type	Description
	Would benefit from using LetsMT! platform as no investment in infrastructure is required. Funding possibilities are limited.
Translation automation solution developers and providers	
Translation Solution Product manager	As user community constantly requires MT solution integration solution developer would benefit from integration of LetsMT! platform into their products/solutions. Also many competitors have introduced MT modules in their solutions. Installation and maintenance of MT requires infrastructure and skilled specialists in MT are scarce. Even access to public SMT systems through products could be seen beneficial to customers as trial of MT integration. More user-tailored solutions could be sold as a separate service.

Table 2. LetsMT! user characteristics

Major requirements and challenges encountered in currently available MT systems that are planned to be addressed by LetsMT! are listed in table below for each user group.

User type	LetsMT! requirement	Challenges
Anonymous Internet user	• Free! • Provide good enough translation to understand essence of foreign language texts	• Current SMT results are hard to comprehend due to poor quality and broad domain of translation text
Business news reader	• Good quality (better than Google) of multi-lingual financial news translations	• MT financial information hard to understand in under-resourced languages
MT enthusiast	• Can upload mono or bilingual corpora for improvement of SMT quality thus influencing quality of translation • Uploaded corpora quality should be assessed	• Very limited influence on quality of publicly available SMT systems
Translator	• Easy to use • Incorporated in workflow or CAT tool • Fast to translate • Quality of translation is critical • Utilization of already available high quality translation memory	• No integration with CAT tools / workflow of current SMT solutions • Poor translation quality of SMT systems • Hard to distinguish translations from TM (trusted) and SMT (not trusted) in current translation workflows

User type	LetsMT! requirement	Challenges
Translation project manager	• Integration in current workflows/practices/tools • Control over corpora included in SMT system training • Single point of corpora/TM storage to simplify management • On-demand SMT system training • Frequent (incremental) re-training of SMT systems	• Resistance of Translators to use SMT • Complicated maintenance of different SMT systems • Need to establish new process to manage use of SMT
Localization Company Manager	• Safeguard IPR and competitive advantage • Integration in current localization and translation practices/tools • Reduction of overall cost and time of project execution	• Maintenance overhead of storing various translation memory and corpora artifacts • Project ramp-up time and context switching for translators too long • Creation of MT infrastructure is expensive
Web developer	• Widget for automatic web-page translation	• No simple solution to translate a web page to foreign language that is usable at least for prototyping purposes
Researcher	• Support for many different SMT systems • Possibility to re-train models often • Detailed reporting	• Very little control over SMT system training translation process OR it's very complicated • No details about SMT system training or translation process
Research Organization Leader	• Possibility to upload large amounts of corpora • Support for many SMT systems	• Hard to maintain infrastructure for corpora storage and many SMT system application in research • Creation of MT infrastructure is expensive
Translation Solution Product Manager	• Open API	• No need to develop and maintain SMT systems

User type	LetsMT! requirement	Challenges
	• Stability / availability • Available trained SMT systems "out of the box"	infrastructure for integration into tools/services provided • Insufficient corpora available for SMT system training

Table 3. LetsMT! requirements summary by user types

1.2 Typical Usage Scenarios

The main usage scenarios which are relevant to define the functionality of the LetsMT! system are described in this section.

1.2.1 General Use Scenario

This scenario describes general use of the LetsMT! system by various user types to acquire translation from user tailored SMT system.

There are a number of freely available translation systems on the Internet; however none of those allow users the options to choose between domain–specific systems. Also users cannot directly influence SMT systems by uploading corpora in user's possession for achieving better quality of translations for their needs. LetsMT! will address these shortcomings.

Users will be able to upload corpora into the system, using a widely accessible client application, for example, by using a web page interface. Users will be provided with an effective way of searching, navigating and selecting a trained SMT system to use. And, of course, translate texts using one of available LetsMT! trained SMT systems.

1.2.2 Localization Use Scenario

This scenario describes a translation project execution processes employed in localization industry companies. The process is summarized in the following diagram:

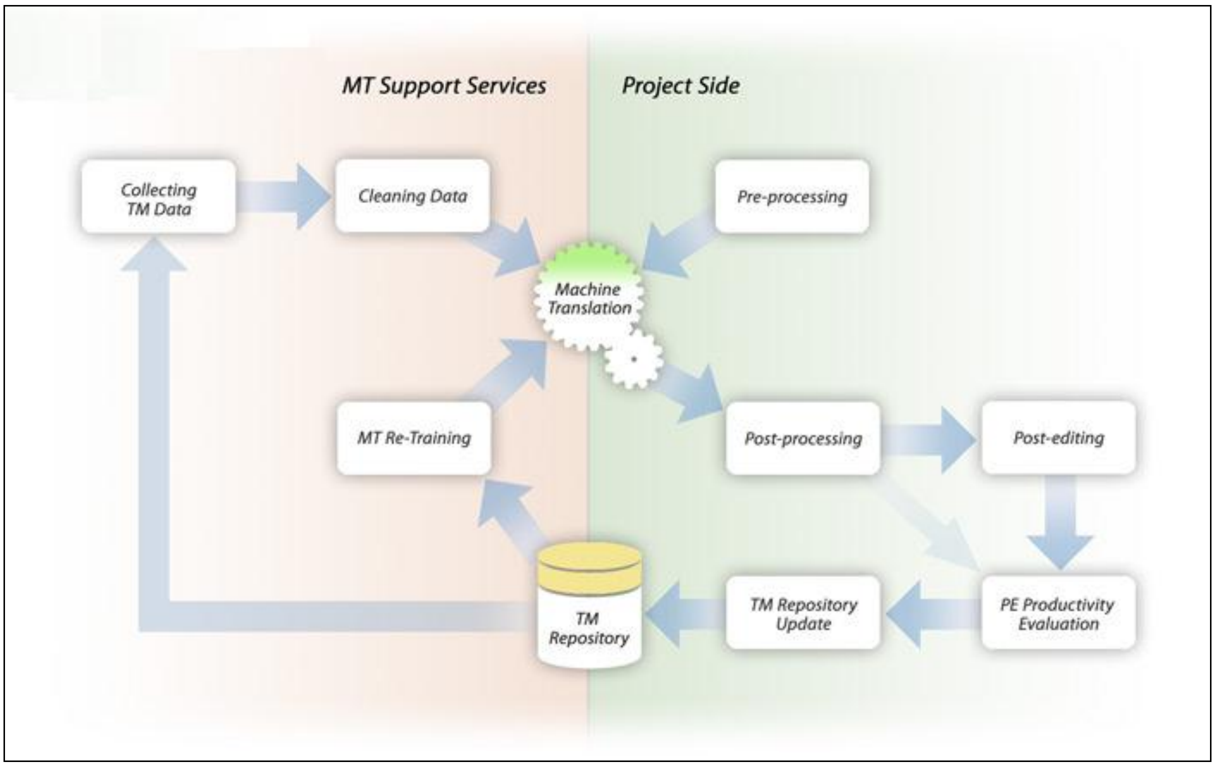


Figure 1. Localization process diagram

In localization (L10N) TM (CAT) technology has been ubiquitous since the late 20th century. Hence, unlike other industries, L10N typically has well aligned bilingual legacy corpora. However, due to contractual obligations that are common in L10N confidentiality is frequently an issue. TMs are often classified as confidential materials not to be used for purposes other than for translating the work the client has contracted. Functionally, LetsMT! Must be able to track the legal metadata of the training materials.

L10N data typically contain significant amounts of code. The code is of two varieties meta-segment and in-line. Whereas meta-segment mark-up is easily filtered out the in-line elements constitute a cleaning challenge. Therefore we cannot just filter out placeholders. In case no working substitution algorithm is found these segments must be thrown away not to pollute the training corpus.

MT in L10N can only be successful if highly specific engines are developed. The MT training capability should be exposed to a technically skilled end user (production coordinator in a L10N company or department) through a simple API and GUI. Advanced meta-data management, including legal is essential in order to select appropriate training data for models. The LetsMT! system must support the management of trained engines and training data and the organization of these by means of various meta-data.

Production pre-processing must be able to recognize factoids of source formats and process them according to strict rules. It must strip or replace all mark-up but store it to attempt reapplication during post-processing.

Major automated evaluation metrics – BLEU, TER (including language specific where available), METEOR (including language specific where available) should be built in or at least integrated as quick and reliable third party web services. These metrics must be available in real time for samples of up to 100K words to allow for hill-climbing monitoring during training. The end-user MT trainer should be able to rapidly assess the efficiency of added training data with respect to the test set.

You can only achieve good enough quality in raw-output-publishing scenarios through human feedback, i.e. post-editing and incremental retraining.

In human quality publishing scenarios integration with CAT tools is a must, because only segments that do not have good TM matches (based on a configurable threshold) are typically sent to MT. Translator/Post-editor typically edits the TM and MT suggestions at the same time. Therefore it is critical for post-editing scenarios to integrate MT suggestions in major CAT tools, such as Trados, Worldserver, etc. Post-edited translations should be automatically stored in a TM repository (ideally as increment to the original training data) to be used from time to time for MT retraining. Ideally, the end-quality post-edited strings should feedback directly into the MT's retraining capability.

1.2.3 Financial News Use Scenario

The LetsMT! consortium has identified that press releases are an ideal source for LetsMT! data, specifically for the online MT service of business and financial news. International press releases are almost invariably distributed in English to the international business community. However, as many companies registered in non-native English countries are required to release important business news in their native language, a large percentage of business press releases are available in two languages.

For the majority of cases, the translation was done by the company themselves, before submitting the press release to the news provider. In this case either the company dissemination (their website) or the local national press release agency provides the original language press release. Listed companies normally provide an overview of their press releases from their corporate websites. National press agencies also support access to the releases made through their service. Full texts are available only for a fee to registered users. In addition also archived data is commercially available. As timing is critical for press release data, the time stamp of the message is a reliable indicator by which the original language document can be matched to the English translation.

1.2.4 Upload and Sharing of Corpora Use Scenario

Generally organizations are relatively willing to and interested in sharing their text resources via the LetsMT! platform. Of all interviewees 44% are to some extent willing to share data even if half of them are reluctant, mostly because of IPR issues. Only 16% of all interviewees dismiss the sharing idea. Collaboration with organizations that have a legal framework of TM sharing is highly advisable.

Another aspect closely related to the organizations' ability to share data, is IPR. This means that a larger subset of the organizations might be able to share data if they can resolve the IPR issues with their customers or clients.

Many different file formats are relevant in connection with upload data. The most popular file format mentioned is TMX (46%), but also Microsoft Word (15%) and Adobe PDF (10%) are popular. The most widely used formats for data collection in Localization scenario are TMX and XLIFF. Support for TBX, csv, and tab-delimited are important for collecting terminology data. The system will have to support large volumes of corpora (mono and bilingual). Files to upload may be of large (several GBs) thus the system must support uploading such data volumes over the Internet and resuming interrupted operations due to network connection interruptions.

Metadata on corpora uploaded in the LetsMT! system can be used to select the most appropriate training material for MT training. The metadata categories listed are the minimum that must be with a possibility to extend meta-data collection to meet specific needs.

- Language pair
- Source language
- Domain
- Text type
- Data owner
- Data provider
- Upload date
- Text production timestamp
- Alignment type

1.3 User stories

1.3.1 Public Website for Translation

One of LetsMT! system features is to provide possibility to try different SMT systems in order to increase awareness of LetsMT! services.

R.1. User selects one of the public LetsMT! systems and enters the text for translation in the text box. After pressing the *Let's Machine-Translate* button, the translation of a text is provided to the user in the *Translation* text box.

R.2. Alternatively the user may enter a URL into the *Text or URL to translate* text box and in that case the system downloads the respective content and provides the translation of a web page in a new browser window or tab.

R.3. If only one word or term is entered, the system may provide a translation from a terminology database, vocabulary or n-best list, depending on the available data in the selected SMT system.

R.4. For system availability purposes translation text is limited to 10Kbytes. Also no more than one request may be issued from a single IP simultaneously.

R.5. Translation system updates the translated text area as soon as portions of text are available from the SMT system (see http://translate.tilde.lv/?lang=en for more details).

R.6. Progress bar will show activity/progress (see http://translate.tilde.lv/?lang=en for more details).

R.7. (optional) Synchronous scrolling of original/translated text areas will be implemented.

R.8. (optional) Tooltip or original text will be highlighted when moving a mouse over the translated text.

1.3.1.1 Pre-conditions and scope

The User has selected a trained SMT system and navigated to the respective system's public translation page.

The User needs to have plain text or a URL to translate.

1.3.1.2 Security

Public SMT systems are available publicly from the Internet to everyone.

1.3.1.3 Post-conditions

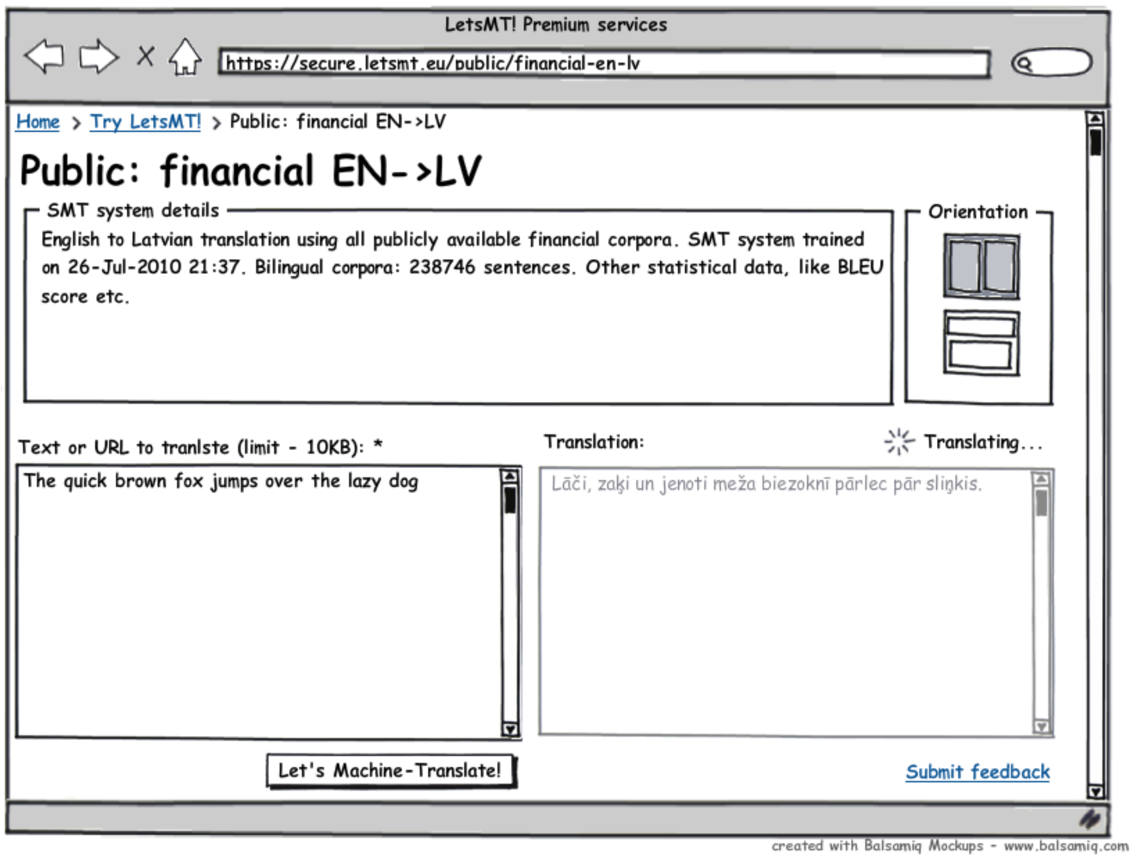
A translation will be provided to the user in a *translation* text box or an error message is displayed stating information about the problem encountered:

- System error (problem with translation engine/LetsMT!)
- Too many requests from the same IP

1.3.1.4 Performance requirements

The Translation should be provided "as-you-type", if system performance allows.

1.3.1.5 UI Mockup


LetsMT! Premium services
https://secure.letsmt.eu/public/financial-en-lv
Home > Try LetsMT! > Public: financial EN->LV
Orientation
Translating...
Let's Machine-Translate!
Submit feedback
created with Balsamiq Mockups - www.balsamiq.com

Public: financial EN->LV

SMT system details
English to Latvian translation using all publicly available financial corpora. SMT system trained on 26-Jul-2010 21:37. Bilingual corpora: 238746 sentences. Other statistical data, like BLEU score etc.

Text or URL to tranlste (limit - 10KB): *
The quick brown fox jumps over the lazy dog

Translation:
Lāči, zaķi un jenoti meža biezoknī pārlec pār sliņkis.

Figure 2. Public translation web page mock-up

1.3.2 SMT system selection

SMT system selection user story describes how users can select a SMT system from the available systems list. Due to the high granularity of SMT systems (e.g. for every customer/domain) the number of such systems may be very high, thus an efficient navigation mechanism must be provided for users to select the correct SMT system from the whole array of available systems.

When a user wants to translate a document, text, URL or perform other actions on a system he/she first has to locate the translation system to use by navigating to the "SMT systems list" page.

R.1. List of systems only includes systems which the user can access (based on LetsMT! service subscriber/group/role membership).

R.2. Possibility to filter by

a. source/target language
b. domain
c. title, description or other meta-data
d. PUBLIC attribute

R.3. "[x] include public systems" value must be persisted to user profile

R.4. Status of a system must be shown to user in systems list

a. Green – system trained, loaded and ready for translation
b. Orange – system trained, has to be loaded before use
c. Red – system not trained yet or training in progress
d. Grey – system training not scheduled yet

R.5. When the user selects a system, the following info appears:

a. Details of a system (with all meta-data)
b. Statistics of sources, corpora, usage
c. Quality metrics (automatic - BLEU, user feedback etc.) in a timescale graph

R.6. Available system functions/actions are displayed based on access rights of a user:

a. Translation of text
b. Translation of files
c. API access information/link to CAT tool configuration info

1.3.2.1 Pre-conditions and scope

User has logged into system or in case of an anonymous user only public SMT systems are available.

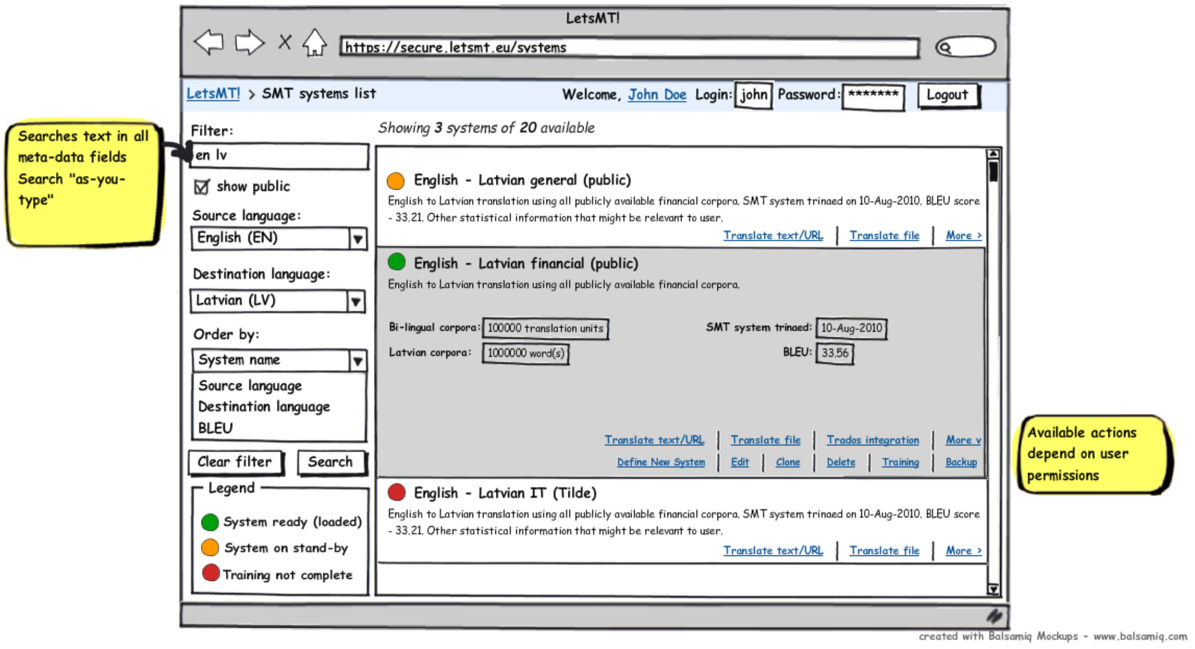
1.3.2.2 Security

- Anonymous and registered users see only public SMT systems
- Professional authenticated users see public systems and service subscriber systems that they are granted access rights to use.
- Users see no systems of other LetsMT! service subscribers.

1.3.2.3 Post-conditions

- Details and statistics of selected system are shown to user.
- User is taken to respective tool page of specific SMT system for further operations – file/text translation/CAT tool usage description.

1.3.2.4 UI mockup


LetsMT!
https://secure.letsmt.eu/systems
LetsMT! > SMT systems list
Welcome, John Doe Login: john Password: ******* Logout
Searches text in all meta-data fields Search "as-you-type"
Filter: en lv
show public
Source language: English (EN)
Destination language: Latvian (LV)
Order by: System name
Source language
Destination language
BLEU
Clear filter
Search
Legend
System ready (loaded)
System on stand-by
Training not complete
Showing 3 systems of 20 available
Translate text/URL | Translate file | More >
Translate text/URL | Translate file | Trados integration | More v
Define New System | Edit | Clone | Delete | Training | Backup
Available actions depend on user permissions
created with Balsamiq Mockups - www.balsamiq.com

English - Latvian general (public)

English to Latvian translation using all publicly available financial corpora. SMT system trinaed on 10-Aug-2010. BLEU score - 33.21. Other statistical information that might be relevant to user.

English - Latvian financial (public)

English to Latvian translation using all publicly available financial corpora.

Bi-lingual corpora: 100000 translation units
Latvian corpora: 1000000 word(s)
SMT system trinaed: 10-Aug-2010
BLEU: 33.56

English - Latvian IT (Tilde)

English to Latvian translation using all publicly available financial corpora. SMT system trinaed on 10-Aug-2010. BLEU score - 33.21. Other statistical information that might be relevant to user.

Figure 3. SMT system selection screen mock-up

1.3.3 Website for File Translation

This user story describes the envisioned *Translator* interaction with the system during the translation process as described in section 1.2.2 LOCALIZATION USE SCENARIO (PAGE 12) of this document.

During the translation process the *Translator* might start with machine-translation for a document and then post-edit it. It is expected that MT should provide good enough results for accurate translations. The story describes the interaction with the LetsMT! system in order to obtain a machine-translation for a document in one of the supported formats.

R.1. User uploads a file to translate via web interface

R.2. Detailed report is generated for user after translation is finished with output from each translation step.

R.3. Progress bar is shown to user during translation process. Progress bar is updated reflecting different steps of translation process.

R.4. Translated text is available as file download when translation is complete in correct encoding etc.

R.5. Error is displayed if:

a. System error encountered

b. File filters where unable to detect and parse correct file type/structure

c. Validators, preprocessors or postprocessors aborted with critical errors during text processing.

R.6. User can stop processing of current translation task by pressing “Cancel current operation” button

R.7. User can download log file for translation process (also in case translation was aborted by user).

1.3.3.1 Pre-conditions and scope

User needs to have a file for translation in one of the supported formats.

User already has selected a trained SMT system.

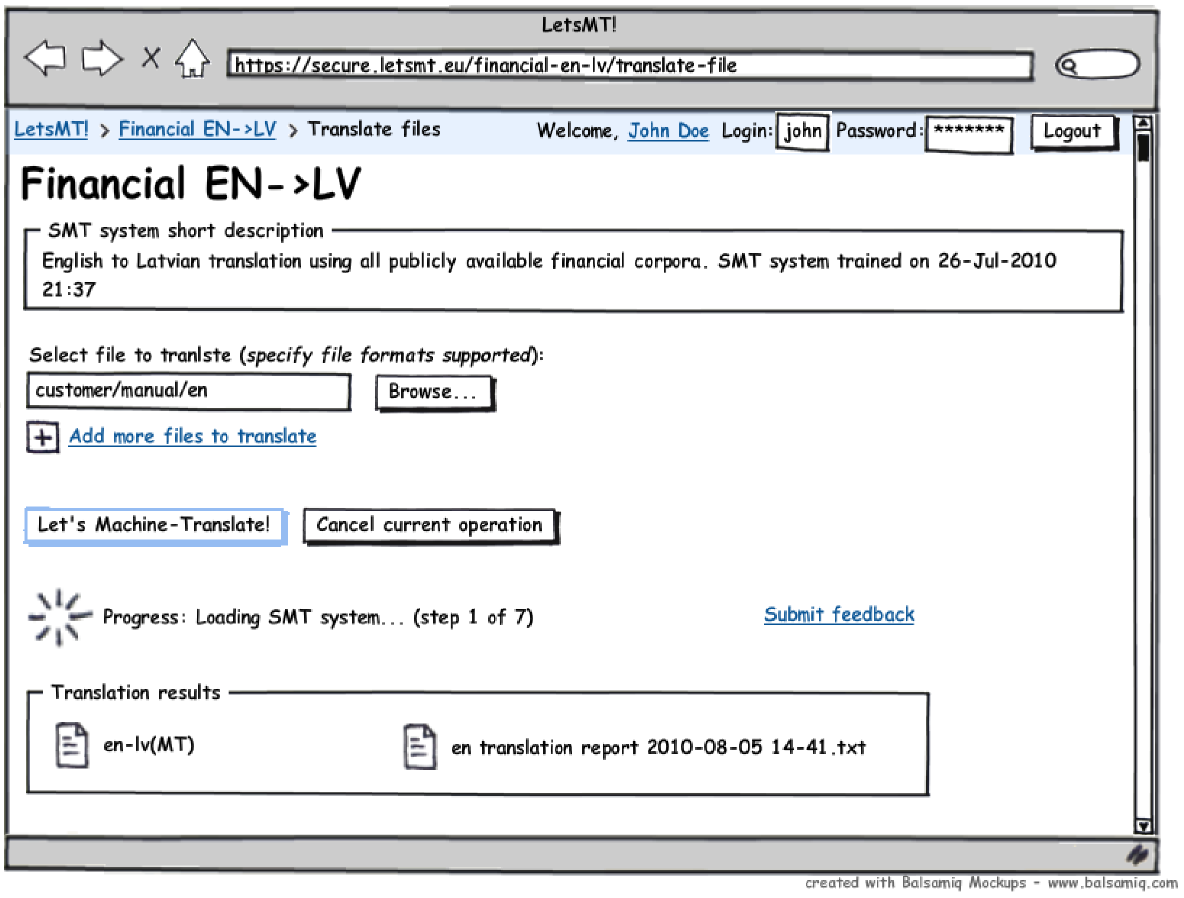
1.3.3.2 Security

This feature is accessible only after logging into system for all types of professional users (*LetsMT! service subscriber Administrator*, *Registered Power User* and *Registered User*) based on LetsMT! service subscriber and permissions.

1.3.3.3 Post-conditions

- A translation and translation steps report provided to user or error message is displayed about problem encountered during translation.
- User interface is updated during translation process reflecting progress of translation.

1.3.3.4 UI Mockup


LetsMT!

https://secure.letsmt.eu/financial-en-lv/translate-file

LetsMT! > Financial EN->LV > Translate files   Welcome, John Doe Login: john Password: ******* Logout

Financial EN->LV

SMT system short description

English to Latvian translation using all publicly available financial corpora. SMT system trained on 26-Jul-2010 21:37

Select file to tranlste (*specify file formats supported*):

customer/manual/en   Browse...

Add more files to translate

Let's Machine-Translate!   Cancel current operation

Progress: Loading SMT system... (step 1 of 7)   Submit feedback

Translation results

en-lv(MT)   en translation report 2010-08-05 14-41.txt

created with Balsamiq Mockups - www.balsamiq.com

Figure 4. File translation screen mock-up

1.3.4 Upload of Parallel Corpora (TMX format)

The User uploads a collection of translation memory files in TMX format:

- User has one or several TMX files laying around
- User puts all files together into one ZIP file

After corpus upload:

- Uncompress TMX files from ZIP
- Check validity (XML, encoding, ...)
- Language check (can be difficult; use lang-attributes in TMX and check if language is correct)
- Show user MD5 checksum of uploaded file

1.3.4.1 Pre-conditions and scope

- User knows how to create TMX files (for example by using Trados export)
- User knows how to compress data files in ZIP archive (optional)
- User has to specify (or create) a corpus into which the data has to be inserted (a corpus for us may consist of many TM's and many language pairs!)
- User finds the upload page
- User provides all meta-data (and understands what he/she has to fill in)

1.3.4.2 Post-conditions

- Corpus is converted to internal LetsMT! format
- Database updated with meta-data and establish authorization of uploaded content
- Set status and make data available according to authorization

1.3.4.3 Security

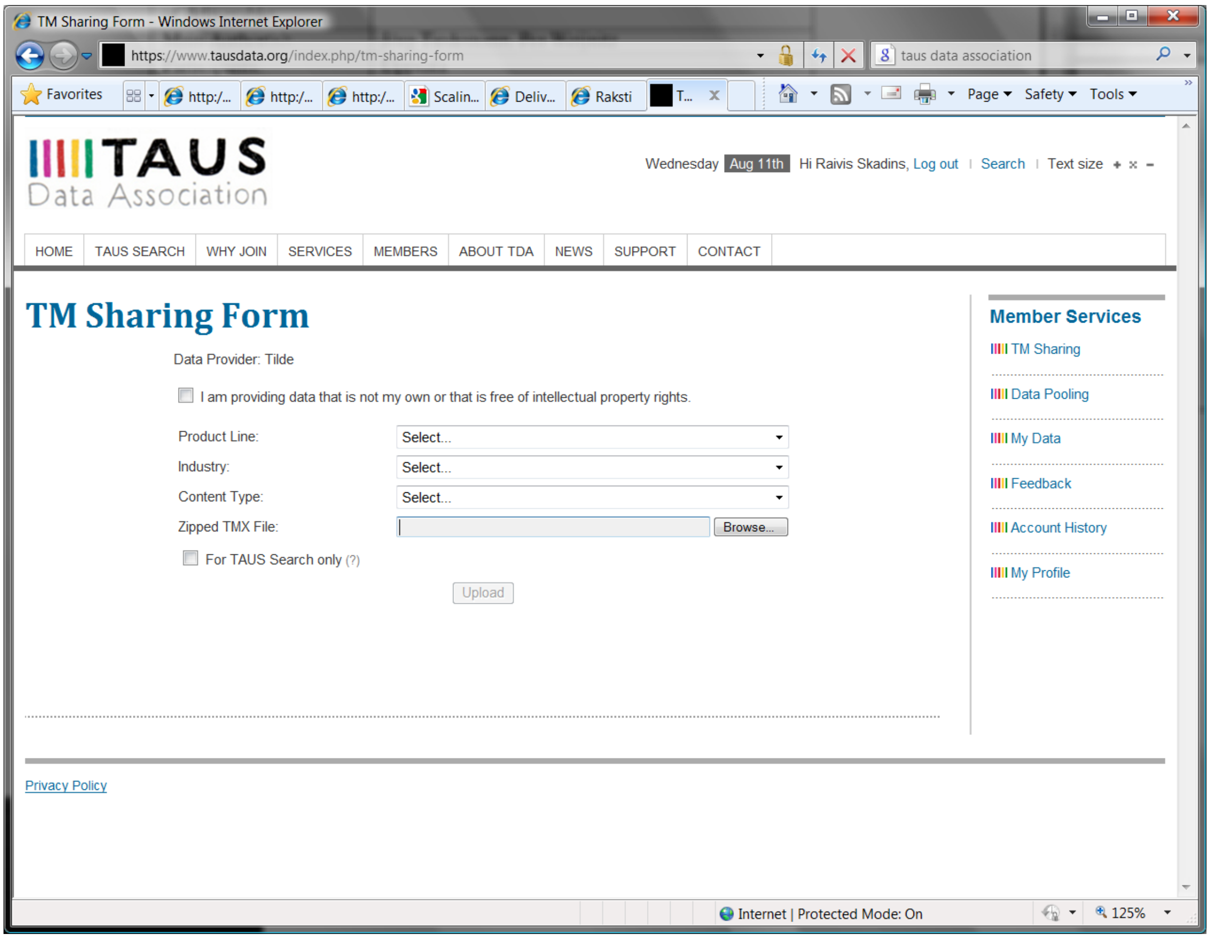
- File names with special characters may disorganize file system or destroy pre-processing pipelines
  - normalize names in some way (but be careful about name collisions!)
- Uploads are always sensitive to viruses and other types of attacks
  - system must verify and validate content of each file uploaded from security standpoint

1.3.4.4 Performance requirements

- Fast upload of up to 100MB compressed data (or more)
- Robust handling of possibly corrupt data
- Progress bar for upload (optional)

1.3.4.5 UI mockup

Upload form should be similar to the TAUS data upload form (see below) + changes according to LetsMT! requirements of meta-data.


TM Sharing Form - Windows Internet Explorer

https://www.tausdata.org/index.php/tm-sharing-form

TAUS Data Association

Wednesday Aug 11th Hi Raivis Skadins, Log out | Search | Text size

HOME | TAUS SEARCH | WHY JOIN | SERVICES | MEMBERS | ABOUT TDA | NEWS | SUPPORT | CONTACT

TM Sharing Form

Data Provider: Tilde

☐ I am providing data that is not my own or that is free of intellectual property rights.

Product Line:	Select...
Industry:	Select...
Content Type:	Select...
Zipped TMX File:	Browse...

☐ For TAUS Search only (?)

Upload

Member Services

- TM Sharing
- Data Pooling
- My Data
- Feedback
- Account History
- My Profile

Privacy Policy

Internet | Protected Mode: On 125%

Figure 5. TDA corpora upload screen

1.3.5 Upload of Parallel Corpora (DOC/DOCX format)

This story describes the case when the user wants to upload a collection of parallel documents in DOC/DOCX format.

- Create zip archives with all doc-files (one archive per language)
- Upload documents in one form (let the user add additional uploads/languages)
- User needs to specify languages, file formats, meta-data ...
- User needs to specify document linking map

Possible allowed mappings:

- Docs with same names are linked
- Docs with similar names except for lang-id-suffix

Provide User with import and alignment log to accept result and proceed with import into LetsMT! internal structures.

Optionally provide user with manual adjustments and re-alignment (different parameters, different alignment approaches, manual interventions, ...)

1.3.5.1 Pre-conditions and scope

- User knows how to zip doc files together
- User produces one zip file per language
- Documents have to include COMPLETE translations

1.3.5.2 Post-conditions

- Contents from each document are extracted and converted to LetsMT! format with sentence align each parallel document.
- Metadata in database is updated.

1.3.5.3 UI

See "Upload of Parallel Corpora User Story (TMX format)" UI mockup section. Field mapping and import result review screens will be defined later during project execution.

1.3.6 Translation Widget

One of the planned features for the LetsMT! platform is the translation widget – an open-source module to be integrated into client websites in order to provide multilingual web features to these websites. The translation widget is planned as a demo-feature to illustrate the capabilities of the LetsMT! platform by using the LetsMT! web-service (API) as a back-end.

The translation widget will be implemented at least in the following web sites (see LetsMT! Definition of Work page 69):

- http://www.thelocal.se/news/0
- http://www.dutchnews.nl
- http://www.praguepost.com/business
- http://www.thenews.pl/business
- http://blogs.wsj.com/new-europe

The translation widget is basically a Javascript module (although other programming languages may also be supported in the future) that a web developer places within website to provide translation services. The module itself accesses the data on the website, feeds it to the LetsMT! translation web-service (API) and displays translation to the user. Therefore, there are two basic user profiles or user classes for the translation widget: the web developer and the end user. Following are the user stories for these user profiles.

Web developer:

R.1. The user accesses the LetsMT! public website and downloads the translation widget module in Javascript or in the programming language of his/her preference, if multiple implementations are available. Downloading the translation widget is coupled by accepting the terms and conditions for its integration and usage within client websites.

R.2. The user integrates the module within his/her website in order to provide the translation service to its end users. The module is integrated within the web in a visible location, preferably the header menu or similar element of the user interface.

R.3. (optional) The user alters certain elements of the module source code, in compliance with the terms and conditions of usage, in order to provide the end users with a better user experience in terms of the client website.

R.4. (optional) The user defines certain parameters in order to improve translations of his/her website by using the translation widget. These parameters include selection of translation models and/or domains. Otherwise, the LetsMT! default settings are used for the translation widget.

End user:

R.1. The user accesses the client website in which the translation widget was previously integrated.

R.2. Using the simple menu within the widget, the user selects the target language for the translation from the *Target Language* drop-down menu and presses the *Translate* button.

R.3. (optional) If the widget version enables it, the user specifies advanced options for the translation, such as the language domain.

R.4. After requesting translation, the website is fed to the LetsMT! web-service and the results are presented to the user on the client website.

1.3.6.1 Pre-conditions and scope

For the web developer user class, the translation widget must be made available on the LetsMT! public website for the users (web developers) to download it. The LetsMT! web-service (API) must be up and running for the translation widget to be utilized.

For the end user class, the translation widget must be integrated within the client website accessed by the user and the LetsMT! web-service (API) must be operational. The user needs to select the target language and, optionally, translation parameters.

1.3.6.2 Security

The LetsMT! translation widget is made available to the general public. For its integration within client websites, it is required from the user to accept the terms and conditions for its usage.

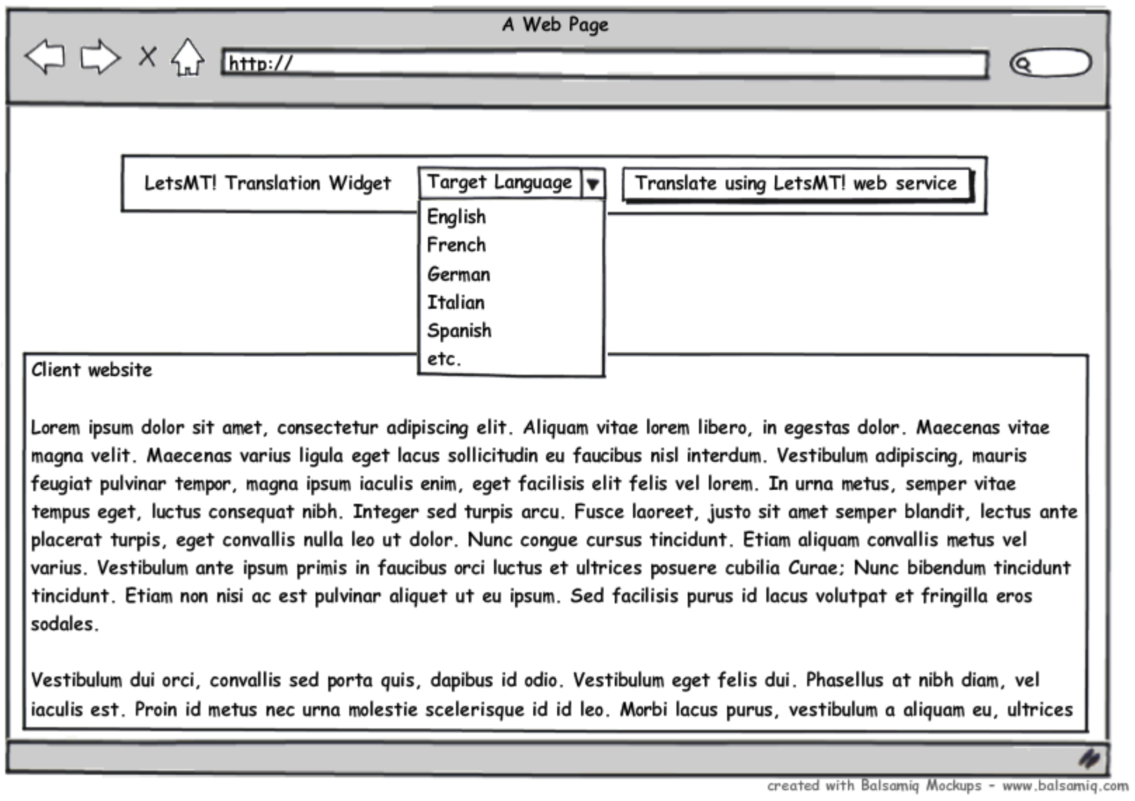
1.3.6.3 Post-conditions

A translation is provided to the user inline, within the client website elements previously containing source language text. If a problem is encountered by the widget or the LetsMT! web-service (API), an error message is displayed to the user.

1.3.6.4 Performance requirements

The translation should be provided at rate of one sentence per second. The performance of the translation widget is dependent of that of the LetsMT! web-service (API).

1.3.6.5 UI Mockup


A Web Page
http://
LetsMT! Translation Widget
Target Language
English
French
German
Italian
Spanish
etc.
Translate using LetsMT! web service
Client website
created with Balsamiq Mockups - www.balsamiq.com

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam vitae lorem libero, in egestas dolor. Maecenas vitae magna velit. Maecenas varius ligula eget lacus sollicitudin eu faucibus nisl interdum. Vestibulum adipiscing, mauris feugiat pulvinar tempor, magna ipsum iaculis enim, eget facilisis elit felis vel lorem. In urna metus, semper vitae tempus eget, luctus consequat nibh. Integer sed turpis arcu. Fusce laoreet, justo sit amet semper blandit, lectus ante placerat turpis, eget convallis nulla leo ut dolor. Nunc congue cursus tincidunt. Etiam aliquam convallis metus vel varius. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc bibendum tincidunt tincidunt. Etiam non nisi ac est pulvinar aliquet ut eu ipsum. Sed facilisis purus id lacus volutpat et fringilla eros sodales.

Vestibulum dui orci, convallis sed porta quis, dapibus id odio. Vestibulum eget felis dui. Phasellus at nibh diam, vel iaculis est. Proin id metus nec urna molestie scelerisque id id leo. Morbi lacus purus, vestibulum a aliquam eu, ultrices

Figure 6. Web page widget use mockup

1.3.7 Translate Using Browser Plug-in

One of the planned features for the LetsMT! platform is the browser plug-in – an extension for the users browser that enables real-time machine translation of websites by accessing the LetsMT! web-service (API). The browser plug-in is planned as a demo-feature to illustrate the capabilities of the LetsMT! platform by using the LetsMT! web-service (API) as a back-end. Support for Mozilla Firefox and Microsoft Internet Explorer is planned, although plug-ins for other browsers may be provided as well. The browser plug-in operation resembles that of the LetsMT! translation widget, as it provides real-time translation on the web. However, the translation widget must be integrated within the client web in order to operate, while the browser plug-in integrates with the client web browser to provide translations for virtually any website. The user story is as follows.

R.1. The user accesses the LetsMT! public website and downloads the browser plug-in for the browser of his/her choice.

R.2. The user installs the plug-in using the plug-in installation interface as provided by the specific browser. A restart of the browser may be required.

R.3. (optional) The user adjusts certain advanced parameters of the browser plug-in by using the plug-in configuration interface via the plug-in configuration interface of the browser, if made available.

R.4. The user accesses a certain client website using the browser.

R.5. After loading the website, the user right-clicks within its content and selects the *Translate using LetsMT!* dropdown menu option provided by the plug-in. The translation dropdown includes a list of target languages for the translation from which the user selects the target language of choice.

R.6. (alternative) Rather than translating the entire web page, the user selects a certain portion of text from the web page and uses the plug-in as in R.5.

R.7. The text is fed to the LetsMT! web-service (API) by the browser plug-in and the translation in the language of choice is displayed to the user inline within the website.

1.3.7.1 Pre-conditions and scope

The LetsMT! browser plug-in must be made available on the LetsMT! public website for the users to download it. Browser plug-in implementation must match the user's selection of browser. The browser plug-in must be properly installed to the client browser. The LetsMT! web-service (API) must be up and running for the browser plug-in to be utilized.

1.3.7.2 Security

The LetsMT! browser plug-in is made available to the general public. For its integration within client browsers, it is required from the user to accept the terms and conditions for its usage.

1.3.7.3 Post-conditions

A translation is provided to the user inline, within the client website elements previously containing source language text. If a problem is encountered by the browser plug-in or the LetsMT! web-service (API), an error message is displayed to the user.

1.3.7.4 Performance requirements

The translation should be provided at rate of one sentence per second. The performance of the browser plug-in is dependent of that of the LetsMT! web-service (API).

1.3.8 Translate Using CAT Tool

Integration with CAT tools will be implemented as support to translate TMX and XLIFF file formats of specified versions and with options to save (meta) data in different file tags.

R.1. Most important formats for data collection in L10N scenario are TMX and XLIFF

R.2. Markup must be preserved during translation process
  - Strip mark-up from source -> Reapply mark-up on target
  - Applies to meta-markup and formatting in-line mark-up
  - Replace mark-up indicating placeholders in source -> Reapply placeholder mark-up on target
    - This applies to content placeholders only

R.3. Progress bar must be shown to user during translation

R.4. Ensure XLIFF and TMX round-tripping (files must be readable in software that produced them)

R.5. XLIFF – add MT suggestion as an <alt-trans><target> element. Populate <alt-trans> attributes with MT metadata.
R.6. Option to replace <target> in <trans-unit>
R.7. TMX – populate empty target with MT suggestion.

1.3.8.1 Pre-conditions and scope

Users have the source data available in a standard format such as XLIFF (1.2) or TMX (1.4). The source content can contain inline markup!

User has already selected SMT system to use.

1.3.8.2 Security

Security is important. Users need to be able to separate their training data as well as translation data from publicly available data and data belonging to other users. Security breaches may cause severe damages and legal repercussions on multiple levels.

1.3.8.3 Post-conditions

- The MT translations will be automatically inserted into dedicated XML fields in the given format.
- All inline mark-up must be preserved.
- Optionally the markup should be tentatively distributed within the target segment.
- XLIFF and TMX outputs can be used as Post Editing input with major CAT tools, such as Trados Studio 2009, MemoQ, Multitrans etc.

1.3.8.4 Performance requirements

- Bulk performance – 100.000 words per minute

1.3.8.5 UI Mock-up

For file translations upload and download dialog should be simple:

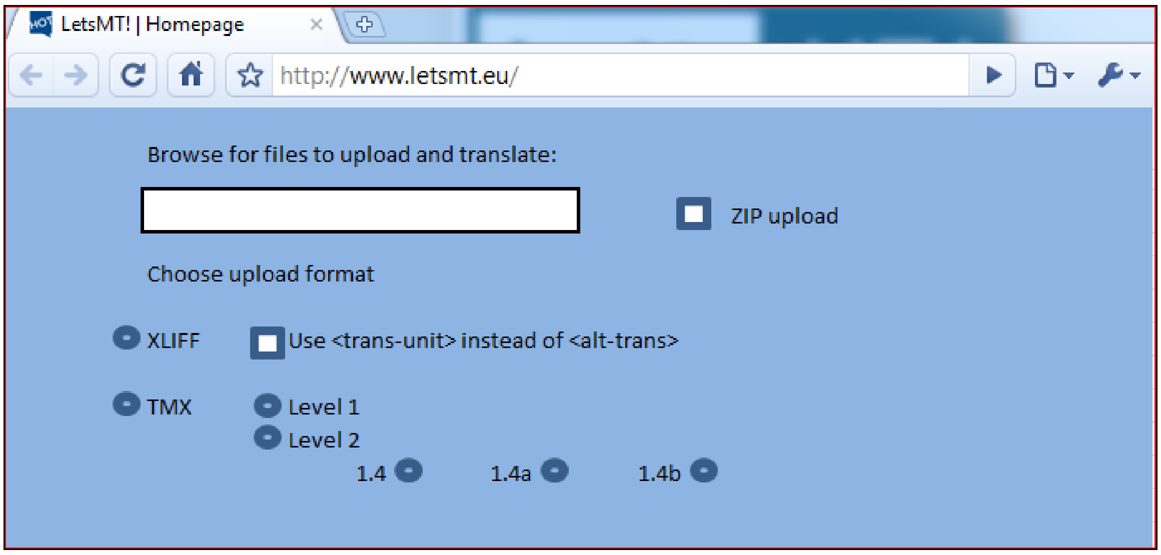
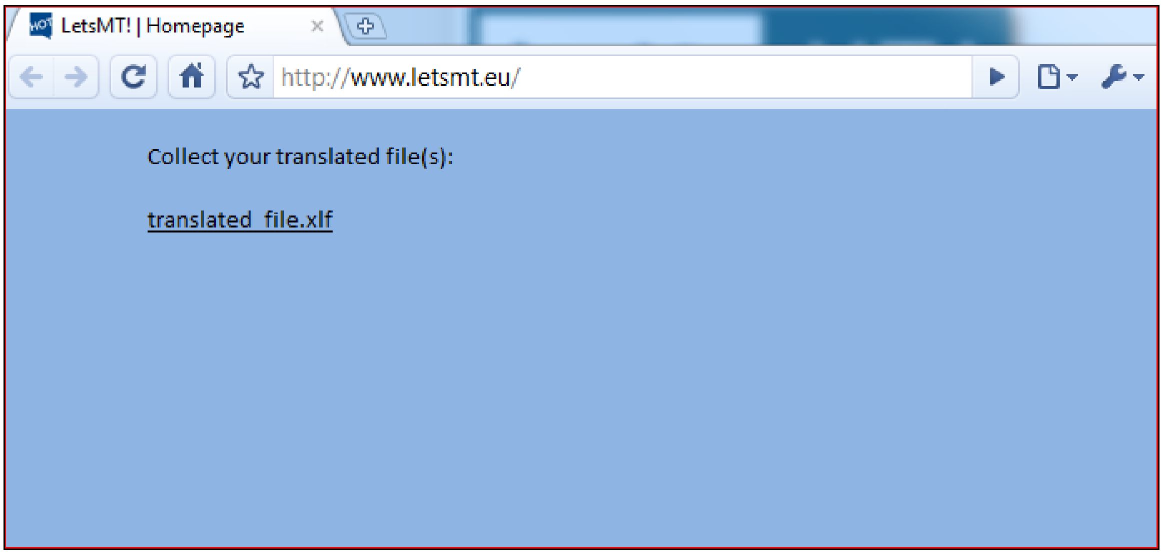


Figure 7. XLIFF and TMX translation web page mock-up

>> Progress bar >>



Collect your translated file(s):

translated_file.xlf

Figure 8. XLIFF and TMX translation result page mock-up

1.3.9 Using SMT systems in SDL Trados Studio 2009

One of the LetsMT! system's features is to provide integration with CAT tools to support "on the fly" translation requests over a network. This story describes usage of LetsMT! within SDL Trados Studio 2009.

R.1. A trained, loaded SMT system must be associated with a project (see Figure 9 on page 28). The configuration dialog might appear where the user's credentials are requested or the user might be asked to choose a translation domain or SMT system if several systems are available. LetsMT! Machine Translation Provider appears next to translation memories (see Figure 10 on page 29).

R.2. When user initiates translating a segment of a project's document (user click on particular segment), the translation result window is populated with probable translations. By default these result are suggestions from translation memories (if any) that are supplemented with results from the LetsMT! Machine Translation (see Figure 11 on page 29).

R.3. By clicking on a selected result item it will be applied as the translation ofa particular segment. The user has the option to change the translation to improve quality.

R.4. (optional) If the user updates the translation (and it is approved by an editor) it is re-cycled back to the appropriate SMT system within LetsMT! as a new parallel corpus to improve and fine tune the SMT.

1.3.9.1 Pre-conditions and scope

- SDL Trados Studio 2009 must be installed
- LetsMT! plug-in for the studio must be installed
- Translation project must be created with documents to translate within SDL Trados Studio 2009
- User must know source and target languages of the SMT system he is going to use for translating. At least one public or private (LetsMT! service subscriber's specific) SMT system must be available, that conform to source and target language.

- LetsMT! plug-in must be enabled for specific language directions

1.3.9.2 Security

- Anonymous and registered users are able to use only public SMT systems
- Professional authenticated users are able to use public systems and LetsMT! service subscriber's loaded SMT systems.
- User cannot use systems of other LetsMT! service subscribers.

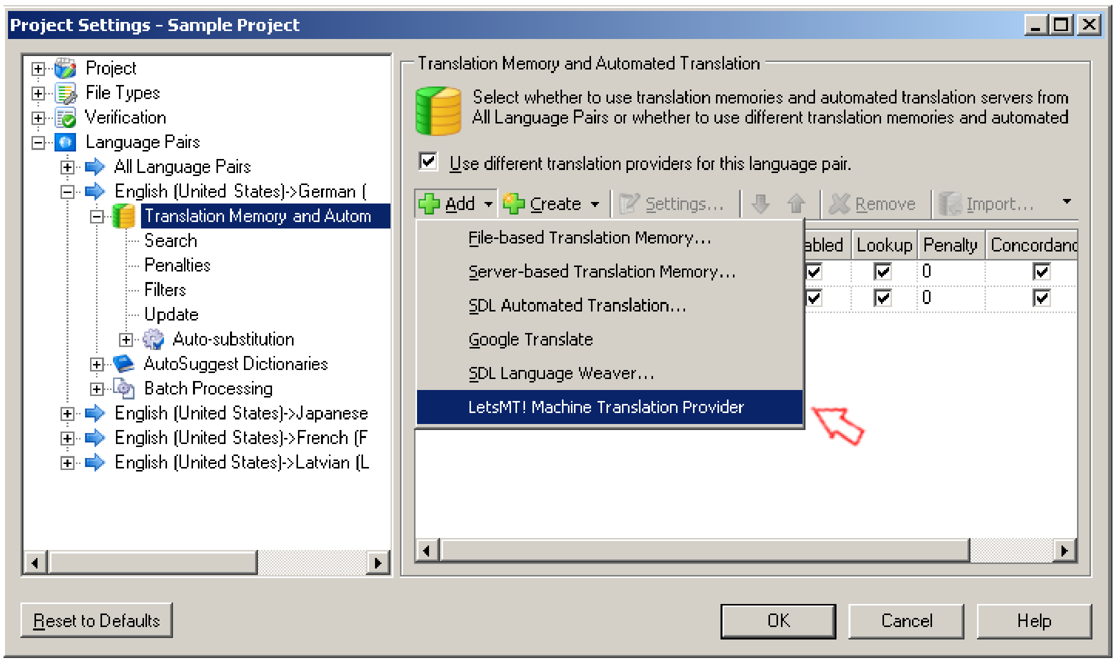
1.3.9.3 Post-conditions

Segment translations of the LetsMT! system appears in the translation result window of SDL Trados Studio 2009.

1.3.9.4 Performance requirements

Segment translation should be provided in real-time, if the system performance allows. Translation performance is dependent on the resources assigned to LetsMT! service subscriber and number of simultaneous translation requests.

1.3.9.5 Screenshots

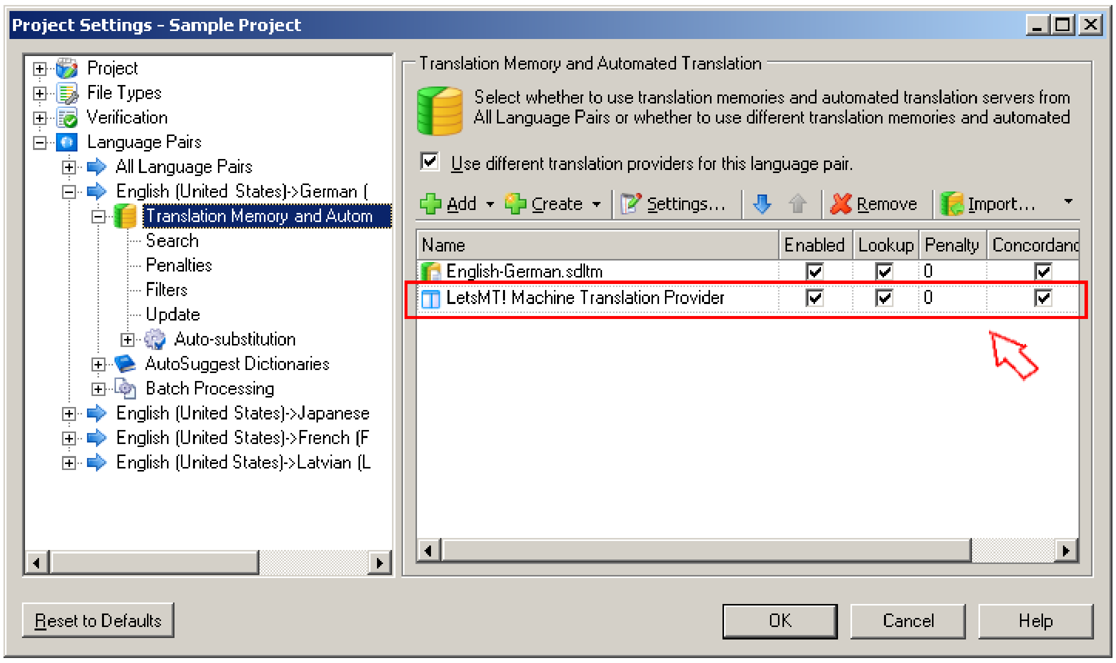

Project Settings - Sample Project
Project
File Types
Verification
Language Pairs
All Language Pairs
English (United States)->German (
Translation Memory and Autom
Search
Penalties
Filters
Update
Auto-substitution
AutoSuggest Dictionaries
Batch Processing
English (United States)->Japanese
English (United States)->French (F
English (United States)->Latvian (L
Translation Memory and Automated Translation
Add
Create
Settings...
Remove
Import...
File-based Translation Memory...
Server-based Translation Memory...
SDL Automated Translation...
Google Translate
SDL Language Weaver...
LetsMT! Machine Translation Provider
abled
Lookup
Penalty
Concordanc
Reset to Defaults
OK
Cancel
Help

Select whether to use translation memories and automated translation servers from All Language Pairs or whether to use different translation memories and automated

☑ Use different translation providers for this language pair.

abled	Lookup	Penalty	Concordanc
☑	☑	0	☑
☑	☑	0	☑

Figure 9. Add LetsMT! translation provider to project



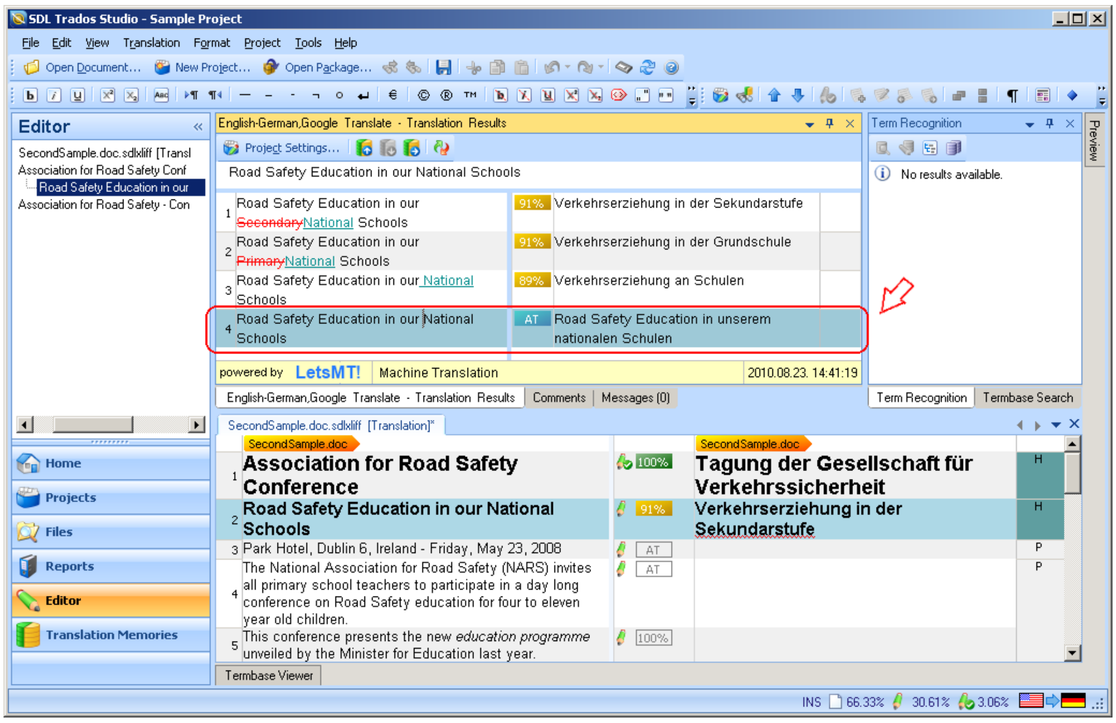
Translation Memory and Automated Translation

Select whether to use translation memories and automated translation servers from All Language Pairs or whether to use different translation memories and automated

☑ Use different translation providers for this language pair.

Name	Enabled	Lookup	Penalty	Concordanc
English-German.sdltm	☑	☑	0	☑
LetsMT! Machine Translation Provider	☑	☑	0	☑

Figure 10. Successful LetsMT! translation provider configuration



English-German,Google Translate - Translation Results

Road Safety Education in our National Schools

1	Road Safety Education in our ~~Secondary~~National Schools	91%	Verkehrserziehung in der Sekundarstufe
2	Road Safety Education in our ~~Primary~~National Schools	91%	Verkehrserziehung in der Grundschule
3	Road Safety Education in our National Schools	89%	Verkehrserziehung an Schulen
4	Road Safety Education in our National Schools	AT	Road Safety Education in unserem nationalen Schulen

powered by LetsMT! | Machine Translation | 2010.08.23. 14:41:19

Term Recognition: No results available.

SecondSample.doc.sdlxliff [Translation]*

	SecondSample.doc		SecondSample.doc	
1	Association for Road Safety Conference	100%	Tagung der Gesellschaft für Verkehrssicherheit	H
2	Road Safety Education in our National Schools	91%	Verkehrserziehung in der Sekundarstufe	H
3	Park Hotel, Dublin 6, Ireland - Friday, May 23, 2008	AT		P
4	The National Association for Road Safety (NARS) invites all primary school teachers to participate in a day long conference on Road Safety education for four to eleven year old children.	AT		P
5	This conference presents the new *education programme* unveiled by the Minister for Education last year.	100%		

INS 66.33% 30.61% 3.06%

Figure 11. Translation suggestion from LetsMT! system

1.3.10 Training user tailored SMT system

A LetsMT! system features is to train user tailored SMT systems.

R.1. User selects source and target languages and specifies the name of the SMT system he would like to train. Optionally the user may select a domain for the SMT system.

R.2. If there are any corpora satisfying source and target language criteria and the user has the right to access them, they are listed in a corpora list.

R.3. Initially all corpora are marked as selected. The user can deselect corpora he is not willing to use.

R.4. Corpora are grouped depending on type – parallel, monolingual (target), development, and evaluation. Only parallel corpora are mandatory.

R.5. By default the target side of parallel corpora also will be used also for language model training.

R.6. Optionally the user may include TAUS Data Association corpora in SMT system definition (training data set).

R.7. If the user has not specified any development corpus then a random sample of 1 000 sentences will be extracted from training data and used for tuning (MERT), these sentences will not be used in training.

R.8. If the user has not specified any evaluation corpus then a random sample of 500 sentences will be extracted from the training data and used for evaluation (BLEU score), these sentences will not be used in training.

R.9. Statistics such as the size of training data and estimated training time will be presented to the user.

R.10. Once the user has specified the translation direction, name and corpora he can launch the training task by clicking the “Train” button.

1.3.10.1 Pre-conditions and scope

The user must know at least the name, and source, and target languages of the system he is going to train. The system can be trained only if there is at least one parallel corpus for the selected language pair.

1.3.10.2 Security

Training can be started only by a user with rights to launch training tasks.

The user can see and select only corpora he has access rights to.

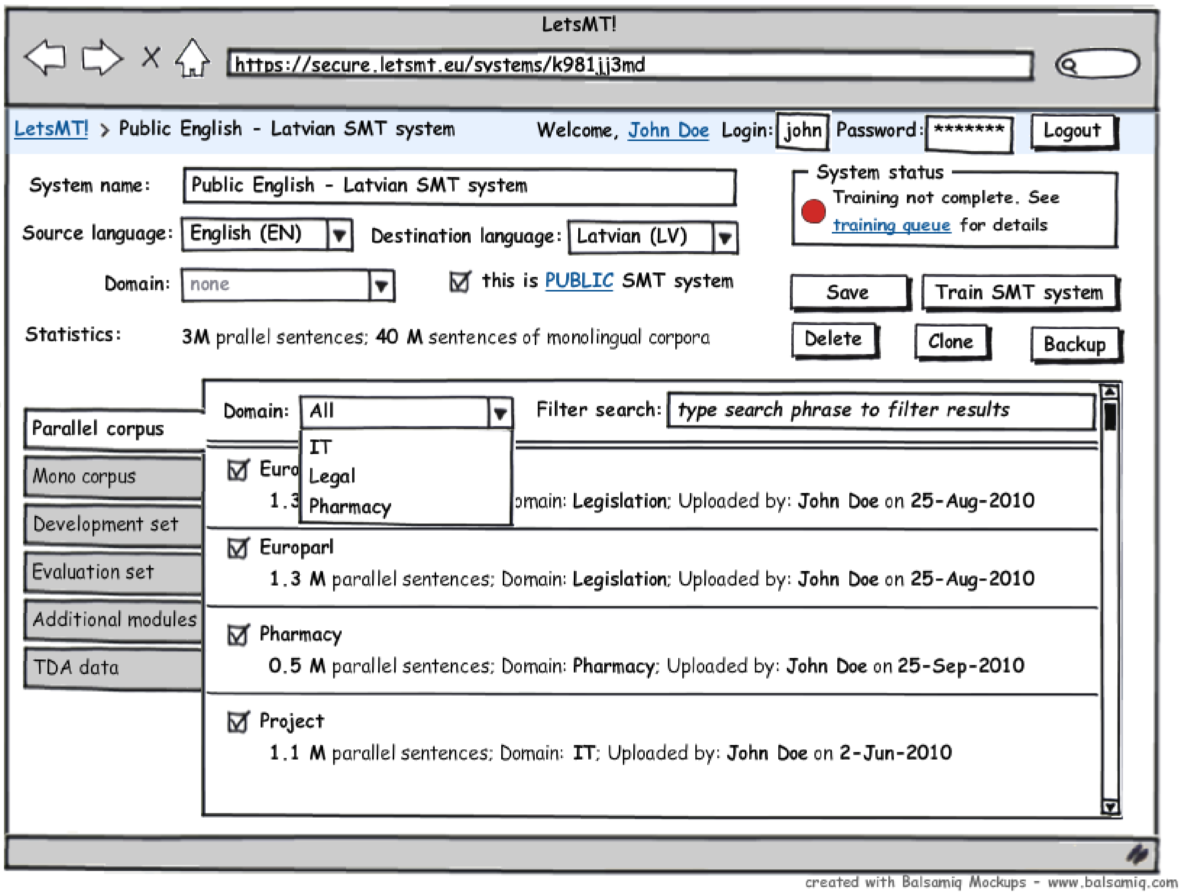
1.3.10.3 Post-conditions

- System training tasks started
- System appears in SMT system list with status “Training not complete”
- User is taken to systems training screen with a list of pending training tasks and estimate of when system will be trained is shown.

1.3.10.4 Performance requirements

The maximum training time for SMT system should not exceed 36 hours and will depend on corpora size.

1.3.10.5 UI Mock-up


LetsMT!

https://secure.letsmt.eu/systems/k981jj3md

LetsMT! > Public English - Latvian SMT system

Welcome, John Doe Login: john Password: ******* Logout

System name: Public English - Latvian SMT system

Source language: English (EN)

Destination language: Latvian (LV)

Domain: none

☑ this is PUBLIC SMT system

Statistics: 3M prallel sentences; 40 M sentences of monolingual corpora

System status

Training not complete. See training queue for details

Save | Train SMT system | Delete | Clone | Backup

Parallel corpus | Mono corpus | Development set | Evaluation set | Additional modules | TDA data

Domain: All (IT, Legal, Pharmacy)

Filter search: *type search phrase to filter results*

☑ Euro
1.3 ...omain: Legislation; Uploaded by: John Doe on 25-Aug-2010

☑ Europarl
1.3 M parallel sentences; Domain: Legislation; Uploaded by: John Doe on 25-Aug-2010

☑ Pharmacy
0.5 M parallel sentences; Domain: Pharmacy; Uploaded by: John Doe on 25-Sep-2010

☑ Project
1.1 M parallel sentences; Domain: IT; Uploaded by: John Doe on 2-Jun-2010

created with Balsamiq Mockups - www.balsamiq.com

Figure 12. SMT system definition and training screen mockup

1.3.11 Translate Financial News

Users will global business and financial news translations, through two facilities: LetsMT! widget and web browser plug-in for web page translation. The data necessary for training business and financial news translation systems will be provided by the LetsMT! publicly available system.

The potential clients range from generic news readers, third party dissemination channels, and proprietary news processing systems, benefits local language news stories from across the world.

The free news translation service will allow clients to specify their news feed in terms of the language(s) that are to be included and whether the original language originals need to be included.

In addition to the conventional English language news service, selected customers will have access to machine translated versions of the news. The added value of the service will be established by using a pre/post assessment using an on-line questionnaire and selective follow up with face to face interviews.

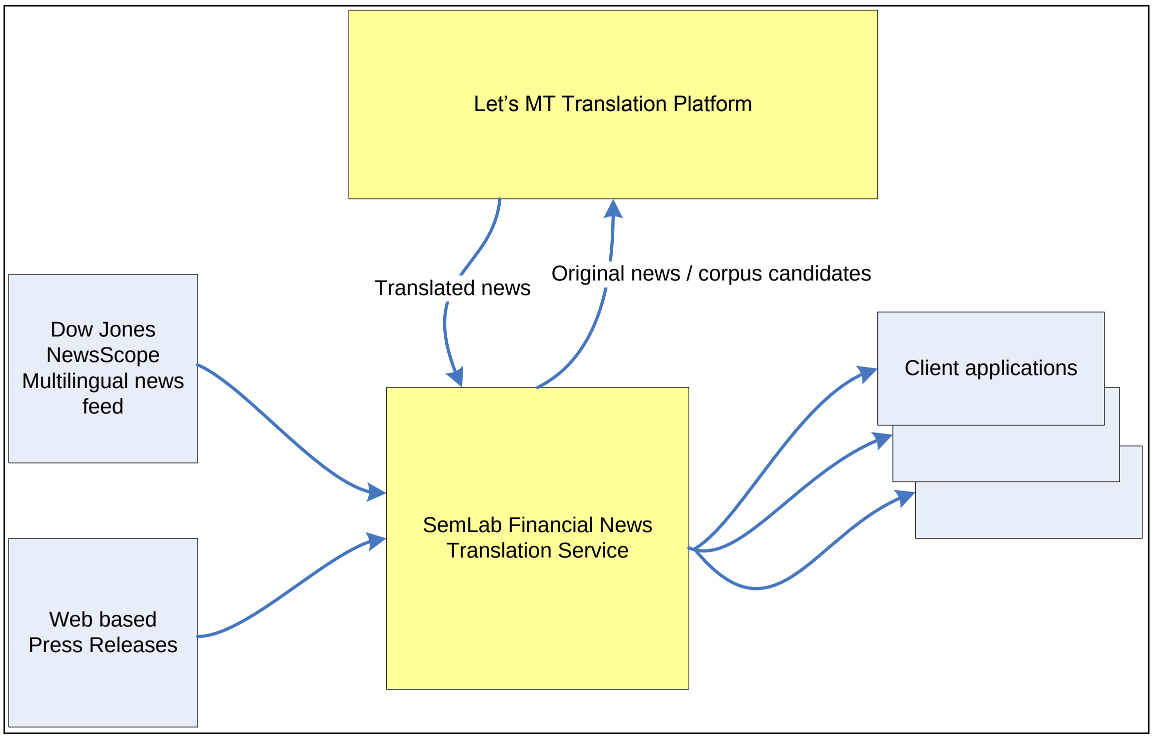

Let's MT Translation Platform
Translated news
Original news / corpus candidates
Dow Jones NewsScope Multilingual news feed
Client applications
SemLab Financial News Translation Service
Web based Press Releases

Figure 13. SemLab financial news translation service integration logical diagram

1.3.11.1 Pre-conditions and scope

The LetsMT! Translation platform needs to be ready for the financial news feeds to be used efficiently.

1.3.11.2 Security

SemLab Financial service will connect to LetsMT! platform using standard web service protocols and use authentication methods required by LetsMT!

1.3.11.3 Performance

On average system should be able to translate approx. 250-word news article in less than 1 second.

1.3.11.4 Post-conditions

Translated news are available as output through a LetsMT! web-service API.

1.4 System security overview

User access control is an important aspect of LetsMT! Platform. LetsMT! will have user authentication and authorization mechanisms based on user names and passwords. User authorization will be used to control:

- Login to website and rights to access functions available through web page interface,
- Access rights to training data (corpora), trained models and SMT systems,
- Initiation and management of training tasks,
- Access through external APIs (data upload, translation, CAT tools)

The system will employ significant security infrastructure in order to meet multi-tenancy, IPR and reliability requirements.

Users will have to identify themselves when they will access the system. The system knows what kind of user or external agent is accessing it and to which service subscriber it belongs. The external systems have to send not only translation queries and other commands but also information necessary for authentication.

Authenticated users will work with the LetsMT! system via a secure internet connection (HTTPS[1]) to ensure a secure transfer of authentication information, data, translation requests and translations.

1.4.1 Multi-tenancy

LetsMT! will have different types of users. There will be individual users mainly working with publicly available LetsMT! services and there will be client services subscribers which will have proprietary data, SMT systems, multiple users etc. LetsMT! system will have a multi-tenancy principle in its architecture and user access rights control. Multi-tenancy[2] refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple clients with completely segregated data sets.

- All objects (corpora, models, trained SMT systems) in the LetsMT! system will belong to (or will be owned by) a single LetsMT! service subscriber. And will not be available users of any other LetsMT! service subscriber.
- Objects may have a special attribute "public" which means that any authenticated system user is allowed to use this resource.
- Objects belonging to one LetsMT! service subscriber are accessible only to that LetsMT! service subscriber users (except for PUBLIC objects). Sharing of resources between LetsMT! service subscribers can be implemented later .
- Access level and available operations is defined by role membership.

The diagram below gives overview of multi-layered security architecture of LetsMT! system.

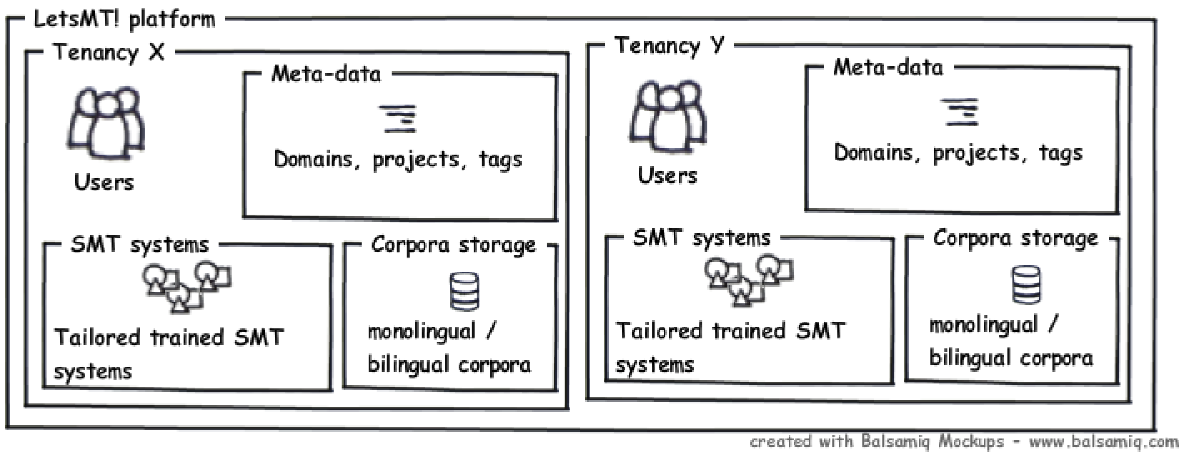

LetsMT! platform
Tenancy X
Users
Meta-data
Domains, projects, tags
SMT systems
Tailored trained SMT systems
Corpora storage
monolingual / bilingual corpora
Tenancy Y
Users
Meta-data
Domains, projects, tags
SMT systems
Tailored trained SMT systems
Corpora storage
monolingual / bilingual corpora
created with Balsamiq Mockups - www.balsamiq.com

Figure 14. Illustration of multi-tenancy support in LetsMT!

[1] HTTPS: http://en.wikipedia.org/wiki/HTTP_Secure
[2] Multitenancy: http://en.wikipedia.org/wiki/Multitenancy
http://msdn.microsoft.com/en-us/library/aa479086.aspx

As described in the conclusions section 5.1 of D1.1. – the LetsMT! system has to support multi-tenancy in order to comply with the most restrictive requirement – 16 % of interviewees dismiss sharing data. Each LetsMT! service subscriber have its own user base, user groups, corpora set, trained SMT system set, meta-data which is accessible only for authenticated users of the same LetsMT! service subscriber.

LetsMT! service subscribers will be allowed to mark corpora and trained SMT systems as PUBLIC thus allowing any system user to use these resources. This will be the only means of sharing data for the initial release of the system. A mechanism for sharing corpora or SMT systems may be developed later to allow access to resources of one LetsMT! service subscriber to users of other LetsMT! service subscriber.

1.4.2 User Roles, Accounts

The LetsMT! system has several user groups. By default, users access the system through a web interface and get access permissions of the 'Anonymous User' group.

The following system user roles will be defined for the LetsMT! system:

- Anonymous User (AU) – Any user from the Internet. No authentication is necessary. Can access public information and public SMT systems for evaluation.
- Registered User (RU) – User who can use trained SMT systems. He can access only public SMT systems and SMT systems in his LetsMT! service subscriber. He can use SMT systems both using the LetsMT! web interface and an API
- Registered Power User (PU) – User who manages training data, training tasks and trained SMT systems of LetsMT! service subscriber.
- LetsMT! service subscriber Administrator (TA) – User who has full control over a LetsMT! service subscriber data set, he/she creates it, manages users of LetsMT! service subscriber etc. He can manage training data, training tasks and trained SMT systems of LetsMT! service subscriber, but he can grant these rights also to *Registered Power User*.
- System Administrator (SA) – User who has the maximum allowed rights. May manage LetsMT! service subscribers; manage all users, resources, training tasks, trained SMT systems etc. SA may also change SMT system definition scripts for any LetsMT! service subscriber/any SMT system.

A new user can create a new account and by default he creates his own LetsMT! service subscriber account and becomes a *LetsMT! service subscriber Administrator* of this subscriber. He can create new accounts for other users (*Registered Users* and *Registered Power Users*) in his LetsMT! service subscriber. All users belonging to the LetsMT! service subscriber can access only public resources and resources in their LetsMT! service subscriber.

Changes in users and roles assignments must be stored in audit log.

1.4.3 Authentication mechanisms

Initially the LetsMT! system will support only internal authentication mechanisms and will maintain a user list within the LetsMT! system. In the future, when customer interest in the system will increase, support for additional authentication mechanisms (for example, LDAP, MS Active Directory Federation Services) can be developed based on the needs of actual customers.

During user authentication no password is sent over the network, but rather challenge-response authentication methods must be employed.

Authentication will be implemented using secure internet connection over HTTPS protocol to ensure secure transfer of authentication information.

1.5 User-Accessible Functions List

The following list of user-accessible functions will be implemented in LetsMT! in order to meet the functional and security requirements described in previous sections.

Functional group	**Description**	**Function**
Public translation services	System will allow testing of SMT translation to anonymous users in order to evaluate possibilities of tailored SMT. Anonymous users will be able to translate text using web page or web browser plug-in.	• Anonymous translation (web page) • Anonymous translation (web browser plug-in)
Corpora management	Corpora management will allow organizations to build domain-specific SMT systems to meet organization specific needs. Corpora repository will allow uploading and storing monolingual and bilingual corpora, alongside with metadata. Management of uploaded corpora in means of revocation and removal of uploaded content will be supported.	• Corpora upload • LetsMT! service subscriber's corpora browsing • Corpora details view • Edit corpora metadata • Corpora removal
Translation services	Translation services will allow to use web page to translate most requested formats, like TMX, DOC and as integration with CAT tools to support "on the fly" translation requests over network.	• Translation (web page) • Translation API (for example, SDL Trados)
SMT system tailoring and training	Option to tailor SMT systems and initiate training of systems to find most appropriate System will allow limited use of extension for translation process like pre/post-processors. LetsMT! will allow to include LetsMT! service subscriber's corpora and public corpora by selecting appropriate domains for specific SMT system. Also limited amount of Moses options will be allowed to change by users.	• SMT system list browsing • SMT system list management • SMT system details • SMT system (re) training request/cancellation • Edit metadata definitions

Functional group	Description	Function
Supplemental system services	User, group, LetsMT! service subscriber and domain classification management. Depending on business model requirements additional supporting services may become necessary to implement different membership levels and accounting of resources used.	• Logging in/out • LetsMT! service subscriber User management • Groups management • LetsMT! service subscribers management • Role membership management • Security log access • SMT training task management • System-wide property management • System-wide user management

Table 4. Let's MT! system functions list

1.5.1 Permissions

The following table lists entities and functions that can be accessed by adding respective roles to registered users.

Function	Role membership				
	AU	RU	PU	TA	SA
Anonymous translation (web page)	X	X	X	X	X
Translation using web browser plug-in	X	X	X	X	X
Corpora upload			X	X	X
LetsMT! service subscriber’s corpora browsing			X	X	X
Corpora details view			X	X	X
Edit corpora metadata			X	X	X
Corpora removal			X	X	X
Translation (web page)		X	X	X	X
Translation API (for example, SDL Trados)		X	X	X	X
SMT system list browsing		X	X	X	X
SMT system list management			X	X	X
SMT system details editing			X	X	X
SMT system (re) training request/cancellation			X	X	X

Function	Role membership				
	AU	RU	PU	TA	SA
Edit metadata definitions				X	X
Logging in/out	X	X	X	X	X
LetsMT! service subscriber User management				X	X
Groups management				X	X
LetsMT! service subscribers management					X
Role membership management				X	X
Security log access				X	X
SMT training task management					X
System-wide property management					X
System-wide user management					X

Table 5. Function-to-role authorization matrix

2 Logical Design

2.1 *Overview of System Architecture*

LetsMT! system will have multitier architecture. It will have (i) an interface layer for user interface and APIs with external systems, (ii) an application logic layer for system logic and (iii) a data storage layer consisting of file and database storage. LetsMT! system will be performing various time and resource consuming tasks; these tasks will be defined by application logic and will be sent to High-performance Computing (HPC) Cluster for execution.

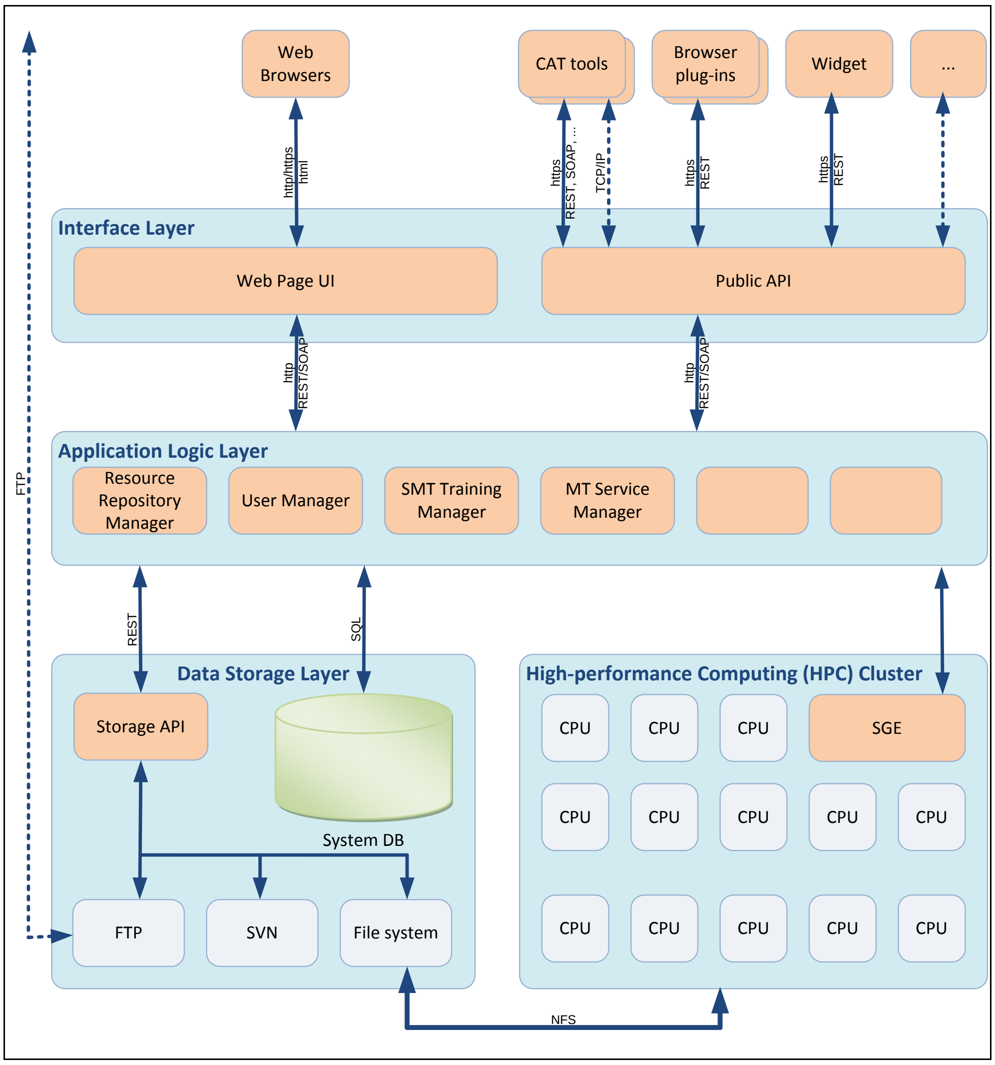


Figure 15. Software architecture of LetsMT! platform

The Interface layer will provide interfaces between the LetsMT! system and external users. The system will have both human and machine users. Human users will access the system through web browsers using the LetsMT! web page interface. External systems such as CAT tools and browser plug-ins will access the LetsMT! system through a public API. The public API will be available through both REST[3]/JSON[4] and SOAP[5] protocol web services. Some CAT

[3] REST: http://en.wikipedia.org/wiki/Representational_State_Transfer
[4] JSON: http://www.json.org/, http://en.wikipedia.org/wiki/JSON
[5] SOAP: http://www.w3.org/TR/soap/, http://en.wikipedia.org/wiki/SOAP

tools or other external systems may require different interfaces; they might be introduced if necessary. A secure HTTPS protocol will be used to ensure secure user authentication and secure data transfer.

An application logic layer will contain a set of modules responsible for the main functionality or logic of the systems. It will receive queries and commands from the interface layer and will prepare answers or perform tasks using data storage and HPC cluster. This layer will contain several modules such as Resource Repository Manager, User Manager, SMT Training Manager etc.. The interface layer will access the application logic layer through both REST/JSON and SOAP protocol web services. The same protocols will be used for communications between modules in the application logic layer.

The LetsMT! system as a data sharing and MT platform will store a huge amount of SMT training data (parallel and monolingual corpora) and trained models of SMT systems. These data will be stored in a network file system. The file system will also be used to store various tools necessary for data processing and SMT training and trained models and SMT systems. As training data will be changing, the training data repository will be based on a version-controlled file system (SVN[6]). SQL based database will be used to store system configuration, metadata and statistics about training data and about trained SMT systems etc. There will also be FTP server which will be used to allow users to upload their training data. Modules from the application logic layer will access data storage layer through REST protocol based web service interface and SQL queries.

A high-performance Computing Cluster will be used to execute many different data processing tasks, training and running SMT systems. Modules from the application logic layer will create jobs and send them to HPC cluster to execute. HPC cluster is responsible for accepting, scheduling, dispatching, and managing the remote and distributed execution of large numbers of standalone, parallel or interactive jobs. It also manages and schedules the allocation of distributed resources such as processors, memory and disk space. LetsMT! HPC cluster will be based on Sun Grid Engine[7] (SGE). HPC cluster will access data stored in data storage layer using Network File System[8] (NFS) protocol.

2.2 *Interface Layer*

The Interface Layer consists of two main modules (see Figure 16. Interface Layer and communication with client tools of the LetsMT! platform on page 40): the Web Page User Interface (UI) and the Application Programming Interface (API) for translation automation and integration with external applications. Both are available directly from Internet.

[6] SVN: http://subversion.apache.org/, http://en.wikipedia.org/wiki/Apache_Subversion
[7] SGE: http://gridengine.sunsource.net, http://en.wikipedia.org/wiki/Sun_Grid_Engine
[8] NFS: http://tools.ietf.org/html/rfc3530, http://en.wikipedia.org/wiki/Network_File_System_(protocol)

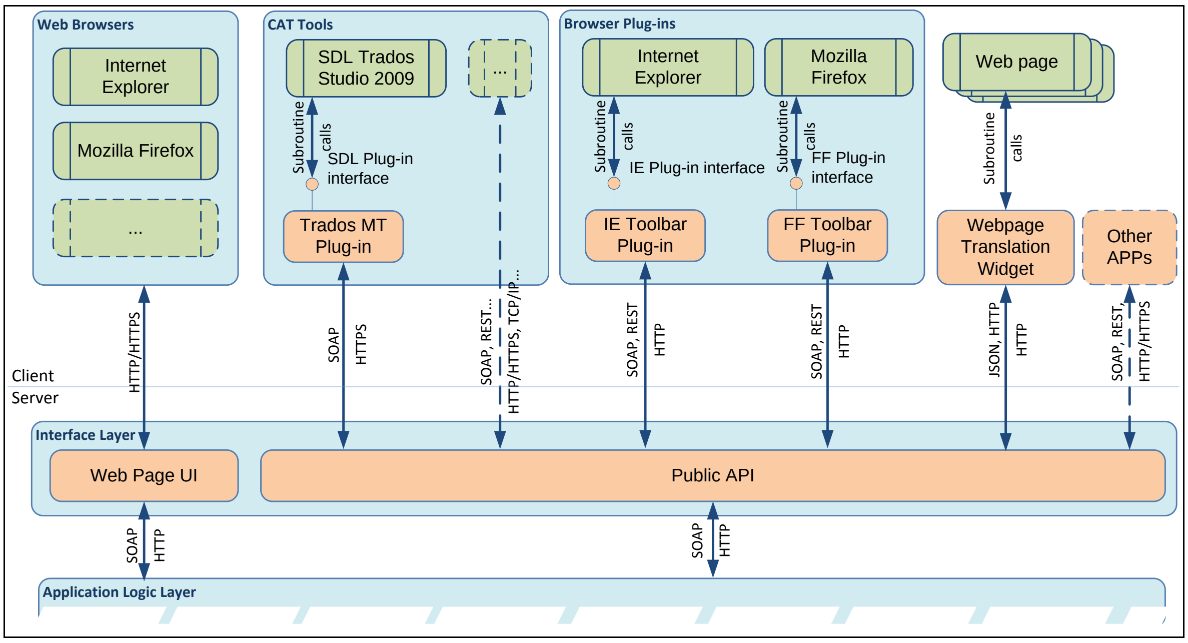


Figure 16. Interface Layer and communication with client tools of the LetsMT! platform

2.2.1 Web Page UI

All LetsMT! services will be accessible through the Web page User Interface (UI). Business and professional users of the localization and translation industry will use the non-public part of the LetsMT! platform for uploading their parallel and monolingual corpora, building custom SMT systems from the specified collections of training data and accessing these SMT systems in their productivity environments (typically, various CAT tools). Non-public part will be available using **TLS/SSL**[9] cryptographic methods to protect users' private content that is sent over the network. All other users will be able to access the public part of the Web Page UI to use publicly available SMT systems.

Web page UI will provide following main functionality:

- User authentication
- Large file upload & data sharing
- Training data management
- SMT system management
- SMT system training
- Translating
- User management

2.2.1.1 Uploading large files

The Web Page UI will support uploading large data volumes over the Internet and resuming of interrupted operations due to network connection interruptions.

Large file upload is related to bilingual and monolingual training and evaluation data uploads. Many different file formats are relevant in connection with upload data.

[9] http://en.wikipedia.org/wiki/Transport_Layer_Security

Supported upload file formats will be TMX, XLIFF, TBX, CSV, plain text files (simple or tab-delimited), PDF, DOC and HTML.

Together with uploaded corpora files metadata should be provided. See 1.2.4 - UPLOAD AND SHARING OF CORPORA USE SCENARIO (page 14) for meta-data type requirements.

Uploaded files will be stored using Data Repository Management functionality.

2.2.1.2 SMT system training

Training job management of the Application Logic layer will be accessible through the Web Page UI. A user will be able (depending on security restrictions) to define SMT system configuration by selecting:

- Translation direction (source/target language),
- Monolingual and bilingual corpora,
- Training and evaluation corpora,
- Access level of the system (weather it will be public or restricted for use within particular LetsMT! service subscriber),
- System's name
- And additional metadata and training options might be specified.

For all defined system configurations appropriate SMT system training job is queued using the Training job management.

2.2.1.3 SMT system browsing

When a user wants to translate a text or perform other actions on a SMT system he/she first has to locate translation system to use. A user will be able to see all available SMT system, depending on its access rights.

Various filters by source/target language, domain and other metadata will be implemented to relieve system selection.

User permissions and LetsMT! service subscriber membership restrictions must be taken into account when providing SMT system list to user.

2.2.1.4 Translating

Web page UI will contain website for text translation. The users of the website will be able to enter text as well as upload entire document or just providing URL to the document or website as input for translating with specific SMT system.

Additionally single words or terms might be translated using vocabulary, terminology database or n-best list, depending on SMT system's configuration.

2.2.1.5 User management

Administrative users will be able to access and administrate users/LetsMT! service subscribers and permissions. Following features will be supported:

- Add/modify/delete user;
- Disable/enable user
- Add/remove LetsMT! service subscriber;
- Modify LetsMT! service subscribers metadata and resources.

2.2.2 Public API

The Public API is a public interfacing component that provides LetsMT! functionality to external applications like CAT tools, webpage translation widgets, web browsers and other translation applications that might be integrated with LetsMT! services. Web developers will be able to use this API to integrate LetsMT! services in their products. The Public API will be able to communicate with external application using SOAP, REST - JSON or simple HTTP serialization methods over HTTP/HTTPS protocol.

In general, The Public API works as a proxy or filter to the Application Logic Layer components providing subset of its functionality. Additionally it could serve specific custom functions based on functionality of the Application Logic Layer or do data transformations that are necessary for integration with external applications, e.g., CAT tools.

The Public API serves functionality that is associated with a translation process:

- User authentication;
- Translation system querying;
- Translation;
- Translation updating.

The Public API will be implemented as a public web service that will be accessible directly from external applications.

2.2.2.1 Translation system querying

Before actual text translation the API client will have to be informed about available translation systems that could be used in the translation process. With translation system querying this API will provide information about available translation directions (source and target languages), available domains, and particular system identifiers.

2.2.2.2 Translation

Text translation is the main functionality of the Public API. External applications will be able to obtain translations by passing required text together with translation system identifier and user authentication data. A system identifier will determine which translation systems will do particular translation task.

2.2.2.3 Translation updating

This API will support translation updating or translation unit upload. CAT tool like SDL Trados Studio 2009 support upload of post-edited translations of its translation providers. This feature also might be used by webpage translation widget users to get more accurate translation for their webpage.

Uploading data will consists of

- Source and target language,
- Source and target text,
- Optionally domain could be provided.

All uploaded data will be stored in translation memory of LetsMT! that could be included in training of translation models.

2.2.3 Performance requirements

The response of the Web Page UI and Public API should be provided in real time. The performance of the system (especially translation services) is dependent on translation input amount, application logic layer and underlying translation system load.

2.2.4 Authentication

When Web Page UI and Public API users access the system they have to identify themselves. The users have to send their username and password before accessing non-public functionality and resources.

Web pages UI will use HTML Form based authentication[10] method, which is supported by all web browsers. Public API will use basic access authentication[11] that is part of the HTTP 1.1. protocol. To make both mentioned authentication methods secure those will be combined with cryptographic methods of application layer like TLS/SSL.

2.3 Client components

The solution will deliver the following client components to integrate LetsMT! services with external tools:

- translation widget provided for inclusion into websites to translate their content;
- browser plug-ins that will provide the quickest access to translation;
- integration in CAT tools.

2.3.1 CAT tools

Localisation and translation industry business and translation professionals will be able to use their custom SMT systems from the specified collections of training data, public systems of the LetsMT! platform and accessing these solutions in their productivity environments (typically, various CAT tools).

Currently is has been decided to integrate the LetsMT! platform with SDL Trados Studio 2009. Integration will be made directly into the translators' daily workflow. Users of SDL Trados Studio 2009 would use SMT systems as easily as translation memories (TM) are used. Particular translation result from the SMT system will be displayed together with the suggestion from the TM.

Later integration with MemoQ, Multitrans, Worldserver, etc. could be considered.

2.3.1.1 Functionality

Two main functions are required for LetsMT! plug-in for SDL Trados Studio 2009:

- Ability to choose which SMT system to use in the translation process. LetsMT! platform might provide more SMT systems for the same language direction and even domain, so each user should be able to choose SMT system that targets the most specific needs.
- Receive translation results from LetsMT! platform of segments to translate.

[10] http://en.wikipedia.org/wiki/HTTP%2BHTML_Form_based_authentication

[11] http://en.wikipedia.org/wiki/Basic_access_authentication

Additionally translation unit upload could be implemented to upload post-edited translations to the LetsMT! platform. Uploaded translation units could be used as additional corpora in SMT system training.

2.3.1.2 Implementation technique

Integration will be developed using SDL Trados Studio 2009 plug-in mechanism and its APIs and SDKs.

Developed LetsMT! plug-in will provide available SMT systems of the LetsMT! platform and deliver segment translations to the SDL Trados Studio 2009. All communication will be developed over SOAP protocol by using the LetsMT! web-service (API) as a back-end.

2.3.1.3 Deployment

LetsMT! plug-in will be client side component that has to be installed on client computer and should be registered into SDL Trados Studio 2009.

2.3.1.4 Security

Users shall register using the Public API authentication mechanism before using the LetsMT! platform SMT systems via CAT tools. Authentication is necessary to identify which custom SMT systems are available for the user. TLS/SSL data encryption will be used to protect the users' content that is sent over the network.

2.3.2 Browser Plug-ins

The LetsMT! browser plug-ins and the web page translation widget serve the purpose of demonstrating the basic capabilities of the LetsMT! platform to a wide spectrum of potential users. The browser plug-in itself is envisioned as a tool for quick and easy access to the basic translation services of the LetsMT! platform by using the user interfaces of client web browsers.

Currently, it is planned to support only Microsoft Internet Explorer and Mozilla Firefox web browsers. After a quick and easy installation to the client browser, the user translates websites by using the provided plug-in interface within the browser.

The plug-ins are implemented as software interfaces to the LetsMT! web service (SOAP API). They mediate between the platform API and the client browser interface and web presentation layer.

2.3.2.1 Functionality

LetsMT! browser plug-ins provide the user with machine translation on the web by using the LetsMT! platform facilities and the interface of their favourite web browser. The basic functionality of this tool includes:

- The ability to translate custom client websites using the browser interface. Basically, the user accesses a website and uses the browser interface to forward the website to the LetsMT! web service (API) and receive the results within the interface, maintaining the structure of the website.
- A configuration tool within the browser plug-in configuration back-end. The back-end tool enables configuration and fine-tuning of the browser plug-in according to specific user requirements such as possibly the choice of translation system(s), etc.

2.3.2.2 Implementation technique

The browser plug-ins will be developed according to specific requirements of selected browsers. The Browser Helper Object and other necessary facilities of the Microsoft .NET platform will be developed for integration with Internet Explorer. For Mozilla Firefox, Javascript and XML will be used in concordance with the manual for the development of Firefox plug-ins. All communication will be developed over SOAP protocol by using the LetsMT! web service (API) as a back-end and the facilities of selected programming languages concerning web services functionality.

2.3.2.3 Deployment

Users will obtain the browser plug-ins from the LetsMT! public website and install them into their browsers by using the plug-in installation interface provided by the browsers.

2.3.2.4 Security

Users shall authenticate implicitly, by using the LetsMT! web service (API) authentication mechanism before using the SMT systems of LetsMT! platform. Authentication is necessary to identify which custom SMT systems are available for the user. If the user does not provide his own authentication within the plug-in configuration tool, the default one for the browser plug-in will be used. TLS/SSL data encryption will be used to protect the user content sent over the network.

2.3.3 Web Page Translation Widget

The LetsMT! browser plug-ins and the web page translation widget serve the purpose of demonstrating the basic capabilities of the LetsMT! platform to a wide spectrum of potential users. The web page translation widget is envisioned for source code integration within the client website in order to enable multilingual web features to these websites by seemless integration of the widget and the LetsMT! platform web service (API).

The widget is to be developed in Javascript and available as open source software for integration within client websites.

The widget is implemented as an interface to the LetsMT! web service (SOAP API). It mediates between the platform API and the client web presentation layer.

2.3.3.1 Functionality

The web page translation widget provides prospective users, i.e. web developers and end users of their websites, with the possibility of enabling multilingual web features in their websites through the usage of LetsMT! platform services. There are two basic functionalities offered by the translation widget:

- Custom integration of the widget source code within the client website. This task is carried out by website developers wishing to enable multilingual features within their websites.
- Translation of websites by using the LetsMT! platform. Basically, the user accessing the website containing the widget has the option to translate the source language into a target language of his/her choice. The widget forwards the content of the current web page to the LetsMT! web service (API) and displays the results to the user, preserving the structure of the web page.

2.3.3.2 Implementation technique

Implementation of the web page translation widget will be done in Javascript. Modules in other languages will possibly also be developed. All communication will be developed over SOAP protocol by using the LetsMT! web service (API) as a back-end and the facilities of selected programming languages concerning web services functionality.

2.3.3.3 Deployment

The widget will be made available to the general public from the LetsMT! public website. Website developers wishing to use the widget will integrate them within their websites according to their own design goals.

2.3.3.4 Security

Users shall authenticate implicitly, by using the LetsMT! web service (API) authentication mechanism before using the SMT systems of LetsMT! platform. Authentication is necessary to identify which custom SMT systems are available for the user. If the user does not provide authentication within the source code of the widget, the default one for the widget will be used. TLS/SSL data encryption will be used to protect the user content sent over the network.

2.4 Application Logic Layer

The Application Logic Layer is the interfacing component between high-level frontend services (LetsMT! website and public APIs) and the low-level backend services (Resource Repository and HPC Cluster). The Application Logic Layer manages the application logic of LetsMT! Platform, which includes the following functional responsibilities:

- User authentication and permission control
- Resource Repository management
- Training job management
- Translation service management
- User settings management
- Report management

The Application Logic Layer will be implemented as a web service that provides an internal API to frontend services. This API will not be accessible directly from outside, and will be interfaced by frontend services.

2.4.1 User authentication and permission control

Application Logic Layer will include a subcomponent that centralizes user authentication. User authentication is necessary for logging in to LetsMT! website and for access control to public APIs. Internally, user authentication is used to control access to resources in Resource Repository and HPC Cluster.

2.4.2 Resource Repository management

This subcomponent will provide an interface for browsing and searching the training corpora and MT systems, editing metadata and data upload.

Internally, it will make the Resource Repository available to HPC Cluster for training and translation jobs, which will need to access the training data and trained MT models directly.

To support large corpora upload via FTP, the Resource Repository will include FTP client and/or server functionality. The upload details will be configured in the LetsMT! website, and processed by the Application Logic Layer.

HTTP upload via rich client on the client side web browser will also be supported. The Application Logic Layer will access Resource Repository to store the uploaded data blocks.

After upload, the Application Data Layer will start a job on the HPC Cluster to extract the text content and normalize the data to the internal storage format and gather metadata (such as word and sentence count). The Resource Repository will retain the corpora in both the original format and internal format as different tools and settings might be used to extract the textual data later.

2.4.3 Training job management

This subcomponent is for preparing and submitting training jobs to the HPC Cluster, as well as monitoring their progress.

The training process is a complex task involving many steps that depend on the training configuration. The training will be managed by EMS (Experiment Management System) that is included in Moses. The EMS supports execution on HPC Cluster, where it is executed as a script on a head node, which submits additional jobs to the HPC Cluster as necessary and monitors their progress.

The subcomponents role will be to prepare the configuration for EMS and submit the training job to the HPC Cluster. The configuration will be template based, requiring the user to specify some required fields (the training corpora) and several optional fields.

2.4.4 Translation service management

This subcomponent is for running the translation engines on HPC Cluster. It will monitor the state of the running engines, launch additional engines and terminate them on demand. To mitigate high demand and response times, several engine instances might be launched. The subcomponent will distribute load between the instances. Several load management policies might be implemented, specifying constraints on total allowed HPC load as well as requirements for individual translation engines (availability, response time, etc).

2.4.5 User settings management

This subcomponent will provide a proxy to access the user preferences for the LetsMT! platform, which will be stored in the Application Database.

2.4.6 Report management

This subcomponent will collect and process logs and state information from training jobs, as well as generate and store reports.

This component also will collect statistics on number of loaded instances, storage utilization and traffic. This data will be used for billing purposes.

2.4.7 Non-functional requirements

The Application Logic Layer will perform various control and management tasks and pass the information between its interfaced components. These tasks are not heavily CPU or I/O dependant.

As the Application Logic Layer is a potential point of failure for the whole platform, the non-functional requirements include fault tolerance.

In case the web service providing the Application Logic Layer is restarted, the implementation will reinitialize its state by querying the HPC and Data Storage, as well as retrieving state information from the Application Database.

2.5 Resource Repository

The **Resource Repository** will provide all functionality necessary for uploading, storing and management of SMT training data and trained models. The Resource Repository has the following key features:

- Provide efficient large file access to HPC Cluster
- Incremental upload of large files
- Fault tolerance
- Scalability

Uploaded and processed training data will be stored in the Resource Repository and will be accessible for browsing, searching and management via LetsMT! webpage. SMT training data stored in the Resource Repository will have both actual training data and metadata describing these data. Actual training data typically are huge files in specific formats, for example, translation memories in TMX format; this data will be stored in the Resource Repository as files. Metadata associated with these files will be stored in SQL database which will allow using of metadata for browsing and searching through Resource Repository.

2.5.1 Data Upload and Storage

LetsMT! users will be allowed to upload data in widely used formats. Tools for processing uploaded resources will convert resources to unified format, validate and evaluate before they are stored in the repository. In most complicated cases uploaded data will be converted from document formats such as PDF or DOC and processed with document and sentence alignment tools. Automatic data quality evaluation will be performed during the data processing to assess a quality of the user uploaded content.

2.5.2 Storage of tools

The Resource Repository will store tools necessary for training and running of SMT systems, including:

- Moses training scripts and tools
- Moses decoder
- SRILM / IRSTLM / RANDLM language model tools
- Preprocessing and post-processing tools

Separate locations for production and test environments will be provided to allow testing of new versions of tools.

2.5.3 Access from HPC Cluster

The HPC Cluster will be able to access the necessary input and output files (training data, trained model files) directly at file system level. The Resource Repository will provide efficient and secure access to these files.

2.5.4 Non-functional requirements

The Resource Repository will be heavily I/O bound for HPC tasks (training and translation), but will require little CPU resources.

As the Resource Repository is a potential point of failure for the whole platform, the non-functional requirements include fault tolerance.

2.6 High Performance Computing Cluster

The **High Performance Computing (HPC) Cluster** will provide all functionality necessary for building new SMT systems and running the translation systems. The HPC Cluster has the following key features:

- Provide computing resources for training data processing, SMT training and translation
- Multi-core and multi-machine job scheduling and queuing
- Status reporting
- Fault tolerance
- Load balancing
- Scalability and dynamic resource allocation

The HPC Cluster will be responsible for execution of two types of jobs:

- SMT training with state-of-the-art Moses SMT toolkits and the necessary support tools (Giza++, SRI LM, etc.). SMT training jobs will be running as batch-processes.
- Running the translation systems (Moses decoder and supporting text processing tools). These jobs will be executed as service (daemon) jobs.

The Application Logic Layer will be responsible for setting up a training job on the HPC Cluster (providing the input files, configuring the job script), monitoring its progress, and finishing the task (copying the results and reports, terminating the processes) when necessary. The HPC Cluster will be responsible for allocating computing resources for the job and monitoring and reporting its progress.

The HPC Cluster will accept jobs for execution on both Linux and Microsoft Windows operating system. The majority of training tasks will be running on machines with Linux OS (Giza++, Moses and SRI LM processes), but there might be some processes requiring Microsoft Windows environment (for example, for some languages there are part-or-speech taggers running only on Microsoft Windows). Hardware infrastructure necessary for SMT training will be easily scalable; it will be an array of identically configured machines and it will be possible to increase capacity of training and translation just by adding more hardware. The proposed HPC Cluster software (Sun Grid Engine) supports also dynamically allocated computing resources in Amazon EC2 cloud.

The translation job script will prepare its environment by accessing the Resource Repository, launch and track the execution of additional jobs to parallelize the training. The models produced by training tasks (translation models and language models) are files in a specific format (not data in database) and will be stored in Resource Repository as files. There will be also metadata associated with trained models and these metadata will be stored in SQL database which will allow using of metadata for browsing and searching through Resource Repository.

The SMT systems trained on the LetsMT! platform will be managed by the Application Logic Layer. The LetsMT! webpage, the widget for web page translation, and interfaces for integration in CAT tools will all interface this layer to access particular SMT system. Currently the SMT decoder Moses implements a batch-process which typically translates several thousand sentences in over an hour - which is the standard use of the decoder in an academic research setting. Server mode will be added to the decoder that responds quickly to TCP/IP requests to enable online translation. The Application Logic Layer will receive translation requests from users and redirect them to translation systems running on several machines and/or processor cores. Running translation systems on several machines is necessary to ensure fast processing of many simultaneous translation requests. Running translation systems on many machines will also make system scalable; it will be possible to add new machines when LetsMT! system receives more requests than can be processed with the currently assigned computational power.

The Application Logic Layer also will be responsible for loading and unloading of translation systems. A new instance of translation system with a particular translation configuration (language pair and domain) will be loaded when a current workload for decoders serving a particular route is getting too high and unloaded/terminated when the particular translation configuration is not needed any more or less computational power is sufficient to serve it. Management will be done dynamically depending on available hardware resources and nature/amount of translation requests. Such dynamic management of loaded translation systems will ensure fast, scalable and resource efficient translation process.

2.7 SQL Database

SQL database will store information about LetsMT! system's resources such as uploaded corpora, trained SMT systems, and processing tools. In addition it will store information about the users and LetsMT! service subscribers, which will use the resources.

Uploaded and processed training data, SMT systems and processing tools will be stored as files in the resource repository. Metadata associated with these resources will be stored in the SQL database, which will allow using metadata for browsing and searching through the Resource Repository.

The database will be used mainly to store and manage resource metadata and user access control information. But it will also be used to store information about system configuration, current state and various logs and reports.

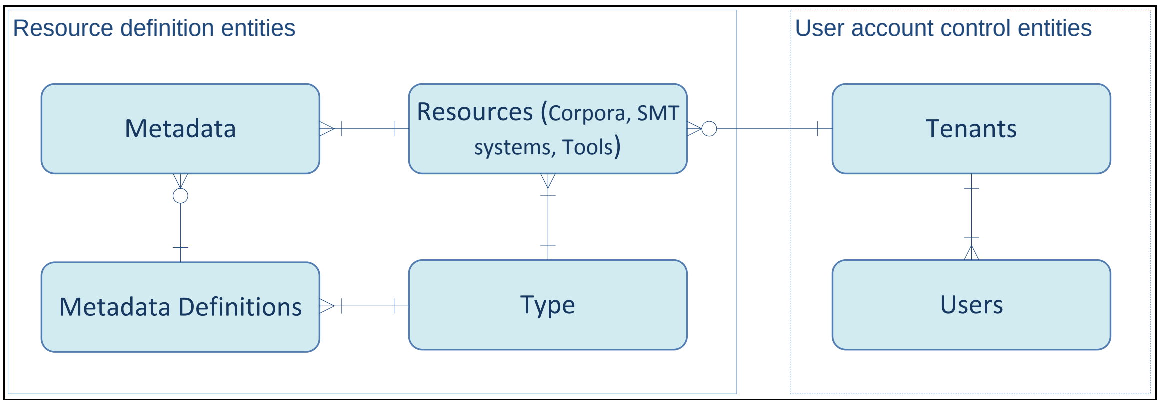

Resource definition entities
User account control entities
Metadata
Resources (Corpora, SMT systems, Tools)
Tenants
Metadata Definitions
Type
Users

Figure 17. Conceptual ER model of LetsMT! SQL database

2.7.1 Resource and Type entities

The resource entity will store information about uploaded corpora, trained SMT systems and processing tools.

Several types of different resources will be stored, i.e.:

- Parallel corpora,
- Monolingual corpora,
- SMT systems,
- Processing tools.

Resource types will be enumerated in type entity including describing information like name and description.

Each resource type will consists of different metadata sets as different properties are relevant to different types of resource.

2.7.2 Metadata Definition entity

LetsMT! metadata will provide administrative information, such as version of processing tool, source and target language of parallel corpora, training status of SMT system, physical location of resource etc.

All LetsMT! system's resources have different metadata information, so the Metadata Definition entity stores predefined metadata definition set for each resource type.

Metadata definition consists of:

- Name,
- Description,
- Resource type,
- Value type,
- Default value,
- Whether particular metadata is mandatory.

Typically each resource (except tools) has its owner (LetsMT! service subscriber) and physical location in resource repository. Each of resource type consists of additional metadata described later in the paragraph.

2.7.2.1 Metadata of SMT system type resource

SMT system type resource additionally will contain metadata as follows:

- source and target language,
- name of particular SMT system,
- domain,
- parallel corpora (i.e., training, development and evaluation corpora),
- monolingual corpora,
- translation model processing tools,
- language model processing tools,
- tools for particular SMT system exploitation (i.e., decoder, data pre and post processing),
- owning LetsMT! service subscriber,
- build state (i.e., queued, training, ready),
- deployed instances,
- whether particular SMT system is public.

2.7.2.2 Metadata of monolingual corpora type resource

Parallel corpora type resource additionally will contain metadata as follows:

- source language,
- name of corpora,
- text domain,
- text type,
- data owner,
- data provider,
- upload date,
- text production timestamp,
- Whether it is public or not.

2.7.2.3 Metadata of parallel corpora type resource

Parallel corpora type resource additionally will contain metadata as follows:

- source and target language,
- name of corpora,
- text domain,
- text type,
- data owner,
- data provider,
- upload date,

- text production timestamp,
- alignment type,
- Whether it is public or not.

2.7.2.4 Metadata of tools type resource

System will contain different processing tools and even different versions could be stored for each processing tool, i.e.:

- Moses training scripts and tools,
- Moses decoder,
- SRILM / IRSTLM / RANDLM language model tools,
- pre and post processing tools.

Metadata of tools will be limited to

- name,
- version
- and physical location in resource repository.

2.7.3 Metadata entity

Metadata entity stores actual metadata values of particular resource instances.

2.7.4 LetsMT! service subscriber entity

LetsMT! service subscriber entity stores information about user/organizations (LetsMT! service subscribers), which will have their own user base, corpora set, trained SMT systems, computing and storage resources.

Following data will be stored:

- Information about LetsMT! service subscribers organization (also could be individual instead of organization);
- Computing and storage resources available on LetsMT! platform.

2.7.5 User entity

User entity stores information about users of LetsMT! platform that contain information as follows:

- Personal information (name, surname, phone number etc.),
- Login and password information,
- Authorization role (see Chapter 1.4.2. User Roles, Accounts),
- Owner or LetsMT! service subscriber.

Each user is a member of one LetsMT! service subscriber. Typically users can manipulate with resources within its LetsMT! service subscriber and public resources regarding on role permissions of the user.

2.7.6 Non-functional requirements

The amount of metadata stored in the SQL database will not be big, therefore, scalability is not an issue.

2.8 *Software Components*

LetsMT! system will be built on top of many existing software packages providing the necessary functionality. This section outlines what software will be used, what adaptations to the existing software are planned and what software will be developed specially for LetsMT! needs.

2.8.1 Statistical Machine Translation

The core functionality of LetsMT! system is SMT training and running of SMT engines. In recent years SMT has provided a major breakthrough in development providing a cost effective and fast way to build SMT systems. This development was particularly facilitated by the open-source corpus alignment tool GIZA++[12], the MT training and decoding tool Moses[13] and other tools. The LetsMT! system will be mainly based on these existing tools. The Moses toolkit will be the core of the LetsMT! platform and it will be enriched with new features, which have been researched but still are not included in the Moses toolkit.

2.8.1.1 Alignment

The first step in building SMT translation models from parallel corpora is automatic word alignment. This part of the process is especially complicated and requires a great deal of computational power especially for large-scale corpora. Standard word alignment for SMT are the IBM models and the HMM alignment model implemented in the freely available tool GIZA++. It can be used as a black-box tool in connection with the Moses toolkit which supports all the necessary steps to build a phrase-based SMT system from a given sentence aligned parallel corpus. The word alignment is carried out in an unsupervised way using EM re-estimation procedures and a cascaded combination of alignment models. Various settings can be adjusted in the alignment procedure and phrase table extraction. The word alignment is time consuming and requires large amounts of internal memory for extensive data sets. Fortunately, there are extensions and alternative tools available with improved efficiency. MGIZA++ is a multi-threaded version of GIZA++[14] and it can run several word alignment processes in parallel on a multi-core machines. Furthermore, the same author provides a cluster-based version of GIZA++ that can be used to distribute word alignment over various machines. An alternative approach that can also run a parallel alignment procedure is implemented in the MTTK toolkit[15]. We expect some users to add relatively small amounts of additional training data in frequent intervals. The incremental training benefits from this data without re-running the entire training pipeline from scratch. Since frequent batch retraining is computationally demanding we will introduce a fast incremental alternative using an online version of the EM algorithm (Levenberg et al., 2010).

2.8.1.2 Training

A significant breakthrough in SMT was achieved by the EuroMatrix project[16]. Among the project objectives were translation systems for all pairs of EU languages and the provision of an open source MT technology including research tools, software and data. The project resulted in the improved open source SMT toolkit Moses developed by the University of

[12] GIZA++: http://fjoch.com/GIZA++.html
[13] Moses: http://www.statmt.org/moses/
[14] MGIZA++: http://www.cs.cmu.edu/~qing/ and http://code.google.com/p/giza-pp/
[15] http://mi.eng.cam.ac.uk/~wjb31/distrib/mttkv1/
[16] http://www.euromatrix.net/

Edinburgh. The Moses SMT toolkit is a complete translation system distributed under LGPL (Lesser General Public License). Moses includes components needed to pre-process data, train language models and translation models. Moses is widely used in the research community and has also reached the commercial sector. While the use of the software is not closely monitored (there is no need to sign any license agreement), Moses is known to be in commercial use by companies such as Systran, Asia Online, Autodesk, Matrixware, Translated.net. The EuroMatrix project has demonstrated how open source tools and publicly available data can be used to generate SMT systems for all language pairs of EU official languages.

The LetsMT! project extends the use of these existing state-of-the-art SMT tools enabling users to build custom tailored SMT systems through simple web based interface. LetsMT! will use Moses as a language independent SMT solution and integrate it as a cloud-based service into the LetsMT! online platform.

Important advancement of the LetsMT! will be an adaptation of the Moses toolkit to fit into the rapid training, updating and interactive access environment of the LetsMT! platform. The SMT training pipeline implemented in Moses currently involves a number of steps that each require a separate program to run. In the framework of LetsMT! this process will be streamlined and made automatically configurable given a set of user-specified variables (training corpora, language model data, dictionaries, tuning sets). The SMT training process will be based on improved Moses Experiment Management system[17].

2.8.1.3 SMT Decoder

LetsMT! system will be based on Moses SMT decoder. We are considering improving Moses decoder to better suite LetsMT! requirements. First the Moses decoder will be adjusted to work with (i) incrementally built and suffix array based translation models (more in chapter 2.8.1.4 Translation Models) and (ii) stream-based randomized language models (more in chapter 2.8.1.5 Language Models).

The current implementation of the Moses decoder has deeply integrated language and translation model functionality. Models are loaded into the Moses process meory. It means that we have to run the whole decoding system in one process on one machine. The machine must have sufficient amount of RAM and disk space, and the Moses decoder has a long launch time as it must load models in launch time. It is being considered to implement TM and LM server support in Moses decoder.

2.8.1.4 Translation Models

An additional important improvement of Moses that will be implemented as a part of LetsMT! is the incremental training of translation models. These advancements will be based on an online version of the EM algorithm (Levenberg et al., 2010) for the word aligning and suffix arrays (Chris Callison-Burch et al., 2005) for translation models.

2.8.1.5 Language Models

Currently the Moses decoder works with the following language models:

- SRI language modeling toolkit[18]
- IRST language modeling toolkit[19]

[17] Experiment Management System: http://www.statmt.org/moses/?n=FactoredTraining.EMS
[18] SRILM: http://www.speech.sri.com/projects/srilm/

- RandLM language modeling toolkit[20]

IRSTLM toolkit compared to SRILM handles LM formats which permit to reduce both storage and decoding memory requirements, and to save time in LM loading. RandLM allows building of the largest LMs possible (for example, a 5-gram trained on several billions of words, such as the whole of the Gigaword Corpus).

Since we are expecting users to add new amounts of additional monolingual training data in frequent intervals, we will need to retrain language models frequently. All the mentioned LM toolkits currently supported in Moses are batch-based and retraining is computationally demanding. Stream-based Randomized Language Models (Levenberg et al., 2009) will be integrated in Moses toolkit to streamline building of frequently increasing language models.

In LetsMT! we will use language model software depending on size and nature of data used to train the model.

2.8.2 Processing of training data

As user-provided shared data plays a major role in LetsMT!, the project should deal with issues related to processing of noisy data and ensuring data interoperability. The component of data management will therefore include various tests and pre-processing tools to validate the data and fix potential errors. There are various ready-made tools that can be used out-of-the-box for data checking. For example, freely available XML parsers, e.g. Libxml2[21], Tidy[22], and TMX[23] validators will be used to detect problems in translation memories provided by users; open-source GNU/Unix tools will be used to detect character encodings and perform conversions; language guessers, e.g. TextCat[24], are also available and can easily be trained for additional language/character sets. It is feasible to use existing tools and integrate them in the LetsMT! platform in order to detect and correct potential errors.

Besides basic validation, the LetsMT! platform requires a number of other pre-processing steps. The most important is a proper tokenisation module, since the majority of users will not provide segmented data. Tokenisation is a non-trivial task and highly language dependent. In the first phase, simple standard tools will be applied that split punctuations from other tokens, e.g. pattern-based tokenisation with the tools provided together with the Europarl parallel corpus. Language specific tools will be used where they are available. Tools for better support of language specific issues will be continuously incorporated, e.g. morphological analysers and lemmatisers.

Initially, only user-provided translation memories containing aligned single-sentence units will be supported. Sanity checks should be carried out to avoid unreasonable training examples such as very long and fragmented translation units or sentences with formatting mark-up or other types of non-textual contents. At a later stage, LetsMT! will also support an upload of other types of parallel data. The idea is to use existing resources in various formats and allow users to create their own training material in the form of sentence aligned corpora.

[19] IRSTLM: http://sourceforge.net/projects/irstlm/
[20] RandLM: http://sourceforge.net/projects/randlm/
[21] http://xmlsoft.org/
[22] http://tidy.sourceforge.net/
[23] http://www.maxprograms.com/products/tmxvalidator.html
[24] http://www.let.rug.nl/vannoord/TextCat/

Support for a number of the most used formats should be provided and the validation process ensured. Standard approaches to automatic sentence alignment are readily available, e.g. Hunalign[25], Vanilla[26], GMA[27]. Post-editing interfaces will be included to verify and improve alignment results online, e.g. ISA as a part of Uplug[28]. In this way, more users will be encouraged to provide parallel data in a variety of formats.

User uploaded content will be stored in internal representation and all uploaded files (in DOC, PDF, RTF, HTML and other formats) will be converted to textual form and aligned in a sentence level. LetsMT! we will also support the upload of parallel documents which have not been aligned at the sentence level yet. This implies that online alignment has to be integrated into the LetsMT! platform. Adding this functionality is a rather complex task. Several issues have to be considered starting from conversion and text extraction from various document formats, sentence boundary detection, tokenization and other necessary pre-processing steps up to the actual sentence alignment. It is planned to support various upload formats but we will concentrate on robustness in order to create reliable data resources in our platform. We will, therefore, only include formats for which we see the largest demand. According to the requirement analysis in D1.1 this will be the DOC formats used in Microsoft Office (various versions), PDF and possibly plain text format with explicit specification of character encodings.

2.8.2.1 Microsoft Word DOC

The largest problem with Microsoft Word files is that the DOC format is a proprietary format with no open standard specifications. Microsoft has the freedom to change and adjust this format as often they like and this has happened frequently in various versions of the Office package. Word documents include a lot of information that is difficult to handle such as revisions, formatting, styles, embedded objects etc. However, any file format that Microsoft Word 2007 (and newer) can open and save as structured XML documents (using Open XML[29]) can be parsed with standard tools. We will support those files trying to extract plain text from the structured information.

During the implementations restrictions with regard to versions will be investigated. We will also investigate the use of available tools such as *antiword*[30] to include experimental support for older versions of Word documents. Another option is Apache Tika[31] which can handle various file formats, for example Microsoft Office documents. Yet another toolkit for processing MS Word files is wvWare[32] which is used for conversion (Word import) in various word processors such as AbiWord and KWord. AbiWord[33] itself can also be used for converting files on the command line. Converting Word documents to text could be done with a command like this: `abiword –to=txt document.doc`. Various other formats are supported, for example, rtf, utf-8, html, latex (all formats supported by AbiWord itself). This could be a valuable option. AbiWord should be rather up-to-date with recent versions of Microsoft Office and probably supports older versions as well.

[25] http://mokk.bme.hu/resources/hunalign
[26] http://nl.ijs.si/telri/Vanilla/
[27] http://nlp.cs.nyu.edu/GMA
[28] http://www.let.rug.nl/~tiedeman/Uplug/php/
[29] Open XML: http://www.microsoft.com/standards/openxml/, http://en.wikipedia.org/wiki/Office_Open_XML
[30] Antiword: http://en.wikipedia.org/wiki/Antiword
[31] Tika: http://tika.apache.org/
[32] WvWare: http://wvware.sourceforge.net/
[33] AbiWord: http://www.abisource.com/

Alternatively a running instance of Word could be used in some kind of batch processing mode. However, it is not clear how robust such a solution would be that has to run on a separate Windows server. It is definitely not build to handle a queue of incoming conversion requests and a work-around might not be very stable.

2.8.2.2 Portable Document Format (PDF)

PDF in general is hard to process. It was originally a proprietary format introduced by Adobe Systems but has been officially released as an open standard in 2008 (ISO/IEC 32000-1:2008). Several versions are around ranging from 1.0 up to 1.7. PDF can be seen as a container format that may include various types of objects. PDF documents may include images (vector graphics and raster images), text (including information about fonts and encodings), interactive elements (forms), file attachments and meta-data.

For LetsMT! only text objects will be interesting. We have to ignore PDF documents that include textual information in terms of images (scanned pages for example). Even extracting running text from PDF is not simple. Multi-line sentences are at risk of being separated. Tables, frames and pictures split the text. Character encodings, especially for non-western languages are problematic. We will present a PDF extraction solution which will be lossy and may compromise the document integrity. The user will be warned. PDF content consisting of images of scanned documents will not be processed.

Our solution will be based on available software. Several PDF readers and converters are around. We have already experience with `pdftotext` which is part of the `xpdf`[34] package. The tool is quite robust and supports several text encodings such as ISO-8859-1 (Latin 1), ISO-8859-2 (Latin 2 for Eastern European Languages), ISO-8859-7 (for Greek), KOI8-R (Cyrillic) and ISO-8859-8, ISO- 8859-9 (for Hebrew and Turkish). There is also an option '-layout' that can be used to maintain the physical layout of a document and in some cases this can be useful to improve segmentation tasks such as sentence boundary. However, layout information may cause problems to identify coherent text blocks especially in multi-column layouts. Some experiments showed success by using simple post-processing for column detection and table conversion. However, LetsMT! users who want to upload PDF documents should have the chance to adjust parameters in order to improve extraction results.

Other tools that we will investigate for the extraction of plain text from PDF documents are listed below:

Multivalent: http://multivalent.sourceforge.net/

PDFBox: http://pdfbox.apache.org/

Tika: http://tika.apache.org/

pdftoxml: http://pdftoxml.sourceforge.net/

Poppler: http://poppler.freedesktop.org/ (GPL version of Xpdf)

pdftohtml: http://pdftohtml.sourceforge.net/

pdftoword: http://www.pdftoword.com/

[34] xpdf: http://www.foolabs.com/xpdf/

2.8.2.3 Open Document Format (ODF)

The Open Document Format[35] is another XML-based file format for representing electronic documents in office applications. It is used in various open source and proprietary software. It is even supported by Microsoft Office 2010 and its growing popularity makes this format a good candidate to be added if the demand increases. The biggest advantage of ODF is that it is an open standard and specifications are available. However, it still requires quite some effort to handle the complex structures of possible ODF documents correctly.

2.8.3 Operating systems, frameworks and servers

The main LetsMT! functions are sharing and processing of training data, MT training and MT running. All these functions will be implemented by integration and adapting existing tools which are designed for Linux platform. It means that all modules in data storage layer and HPC cluster will be running in Linux environment.

HPC Cluster will be used to execute many different data processing and SMT training and running tasks. Modules from the application logic layer will create jobs and send them to HPC cluster to execute. LetsMT! HPC cluster will be based on Sun Grid Engine[36] (SGE) running on Linux platform.

LetsMT! database will be developed using SQL server. It is important to make database portable and scalable and easy deployable in different hardware infrastructures. For example we might have local servers for test and development environments and cloud based hardware solution for public use. Decision about SQL server is not made yet, it will be done during development, but MySQL or PostgreSQL servers are being considered.

LetsMT! web page, public API and many parts of application logic will be developed using Microsoft .net framework as it provides powerful and easy scalable tools for web application and web service development and Tilde has long experience and skills working with this technology.

2.8.4 Software developed in scope of LetsMT! project

There are areas in LetsMT! system where we cannot use existing software and we will implement new features in scope of the project.

First of all we will implement all functionality of interface layer and application logic layer as this is very project specific functionality. LetsMT! application logic will integrate together all tools which we will use for data processing, SMT training and SMT running.

Many data pre-processing tasks will be based on existing tools such as aligners, file format converters, but we will need to develop tools to process some file formats, to validate results and to integrate all pre-processing tools together in one pre-processing chain or workflow.

2.9 Infrastructure Design

The hardware infrastructure of LetsMT! platform will be heterogeneous:

1) The majority of services will be running on Linux platform (Giza++, Moses, data processing tasks);
2) Other services will run on Windows platform.

[35] ODF: http://www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.0-os.pdf
[36] SGE: http://gridengine.sunsource.net, http://en.wikipedia.org/wiki/Sun_Grid_Engine

System hardware architecture will be designed to be highly sizable. The LetsMT! platform will contain several machines with both continuous and on-demand availability:

1) Continuous availability – the core frontend and backend services that guarantee LetsMT! webpage and external API availability;
2) On-demand availability – training and translation services (HPC cluster nodes).

LetsMT! might provide some translation services with continuous availability as well.

2.9.1 Deployment within Amazon Web Services

The Consortium, instead of buying servers, intends to lease capacity. This is economically efficient and will provide flexibility in adding new resources exactly when necessary.

The LetsMT! platform is planned to deploy completely within the Amazon Web Services (see Figure 3.2) as this is tested solution. Alternative cloud computing suppliers may be selected if AWS fails to meet requirements of LetsMT! system. Two services from Amazon should be used:

1) Amazon Elastic Computing Cloud (EC2) for LetsMT! platform servers and HPC cluster,
2) Amazon Elastic Block Storage (EBS) for Resource Repository.

Amazon EC2 offers flexible way to acquire on-demand computing instances. These instances can also be reserved for 1 year or 3 year term to reduce operating costs of heavily used instances. LetsMT! instances that require continuous availability should be reserved. On-demand instances will be acquired dynamically from Amazon EC2.

To improve service availability within Europe, the LetsMT! platform should be deployed in the European zone (Amazon Ireland).

The backend server, database server, HPC cluster host and nodes must be protected by placing behind a firewall provided within the Amazon EC2 framework.

EC2 Instance function	EC2 instance type	Amount
Core infrastructure (continuous availability)		
Frontend server	Small	1
Backend server	Small	1
DB server	Small	1
HPC Cluster Master	Small	1
HPC Cluster Node (translation)	Large / Extra Large	0-5
On-demand SMT training		
HPC Cluster Node	High Memory Extra Large	1-2 per training task
On-demand translation		
HPC Cluster Node	Large / Extra Large / High Memory Extra Large	1-10 per translation task

Table 6. Amazon EC2 instances for the LetsMT! platform

The following considerations have been taken into account when estimating the LetsMT! Hardware solution:

- Training of one SMT system (on training data with a size comparable to DGT-TM corpus size, without factoring) takes about 10 hours and 5 GB RAM training on one CPU. We are assuming that during the project a typical training task (with more training data and some factors) will be running for about 24 hours, and there will be 2-3 parallel training tasks running and 2-3 training tasks queued. So we will need hardware with 3-5 CPUs and about 10 GB RAM for each CPU to run training tasks during the project.
- Currently one Moses decoder running on one CPU can process 1-2 sentences per second in average; but it depends on complexity of models used in translation. As we will add a server mode to the Moses decoder that responds quickly to TCP/IP requests we expect that performance will increase.
- It is planned that the infrastructure created during the project will be able to serve the 5 most used MT systems with performance of 6 sentences per second and lesser used MT systems will be loaded/unloaded dynamically. Dynamically loaded systems will have a performance of at least 1 sentence per second and we will have 10 simultaneously loaded such system. To ensure such performance we will need hardware infrastructure with about 20-25 CPUs and 5 GB RAM per CPU. With CPU here we mean one processing unit (core). Currently available servers have one or more Dou and Quad core processors. Server with two Quad core processors can process 8 times more than with one processing unit. If we would build translation hardware infrastructure using servers with 2 Quad core processors we would need 3-4 such servers with 40 GB RAM each.

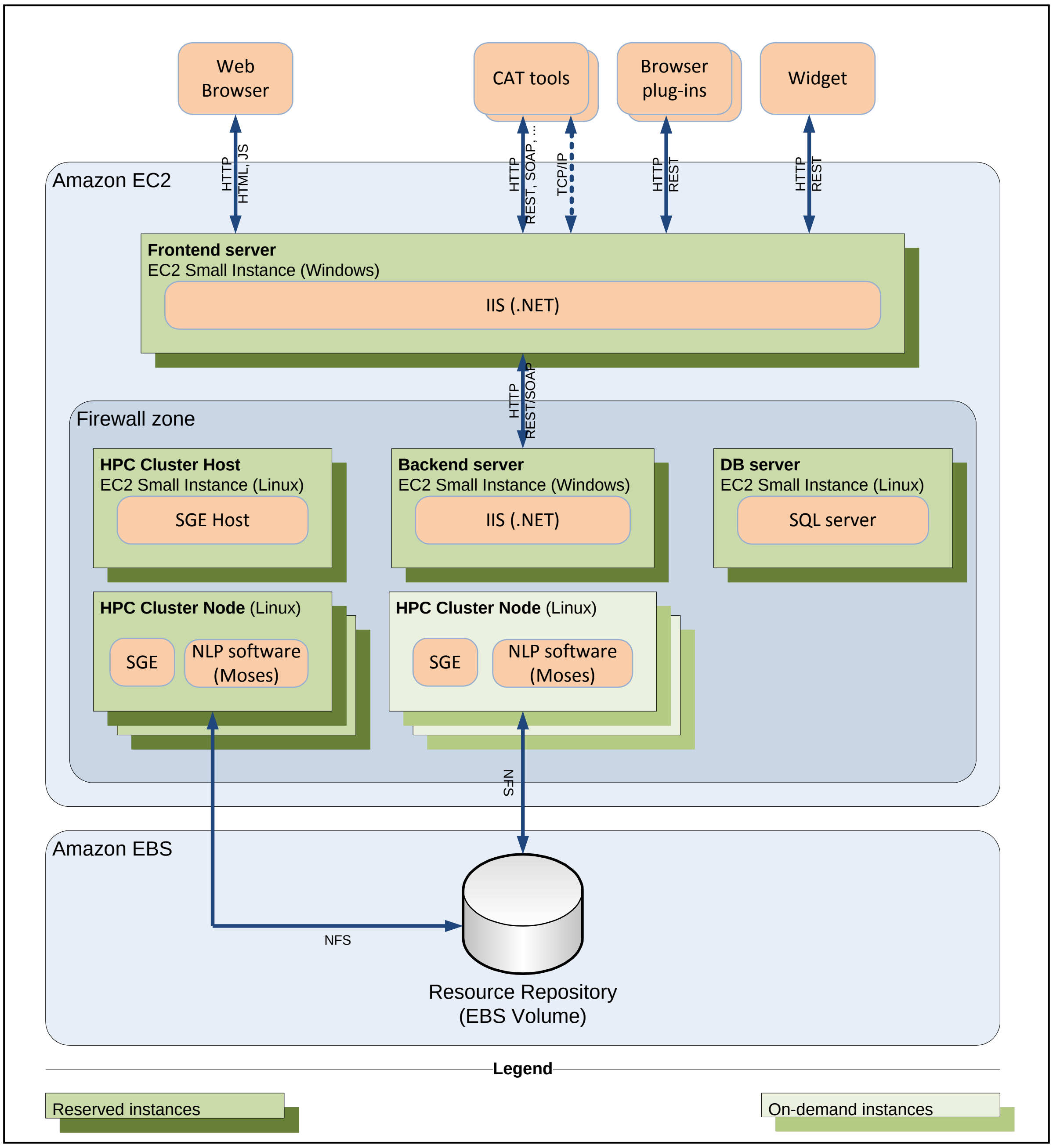


Figure 18. Hardware architecture of LetsMT! platform

References

Abby Levenberg, Chris Callison-Burch and Miles Osborne. 2010. Stream-based Translation Models for Statistical Machine Translation. NAACL, Los Angeles, USA.

Abby Levenberg and Miles Osborne. 2009. Stream-based Randomised Language Models for SMT. EMNLP, Singapore.

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 255–262, Ann Arbor, Michigan, June. Association for Computational Linguistics