Direction Générale de l'Innovation et du Support technologique
Direction du développement et du support
Unité CONCEPT
Le Chef d'Unité

Parlement européen

Luxembourg, 12 September 2016

GEDA D(2016) 39778

Consortium CGI Luxembourg S.A. – Intrasoft International S.A.
7, Zone d'Activité du Bourmicht
L-8070 BERTRANGE

E-mail : jean-louis.basso@cgi.com

CC : ITEC-PAC-DMO@ep.europa.eu
CONCEPT-Unit@ep.europa.eu

Subject: Request for an offer
Specific contract ITS14-L3-R1/PE-ITEC-DES-CONCEPT-PROJECTS/CS021
Project: Trilogue

Dear Mr Basso,

I hereby invite you to make an offer for the attached contract statement of work.

In accordance with the provisions of ITS14, please indicate within a period of 3 working days whether you intend to make an offer for the contract statement of work in question.

Should you wish to make an offer, the offer must be presented to the European Parliament within 10 working days of sending of this request.

Your intention to submit an offer, the signed and dated offer as well as possible subcontracting documents shall be sent by e-mail to ITEC-PAC-DMO@ep.europa.eu.

Yours sincerely,

Ludovic Delepine

Head of Unit

Annexes: Contract statement of work
Request for offer form
Three subcontracting documents

L-2929 Luxembourg - Tél +352 43 00 22091
F-67070 Strasbourg - Tél +33 3 88 1 72640
ludovic.delepine@europarl.europa.eu - www.europarl.europa.eu

ITS14 - REQUEST FOR OFFER FORM

Dear Contractor,

I hereby invite you to make an offer for the attached Contract Statement of Work which is summarized below:

Request for offer reference: D(2016)39778

Specific Contract reference: **ITS14-L3-R1/PE-ITEC-DES-CONCEPT-PROJECTS/CS021**

Estimated starting date: **01/10/2016**

Estimated end date / Duration: **15/02/2017**

Estimated budget (if known): **- €**

Type of specific contract: Quoted Time and Means

Planned work outside normal working hours: No

Planned missions to other places of work: YES (Luxembourg/Strasbourg)

Regular place(s) of work *(not including occasional missions to other places)***:**

	On-site	*Near-site*	*Off-site*	*On-call*
Brussels		***x***		
Luxembourg				
Strasbourg				
Other:				

Justification "near-site" :

- Présence requise pendant la phase d'étude ou d'analyse des besoins pour les projets de développement d'applications
- Contribution à des projets en cours de développement nécessitant une présence soutenue à des réunions de coordination autres que les réunions périodiques de suivi de projet qui peuvent faire l'objet d'une planification, voire se faire en visioconférence.

Administrative contact point: dgitec-marches@ep.europa.eu

Technical contact point: olivier.leboeuf@europarl.europa.eu

In accordance with the provisions of ITS14, please indicate within a period of 3 working days whether you intend to make an offer for the Contract Statement of Work in question.

If yes, the offer must be submitted to the European Parliament within 10 working days from receipt of this request. The European Parliament will then have 10 days in which to provisionally confirm acceptance.

Your signed and dated offer shall be sent by E-Mail **ITEC-PAC-DMO@ep.europa.eu**

Directorate-General for Innovation and Technological Support
Directorate for Development and Support
Unit ICT Conception & Development

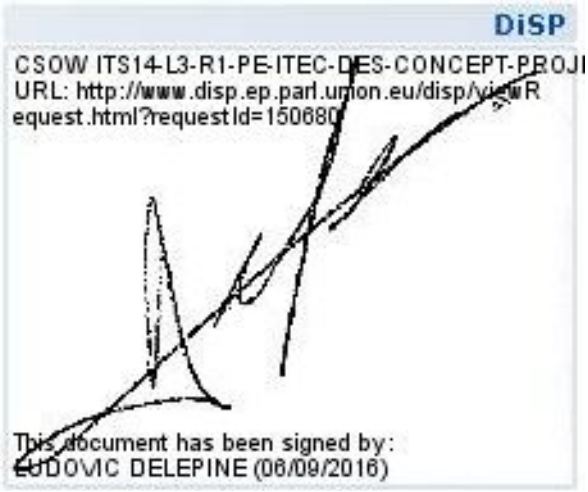
European Parliament

Contract Statement of Work

Quoted Time & Means

"Trilogue Tables Editor"

ITS14-L3-R1/PE-ITEC-DES-CONCEPT-PROJECTS/CS021



Template last updated: 21/5/2015-FP

Document name	CSOW_CS021_Trilogue Tables Editor 0.5	Number of pages	10
Created on	29/02/2016	last updated	06/09/2016
Document status	FINAL	Version	0.5
Author	Gianluigi Alari, Tuukka Lindberg, Olivier Leboeuf		

APPROVAL

Name	Approval date	DG or Company
Ludovic Delépine	See digital signing	DG ITEC
Olivier Leboeuf	See digital signing	DG ITEC

CIRCULATION

Name	Role	DG	ERICA[1]
Gianluigi Alari		DG ITEC	C
Tuukka Lindberg		DG ITEC	R

UPDATES

V	Date	Changes	Description	Author
0.1	1/7/2016	-	Version for Leboeuf review	Alari, Lindberg
0.2	15/7/2016	-	Leboeuf initial review	Leboeuf
0.3	8/8/2016	-	Minor updates	Lindberg
0.4	9/8/2016	-	Alari's comments, document finalization	Lindberg
0.5	30/8/2016		Wording of end date	Leboeuf

[1] **E**: Examination, **R**: Responsible, **I**: Information, **C**: Contribution, **A**: Approval

TABLE OF CONTENTS

1. GENERAL INTRODUCTION

1.1. Purpose

The purpose of this "Contract Statement of Work" document is to define the contract framework, in terms of functional and technical requirements, management processes and acceptance criteria, in order to enable the supplier to decide whether he will make an offer.

1.2. Applicable Documents

This Chapter details the applicable documents used in drafting this document. The applicable documents are standards, specimen plans which need to be applied and whose application is verifiable.

N°	Title
[1]	ITS14 Framework Contract (with its annexes)
[2]	Delivery Note Template

1.3. Appendixes

This CSOW document has the following appendixes.

N°	Title
[A1]	Specific Requirements for Development Work – v1.5.1 - 26/05/2016

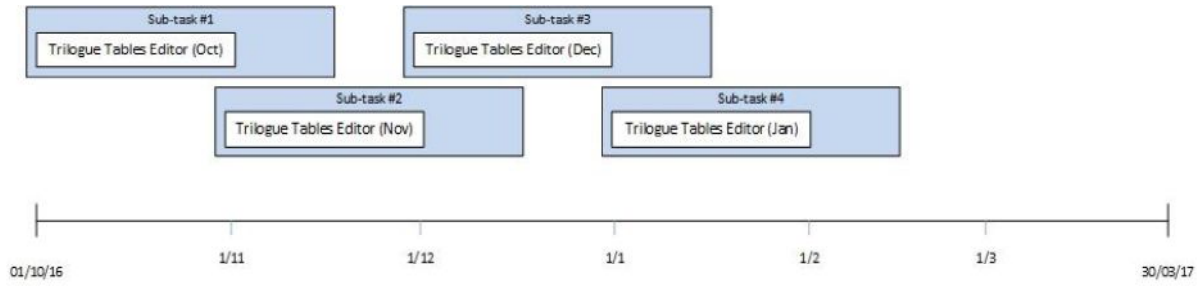
2. SUBJECT

The subject of this contract is to provide EP with artefacts and work related to the near-site *Services* for PROJECTS department. The *Services* to be provided are:

1. Functional Analysis and Architecture area:
    a. Gathering, analysing, modelling, verifying and managing requirements (both functional and non-functional);
    b. Translating business-level requirements to system-level requirements, application architecture and end-to-end solution design;
    c. Contributing to the design of the critical components and patterns of the defined solution architecture;
    d. Following, counselling and verifying the development of proof of concept versions of the defined solution architecture;
    e. Authoring, reviewing and maintaining requirements, functional architecture and solution design documentation, systems architecture recommendations and solutions, technical specifications;
    f. Creation of screen wireframes and mock-ups;
    g. Design, document and execute test strategy, scenarios and scripts in collaboration with the Test Team;
    h. Assisting the project manager during key users' acceptance tests;
    i. Assisting the project manager in drafting change requests documents during the project execution and up to the end of early life support;
    j. Authoring of user manuals and training documentation as well as supporting end-user training sessions;
    k. Assisting the project manager in scope management, scheduling, estimation and documentation areas;
    l. Providing assistance to the project at large (project manager, key users, UI experts, developers, testers, outsourced development teams) by giving expertise and direction mainly in the functionalities and requirements areas.
2. e-Parliament - Trilogue Tables Editor development area. Providing software development, architecture prototyping and design, testing and deployment, production support and other related activities to the Trilogue Table Editor project of the e-Parliament program.

3. SCOPE OF THE SPECIFIC CONTRACT

The contracted work will be split into several sub-tasks. Each sub-task will contain a subset of the *Services* mentioned in the previous Chapter. The following picture depicts the sub-tasks decomposition (note that the dates in the picture are used as an example only).



As the Contractor is able to finalize some deliverables only after the work has been done, Sub-tasks #1 - #3 last 1,5 months. For example, within the Sub-task #2 the *Services* are being provided during November only, but for example the deliverable "Service Enhancement Plan", if that is to be written, shall be delivered to EP by mid-December. Sub-task #4 may be a bit shorter or longer than the other Sub-tasks.

The Contractor shall provide the *Services* only for the projects and activities within the PROJECTS department.

The following work is out of the scope of this *SC*:

- A transition from EP - or from the existing Contractor(s) providing services described above - to the new Contractor at the beginning of the *SC* is not specifically requested by EP.
- Functional Analysis *Service* excludes Business case development and project management

4. DELIVERABLES

A detailed list of the deliverables can be found in the sub-task forms.

All deliverables are to be written in English, unless otherwise agreed in written form with EP. Examples of deliverables are (non-exclusive list): source code, JavaDoc documentation, binaries, technical specifications (DTA, STR, STS, designs, sketches, prototypes,...), functional specifications (use cases, user group definitions, SFS...), testing related deliverables (MTP, test cases, performance testing procedures, code compliance reports, newtest probes, ...), guides (user, installation, operating), user training materials, application mock-ups, ARIS process and application models, deliverables related to meetings and trainings (agendas, presentations, minutes, training reports...), wiki updates, other reports (see Chapter 6.6) ...

5. KEY ACTIVITIES

EP expects that the Contractor's resources perform the following key activities during the SC execution:

- Periodical status and ad-hoc meetings held at the EP premises (or over videoconference connection if agreed by both parties) as deemed necessary by the EP.

- Exchange of information face-to-face with EP officials and consultants, as deemed necessary by EP
- Work status and progress reporting with the tool(s) EP is using (currently Jira).
- *Service* specific activities defined in the Sub-task form.

The Appendix "Specific requirements for development work" [A1] defines some key requirements for technical deliverables.

Whenever applicable the Contractor will have to produce documentation following specific PMM4EP templates.

6. ORGANIZATION AND HUMAN RESOURCES

6.1. Work location

The place of work for this SC is in Brussels. The work needs to be done near-site.

6.2. Required service availability

In order to confirm smooth delivery of the *Services* contracted and good communication/co-operation between different teams, Contractors consultants should in general follow the working hours of EP officials.

In case of near-site assignments, one EP official or any other person mandated by EP and working in the context of the Trilogue Tables Editor project may be occasionally visiting the Contractor's premises, where work related to the requested *Services* is executed. On these occasions, the Contractor is required to provide the visitor facilities that would allow them to work together with the Contractor's team.

The *Services* provided by the Contractor need to be available during *Normal working days*. EP will define the requested exact *Service hours* in the Sub-task form. Normally the *Service hours* are one of the following:

- *"Both AM and PM"*: the *Service* will be available during the whole day (1 day)
- *"Only AM"*: the *Service* will be available only in the morning (1/2 day)
- *"Only PM"*: the *Service* will be available only in the afternoon (1/2 day)
- *"No service"*: none of the categories above apply

6.3. Contractor's resources

The following table describes the profiles/roles and the number of working hours requested by the EP.

Service	Profile/Seniority level	Requested hours	Comments
Functional Analysis and Architecture area	PANL-SL3	550 h	Service in Brussels

e-Parliament - Trilogue Tables Editor development area	PDVS-SL3	550 h	Service in Brussels
	Total	1 100 h	

6.4. Administrative EP resources

The following table describes the persons EP has assigned to specific administrative tasks (see "Annex II.5 to FC – Service Level Agreement" [1]).

Role	Person	Description
Specific Contract Manager	Olivier Leboeuf	Head of Department.
Service Manager	Tuukka Lindberg	Monitors Specific Contract implementation.
Project Manager	Gianluigi Alari	EP official responsible for this project. Primary contact person for technical matters.

6.5. Meetings

For this Specific Contract, a "Committee based follow-up service meetings organisation" (see "Annex II.5 to FC – Service Level Agreement", [1]) will take place. In order to limit the administrative overhead, a monthly "ITS14 Technical Committee" will normally discuss all the active common projects/services the Contractor is providing to the PROJECTS department.

6.6. Reporting

EP may request the Contractor to provide it with detailed ad hoc reporting regarding the *Services* provided and/or resources providing the service. This could take place e.g. in a situation where EP is not satisfied with the service quality and wants to follow-up the services more closely.

7. DELIVERIES AND ACCEPTANCE

A delivery will take place at the end of each Sub-task. The Contractor needs to provide EP with the following documents with each delivery:

- Delivery note: This document will be approved by the EP once it has been verified that all the individual deliverables in the list have been accepted. The EP may refuse the whole delivery if it is not complete. The "Delivery Note Template" [2] is available as an applicable document.
- Sub-task form: When EP accepts the delivery, it will sign the last page of the Sub-task form (template as Annex I.11 to FC [1]). Signed Sub-task form has to be attached to the Invoice.

Please refer to CSOW appendix "Specific Requirements for Development Work" [A1] for more information on how the specific IT deliverables shall be accepted.

8. PLANNING

If the Contractor is not able to meet the scheduling and milestones defined in this Chapter, it has to state it clearly in its offer.

8.1. Specific Contract scheduling

EP foresees the following dates for the Specific Contract.

	Date
Desired start of the work	1st of October 2016
Desired end of the work	15 February 2017

8.2. Milestones

Milestones will be defined in the Sub-task forms.

8.3. Work and Financial limits

The maximum size of this *SC* is 1 100 hours.

9. GLOSSARY

Term	Definition
BPM	Business Process Modelling.
CONCEPT	*DES* Unit for Conception & Development.
CSOW	Contract Statement Of Work, Specific Contract appendix giving exact details about the scope of the work contracted.
DES	*DG ITEC* Directorate for Development and Support
DG ITEC	*EP* Directorate General for Innovation and Technological Support
DTA	Technical Architecture Document. *PMM4EP* document describing the high-level architecture, integrations, applications and tools for some specific project. This document has to be based on a *PMM4EP* template and approved in advance by *EP*.
EP	European Parliament
FC	ITS-14 Framework Contract signed between Contractor and *EP*.
MTP	Master Test Plan. This document has to be based on a *PMM4EP* template.
Normal Working Day	Effective working days at *EP*. See document "Annex II.4 to FC - Service Requirements" for additional details.
PM	Project Manager
PMD	Source code analysing tool being used at *EP*.

PMM4EP	Project Management Method For European Parliament.
PROJECTS	*CONCEPT* Department for Project Management.
SC	Specific Contract signed between Contractor and *EP*.
Service	A set of activities and work that the Contractor shall provide to *EP*. The outcome from a service can be a concrete deliverable (e.g. *SFS* document) or intangible work (discussion with a customer, being present at the office so that customers can call...)
Service Enhancement Plan	A document that will be drafted by the Contractor in case the *SLA Report* document contains at least one *Service*, where *EP*s overall satisfaction is not "Good". The document proposes changes that should be implemented (normally by the Contractor) in order to enhance *EP*s satisfaction to the *Services*.
Service Hours	Time period when a specific *Service* is being provided normally and is to be used by available for *EP*.
Service Management	Activities performed to plan, deliver, operate and control the services being offered to the customers
SFS	Functional Specification document, which gives exact details about the features and functionalities of the work contracted. This document has to be based on a *PMM4EP* template.
SLA Report	A document summarizing *EP*s satisfaction on the *Services* provided by the Contractor.
STR	Specific Technical Requirements document. The document lists major technical requirements, which need to be documented in STS and implemented in the final system/deliverables.
STS	Technical Specification document. Gives exact details about the technological requirements of the work contracted. This document has to be based on a *PMM4EP* template.

Specific Requirements for

Development Work

Appendix to the Contract Statement of Work

APPROVAL

Name	Approval date	DG or Company
Olivier Leboeuf	25 Apr 16	DG ITEC

CIRCULATION

Name	DG or Company	ERICA[1] code
Gianluigi Alari, Maria Ascorbe, Neven Borojevic, Jesus Cerro, Emmanuel Collin, Olivier Fabbri, Olaf Grossman, Jean-Marc Klein, Christopher Scott	DG ITEC	C
Prosper Ayegnon	European Dynamics	C
Tuukka Lindberg	DG ITEC	R

UPDATES

V	Date	Sections changed	Description	Author(s)
1.0	5 May 15	Approval	Default approval	Tuukka Lindberg
1.1	7 May 15	1.2.1	New version of Sonar metrics document	Tuukka Lindberg
1.2	26 Jun 15	Several	Typo corrections	Tuukka Lindberg
1.3	9 Mar 16	1, 2.5, 2.6.1, 5, others	Merged with Sonar Metrics v2.4, typo corrections, Chapter 2.5 added, 2.6.1 re-written and [2] changed, minor changes & restructuring, MA comments.	Tuukka Lindberg
1.4	23 Mar 16	Several	OL comments, Unit test naming convention, finalization. OL approval.	Tuukka Lindberg
1.5	25 Apr 16	2.12	Updates based on CARMEP deployment. OL Approval	Olivier Fabbri
1.5.1	26 May 16	2.12	Added consolidated scripts	Olivier Fabbri

[1] **E**: Examination, **R**: Responsible, **I**: Information, **C**: Contribution, **A**: Approval

TABLE OF CONTENTS

1. GENERAL INTRODUCTION

The purpose of this CSOW Appendix "Specific Requirements for Development Work" is to describe the general technical requirements EP has for information systems being developed. Please note that project specific technical requirements will be documented in other documents (DTA, STS among others).

1.1. Applicable Documents

The applicable documents are standards, which need to be applied and whose application is verifiable.

N°	Title
[1]	ITS14 - IT-environment of the European Parliament Annex II.6 to the Framework Contract
[2]	CWE – Common Weakness Enumeration, A Community-Developed Dictionary of Software Weakness Types – Version 2.9 or newer https://cwe.mitre.org/data/published/cwe_v2.9.pdf
[3]	OWASP Secure Coding Principles https://www.owasp.org/index.php/Secure_Coding_Principles
[4]	OWASP Secure Coding Practices - Quick Reference Guide - Version 2 or newer https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

1.2. Reference Documents

Reference documents help in defining the work but are not always directly applicable (e.g. software reference manuals, sample files from outside firms, meeting papers).

N°	Title
[R1]	EP DEVELOPMENT RECOMMENDATIONS Overview - V1.4 - 09/04/2014 http://www.standardsnet.ep.parl.union.eu/standards/webdav/site/main/groups/Cellule-ALSA/public/EPFoundry/EP-Foundry-Development-Recommendations_EN-v1%204.pdf
[R2]	Object/Relational Mapping (ORM) with Hibernate /JPA - V1.4 - 22/02/2012 http://www.standardsnet.ep.parl.union.eu/standards/webdav/site/main/groups/Cellule-ALSA/public/EPFoundry/Persistance/IMS-EP-Foundry-ORM-Hibernate-JPA_EN-v1.4.pdf
[R3]	Guide Design and Assessment of Graphic Interfaces - V4.0 - 02/2013 http://www.standardsnet.ep.parl.union.eu/standards/webdav/site/main/groups/Cellule-MAEL/public/Guides/ERGO_GU_Software-Ergo_EN.pdf
[R4]	CSOW appendix "Glossary"
[R5]	Recommandations ergonomiques pour la conception de pages Web - v1.1 - 11/02/2013 http://www.standardsnet.ep.parl.union.eu/standards/webdav/site/main/groups/Cellule-MAEL/public/Guides/ERGO_GU_WebErgo_FR.pdf
[R6]	Graphic Web Charter – v 2012

	See also [R7]
[R7]	FA : Ajouts à la charte graphique – v2.0 – 15/07/2013
[R8]	OWASP Testing guide (version 4 or newer) https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

2. TECHNICAL REQUIREMENTS

Any software developed for the European Parliament must comply with the requirements stated in this document, if not overridden in the project specification documents. Any exception to the following recommendations must be agreed with EP in written form.

It should be noted that the EP requirements, standards, and recommendations change over time. The Contractor needs to confirm the newest versions of the EP standards and EP recommendations when the detailed information system specifications or documents are being drafted. For example, the specific tool, library and application versions mentioned in the DTA attached to the CSOW might change during the contracting procedure and thus it's important to confirm the versions when the specific contract has been signed. If there are any changes, the contract parties have to agree on the version to use, which should in general be the newer version.

In the beginning of the project the Contractor has to agree with EP how the needed applicable requirements, standards, recommendations and templates shall be delivered. The options are:

1. An access to a Subversion (SVN) repository (https://ep-foundry.secure.europarl.europa.eu/viewvc), accessible from the outside of the European Parliament. The repository contains the last versions of the applicable requirements, standards, recommendation and templates.
2. The documentation can be delivered to the Contractor on a CD or an USB key.

2.1. IT environment and technical architecture

The technical architecture of the solutions must comply with the IT infrastructure of the European Parliament which is detailed in the applicable document [1].

As default architecture:

- a Java EE n-tier architecture will be used
- a database will be used for data persistence (Oracle database by default)
- an application server will host the application (Tomcat by default)
- the client part will be managed by a web browser (cross-browser compatibility by default)

2.2. Delivery

The delivery of source code and/or documentation will preferably be done through a check in on a Subversion repository, accessible from outside the European Parliament. Each version delivered will have to be tagged with a version number.

In the event that there is no Subversion repository dedicated to the project, or the Subversion repository is not available on the due date of the delivery, the source code and/or the documentation should be uploaded on a secured FTPS repository of the EP. Each version delivered through FTPS will have to be delivered in a subdirectory named with the version number.

2.3. Source code

The source code needed to build and run the application has to be part of the deliverables.

The recommended structure for Java EE projects and the organization of the source code is given in document [R1]. The source code must comply with the rules described in this document. Exceptions should be managed in an appropriate way.[2]

To access the database, the European Parliament recommendation is to use the Object Relational Mapping tool (ORM) Hibernate [R2].

The use of all libraries must be declared in the Technical Architecture Document (DTA) and must be explicitly authorised by the European Parliament.

Libraries, which are available in the European Parliament Teles Standard Repository (http://www.teles.ep.parl.union.eu/repositories/nexus/index.html#view-repositories;teles~browsestorage), are considered as standard, and with normal situations their use shall be accepted. However, all libraries used need to be declared and approved in advance.

The Contractor has to build the application from the code and standard libraries using standard EP tools and versions (ant, maven via POME files…) and provide documentation describing how to build the application. The binaries generated by the compilation of the delivered source code must also be part of the delivery.

2.4. Configuration and properties files

Subdirectories are not supported in European Parliament for configuration and properties files stored in the "/etc" folder.

If the data requires structuring, composed filenames should be used, using "." as separator (for instance: "nameOfTheSystem.whateverService.whateverExtraInfo.properties").

[2] Exception management: Generating an exception has a cost (e.g. generate the stack trace). Exceptions should be generated only in unexpected cases. If a condition is known in advance this condition cannot generate an exception. It's not admissible to have exceptions to exit from a for, while or function. It's not admissible to generate exceptions when the user type wrong data in the application. Expected cases cannot generate an exception.

2.5. Documentation

As a general rule, upon the demand of a concrete deliverable provided with a defined standard EP template, the documentation provided by the Contractor will have to respect it. The delivery of a document in other format will be automatically considered as a non-conformity during the acceptance process. In the cases where no template has been provided, the supplier will be free to choose the way of presenting it, in both terms of content and structure.

Regarding the source code, it shall include the complete documentation in the Javadoc format in English. Non-trivial classes/methods need to be documented with the following tags at minimum: @exception, @param, @serial, @serialData, @serialField, @since, @throws and {@value}. Tags @deprecated, {@inheritdoc}, {@link}, {@linkplain}, @return, @see and @version are proposed to be used when they provide useful information for the maintenance team.

The following annotations or comments are forbidden by default and delivery will be rejected in the event that these are found in the code (if not pre-approved in written form by EP): //NOSONAR.

Comments should also give information in which use cases, if any, the class/method is being used.

As a minimum requirement, any delivery of application, component or module must come with the following documentation (either a new document or an update of a previously drafted document) in .docx format:

- Installation guide
- Setup guide
- User guide
- Technical Architecture document
- Performance scenarios
- Javadoc
- NewTest monitoring document (only for applications)
- Technical specifications
- Sonar execution report
- Unit tests report
- Build guide

These documents are the base for any delivery. Depending on the technology and the architecture of the solution, some other documents can also be requested. All documents and documentation shall be written in English.

2.6. Security

Secure coding principles

In general, the Contractor has to avoid common security issues like javascript injection, cross-site injection, phishing, SQL injection, unauthorized access to information via URL manipulations (e.g. changing IDs), unauthorized access to functions via URL, unauthorized access to admin modules via, unauthorized changes of data, not using antivirus tool for internet applications etc.

The Contractor must follow the industry best practices when developing information systems. As a minimum requirement, the use of general secure coding principles (see [3] and [4]) must be properly implemented and documented by the Contractor. Detailed guidelines how to address several common security issues can be found in [2].

Authentication

The Contractor should use JAAS security standards for the authentication using BASIC authentication. Definitions should be mapped in external files (e.g. Roles in an APP_ RSA.XML). The Contractor can use external User/password_files with JAAS in order to test the authentication.

This easy implementation will allow the EP to switch to internal authentication when the application is deployed. For that JAAS libraries will be replaced/complemented by internal compatible libraries allowing different type of authentication mechanisms (Token, Spnego, EPnet, OTP…), and connect to the internal LDAP to verify the user/password.

The implemented authentication must be agnostic of the internal EP implementation that for security reasons cannot be provided.

JAAS configuration must be done at container level and not at library level, Spring security must be avoided.

Authorisation

Once authenticated, the application users will have access to the application, knowing that only one generic database user will access the database behind this application (standard n-tiers design).

These users will be authorized to access several functionalities and data. Normally the security information will be configured and managed inside of the system, thus the EP Active Directory is not used for this purpose. Role information determined during the authentication could also be used in the authorisation.

In general there are three different ways to grant access rights to the users:

1. Users and their roles/access rights are managed in a database, updated either by application “Superusers” with an application specific GUI, or by the maintenance team.
2. Users and their roles are managed in EP Active Directory. This option should be foreseen only with applications that have limited amount of roles, and where the (DG ITEC) ServiceDesk is able to assign roles for each user.

3. Mixed-model: some generic roles are stored in LDAP (option #2 above), but rights can be fine-tuned within the application (option #1 above).

2.7. External data and user provisioning (CODICT, LDAP)

In a situation where EP has some master data or translations already available (e.g. some information about persons/groups within EP can be found from CODICT and LDAP) the Contractor has to use this data and not to introduce redundant data or processes to maintain this data. Exact details about integrations to the master data applications have to be defined in the DTA. The integrity of reference data between EP systems is important, so existing reference data repositories should be used and EP IDs and codes that must be respected.

2.8. GUI and Ergonomics

Parliament has written a document [R3] with a complete set of ergonomics recommendations. The development of the application requested should follow, as a general rule, the principles outlined in it.

In particular, the obligatory rules are the ones considered as strong requirement, marked in the document with a flag. The rest of the rules would remain only as desirable.

During the specification phase the different devices (desktop computers, laptops, tablets, smartphones, etc.) that will be used to access the application shall be defined. The ergonomics characteristics will then have to be adapted according to the characteristics of these devices (screens, keyboards, browsers...).

All the applications that are available to the people outside EP network (=internet and DMZ) have to follow the guidelines of the DG COMM. All GUIs are subject to the pre-approval by DG COMM. In general these applications should follow the Europarl graphical guidelines [R6] and [R7].

2.9. Performance

EP is responsible for providing the adequate, reliable, robust and scalable hardware in order to run the applications according to the load scenarios. The Contractor is responsible for providing EP with recommendations regarding the tuning, configuration and setup of the EP environment to run the application efficiently.

While EP will be responsible for the running of SQL script for database creation, and afterwards of the database hosting, the Contractor will bear the responsibility of tuning the rest of the elements:

- The Java application
- The SQL statements embedded in the persistency layer (both structure and data population)

In other words, the Contractor will provide the software deliverables designed in the most optimized way that could guarantee a smooth, secure, scalable and robust execution in the

EP's environment. With that in mind the application must respect performance requirements relying mainly on the approach of measuring the response time of applications from the perspective of the end user.

The target performance criteria and thresholds will be defined in the document "Performance Test Request".

2.10. Multilingualism

The application must be multilingual in the sense of showing the GUI with all the labels in different languages. English language is established as the default language at every module's initiation.

The users are allowed to switch the user interface language dynamically from each screen, maintaining all the existing content but changing the labels of the different elements displayed. This feature allows the user to choose a new language choice at any time during their usage session without rebooting his equipment or restarting the application.

However, the data model has been defined as not multilingual. That means that all the information will be stored in English.

The different translations of the labels that are used to handle the multilingualism must be stored in a way that they are modifiable without any code compilation, i.e. using database or properties file. Hardcoding of label translations is prohibited.

At EP it is mandatory to use UTF-8 encoding for any text, including data, source code, configuration files and any other produced material. The encoding of text files must be Unix-compliant.

Normally EP will provide the Contractor with the GUI labels and field translations required. The Contractor is responsible for implementing a means to introduce these translations – and future changes, if any – efficiently to the system.

2.11. NewTest

NewTest probes are used by EP to automatically verify that applications are up and running. Any applications developed for the European Parliament has to offer at least:

- a dedicated method to assess that the application is running ("isAlive" method)
- a dedicated method to assess the database is responding (simple select query)

These methods and the way to call them must be clearly defined in the NewTest monitoring document provided with the delivery of any application to be deployed in pre-production or production environment.

2.12. Deliverables

All documents and deliverables needed to deploy the application properly are required with all the deliveries provided by the Contractor. These contain at least:

- The application (sources and binaries of the application/module/component),

- SQL scripts (format to be defined by EP), in the event that it affects the database, for both database structure and data population,
- Required documentation (see Chapter 2.5), and
- Configuration files properly set depending on the environment

Concerning SQL scripts, the following rules must be followed:

- All SQL commands should be written in one line, even if large. Line change is considered a way to define different command
- Large SQL commands should be split into small pieces
- Scripts must follow the following approach
    - Put them in a dedicated order and name them with a number as prefix
    - The first one, name 00_install_projectname.sql, is the main script that launch the other ones
    - Before launching scripts to create a DB object (table, sequences, etc...), first launch scripts to drop these objects, so that the scripts can be executed to reset the whole schema
- With any new delivery, the provider must provide EP not only with SQL scripts to update the existing schema to the new version, but also consolidated scripts in order to be able to fully generate the new version from scratch

Concerning the application archive (.war) deployed on a Tomcat server, the following instructions must be respected

- Libraries for the database drivers (ie. oracle driver) are setup directly on the server by European Parliament teams. They then must not be embedded in the war file.
- The META-INF/MANIFEST.MF files must contain the correct version of the application, according to the pom.xml file.
- For an application named "myApp", datasource must be named "jdbc/myAppDB" and be placed inside the "apps-repo" folder inside "conf\Catalina\localhost" of the Tomcat server

These documents will be checked before the deployment, which will be done by the Outsourcing cell of the CONCEPT Unit. In the event that the documents are not detailed or clear enough to perform the deployment, the Contractor may be requested to come in the EP's premises to perform this deployment. As a result of this mission, the documents must be updated in order to be sufficient for the next release.

3. SONAR METRICS AND THRESHOLDS

In this Chapter we define a set of metrics to be used when assessing the quality of code delivered by the Contractor. It also defines the thresholds expected within the PROJECTS department.

3.1. Required artefacts

Code documentation

Beside traditional documents, the supplier has to deliver a Maven application site including the following elements:

- Javadocs
- code cross reference (PMD reports)

These artefacts are essential to allow "manual inspection" of the code during the acceptance phase and to facilitate the system maintainability.

Unit tests

The system will be delivered with sets of unit tests linked to portions of code (identified during the contractual phase) that for their criticality need to be validated.

Unit tests are essential in the maintenance phase in order to assure that changes in the code didn't disrupted application reliability. They reflect the robustness of the software components.

Unit tests are useful as well to assert that the code is "strong" enough to resist to the "nominal stress". Using a metaphor they could be seen as non-destructive tests in manufacturing (like, for instance, X-ray welding inspection in ship building).

Unit tests should cover the corner cases as: entry value null, empty strings etc. "Too Simple to Break" methods can be skipped from Unit Testing.

Guidelines:

- Name your unit tests clearly and consistently (by default, the following naming should be used: subject, the scenario and the result**, for example createNewUserAccount_UserAlreadyRegistered_ShouldShowErrorMessage()**.
- Use "Orthogonal array testing" technique, where feasible
- Don’t make unnecessary assertions
- Test only one code unit at a time
- Mock out all external services and states
- Avoid unnecessary preconditions (Avoid having common setup code that runs at the beginning of lots of unrelated tests. Otherwise, it’s unclear what assumptions each test relies on, and indicates that you’re not testing just a single unit.)

- Don’t unit-test configuration settings (your configuration settings aren’t part of any unit of code)

Reports

Sonar version 3.3.1 will be used to create reports displaying the quality elements of the delivered system and its trends from one version to the next one. Sonar relies on three configurable Maven plug-ins: Checkstyle, PMD and Findbugs.

3.2. Metrics

Methods Cyclomatic Complexity

The Cyclomatic Complexity Number is also known as McCabe Metric. It all comes down to simply counting 'if', 'for', 'while' statements etc. in a method. Whenever the control flow of a method splits, the Cyclomatic counter gets incremented by one.

Each method has a minimum value of 1 by default except for accessors which are not considered as methods and so they don't increment the complexity.

According to Sonar, 1-4 is considered good, 5-7 ok, 8-10 consider re-factoring, and 11+ re-factor now!

Coverage

Coverage metric is a mix of the line coverage and branch coverage metrics to get an accurate answer to the question "how much of a source-code is being executed by your unit tests?

On a given line of code, line coverage answers the question: "Is this line of code executed during unit test execution?"

On each line of code containing some Boolean expressions, the branch coverage simply answers the question: "Has each Boolean expression evaluated both to true and false?

The categories to be taken into an account:

- Function coverage
- Statement coverage
- Decision coverage
- Condition coverage
- Loop coverage
- Parameter Value coverage

Rules compliance (violations)

Total of violations of the following rules: efficiency, maintainability, portability, reliability, usability, clarity, flexibility, modularity. These violations are categorized in five groups: blocker, critical, major, minor and info.

Duplications (density of duplicated data)

The ratio between the number of lines touched by duplication and the number of physical lines (expressed as percentage). It is crucial to have any block functionality present only once in the code for easy maintenance. Wherever redundant code occurs it should be separated into a specific piece of code in a method in the same class or in a utility class.

3.3. Thresholds within PROJECTS department

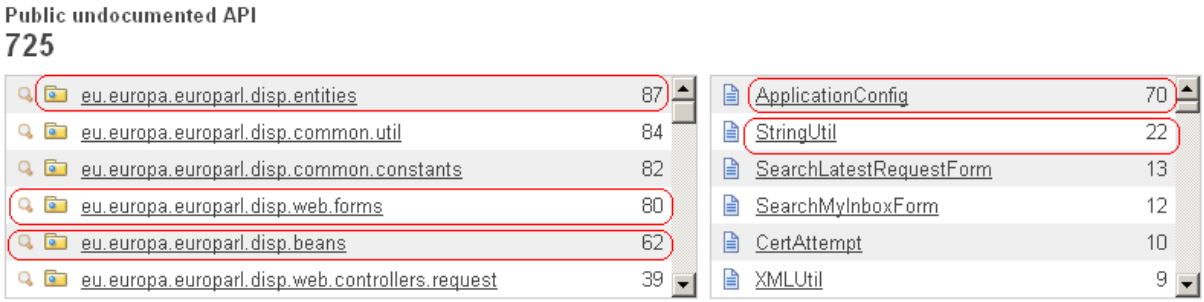
The following thresholds are expected for the PROJECTS department.

"Sonar way with Findbugs" will be used. In this context:

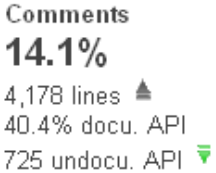
Metric	Threshold
Methods Cyclomatic Complexity	<=9 (set in the rules)
Class Complexity	<=10 (Sonar output, see example) (exceptions to be listed with a proper justification)
Coverage	>=70%
Unit test success	100%
Rules compliance	0 blocking violations, 0 critical violations, >= 90% General Rules Compliance
Duplications	<= 4% (exceptions to be listed with a proper justification)
Comments	>=20%
Undocumented APIs, which are not in the 'exclusion list'	<=25% (exceptions to be listed with a proper justification)

Exclusion list: includes all the classes that are simple or straight forward to worth being commented (ex: entities, beans, forms, constants, configuration files)

These are the numbers we add up to the 'exclusion list':



Undocumented API is calculated as such: Reported *undoc API* minus *'exclusion list'* divided by *total API*.

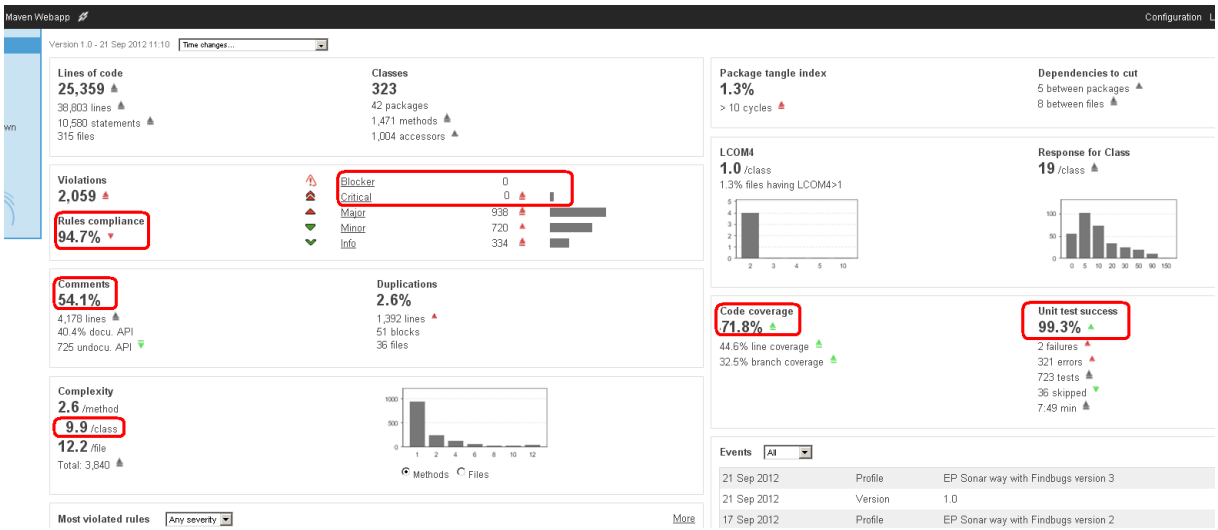


An example: (725 - 321)/(725 *1/59.6%)=33%

Exceptions:

Defined threshold could be overpass in special cases with proper justification. The classes that will be excepted from the rules have to be listed with a clear justification why they don't follow the rule. The aim is to have a code that can be easily maintained after the project is delivered.

An example:



4. ACCEPTANCE PLAN

This section outlines the strategy for managing the technical deliverables as part of the acceptance process. It provides a high-level overview of the approach to verify that all the elements (software pieces and documents) have been delivered according to all the requirements stated and provided to the Contractor.

Upon the reception of the deliverable package, that generally will comprise software artefacts and documentation, EP will confirm the receipt after verifying that all the elements expected are included in the delivery package, regardless of its content.

Then a Delivery Testing sub-process starts with the Parliament's project team liable for it. This process contains the following tasks:

- documentation review
- source code quality testing
- deployment and integration testing
- security testing

- ergonomics testing
- preliminary functional testing
- performance testing
- user acceptance testing

The next Chapters give more details about each of these activities.

4.1. Documentation review

Upon delivery of a document, the European Parliament will enter the acceptance process for the documents. Documents shall be checked for both content and structure.

4.2. Source code quality testing

In order to verify that source code is compliant with the EP requirements, a code review will be performed by the EP's project team once the package has been imported into the SVN repository.

This process will consist of two main tasks:

1. An automated test tool Sonar (see Chapter 3 details) will be used in order to check that the code is compliant with the European Parliament recommendations. The sonar configuration files are available upon request. Chapter 3.3 gives the threshold values that will be checked to accept the source code.

   Any exceptions to the thresholds need to be justified by the Contractor and approved in a written form by EP before the delivery.

2. A complementary review will be carried out by the project team in order to ensure that the rules and conventions provided in the European Parliament's recommendation documents, which will not be covered by the automated test tool, are fully respected. This review may contain a manual inspection of some of the code selected randomly.

During the source code quality testing EP will also check that the javacode will comply with the other rules stated in Chapters 3.

4.3. Deployment and integration testing

The application will be deployed in the European Parliament premises either by the Outsourcing Cell following the procedures explained in the installation and setup guide or by the Contractor during a dedicated trip with the help of Outsourcing Cell.

A first test consists in verifying that the application can be launched, is up and running without any incidents, and that it is properly integrated to the other applications. This test round, commonly called as Smoke Tests, must be successfully completed before starting the other tests, so their non-completion is blocking the acceptance process of the delivery. This first test phase will be repeated for any deployment done in EP premises, in any environment (development, integration, pre-production, production).

4.4. Security testing

During the process of security testing, also under the charge of the Parliament's project team, the following topics, among others, will be covered to measure the respect of the security policy laid down as security requirements:

- Authentication.
- Authorisation to allow or restrict access to the different resources will be handled by roles and the assignments of roles to users. This information will be stored in specific entities in the application's database or provided by the authentication.

  Considering that two main resources will be treated (business modules and data), the following three types of authorisation have to be managed and tested:

  - Modules: each user will bear access only to certain modules of the application.
  - Functionality: inside each module, some restrictions regarding the business functions will be also applied.
  - Data entities: now we do not limit the access to the functionalities in themselves but the data that the functionalities will manage. Normally this authorisation aspect will be kept at the database level.
- Privacy. Although a specific user has access to a data entity that does not mean that he is entitled to access the entire entity. Only a subset of the entity and/or a subset of the attributes would be enabled to a user. This aspect is also known as fine-grained security. Therefore the two kind of data privacy aspects will be tested:
  - At record-level. The user's permissions will allow him to access to all the attributes of a subset of an entity, established according to the business rules mainly derived from the user's membership to a concrete group.
  - At the attribute-level. This feature would restrict the visibility to certain attributes within the subset of data the user is enabled. In other words, the content of the defined attributes will remain hidden to the users.

In addition, an Application Intrusion Test will be performed. The approach, based on [R8], is divided in two phases:

1. Phase 1 Passive mode: In the passive mode the tester tries to understand the application's logic and plays with the application. Tools can be used for information gathering. For example, an HTTP proxy can be used to observe all the HTTP requests and responses. At the end of this phase, the tester should understand all the access points (gates) of the application (e.g., HTTP headers, parameters, and cookies). The "Information Gathering" Chapter of OWASP Testing Guide ([R8]) explains how to perform a passive mode test.

2. Phase 2 Active mode: In this phase the tester begins to test using a set of active tests, which have been split into 11 sub-categories[3]:
    - Information Gathering
    - Configuration and Deployment Management Testing
    - Identity Management Testing
    - Authentication Testing
    - Authorization Testing
    - Session Management Testing
    - Input Validation Testing
    - Error Handling
    - *Cryptography*
    - *Business Logic Testing*
    - *Client Side Testing*

4.5. Ergonomics testing

The ergonomic test will verify the compliance of the EP's ergonomic recommendations laid down e.g. in [R3] and the overall usability of the application.

4.6. Preliminary functional testing

Functional testing covers how well the system executes the functions it is supposed to do according to the functional and technical specifications provided to the Contractor. A preliminary functional testing will be carried out by the project team in order to verify that the functionality requested is included in the deliverables. Thus it will include user commands, data manipulation, searches, business processes, user screens (GUI), and also integrations with the other systems.

4.7. Performance testing

The performance tests will be executed by EP in the pre-production environment (similar to production environment), where previously the new system has been deployed. Three main sorts of tests will be run:

- Load testing which measures the system's ability to handle varied workloads. It will be conducted to understand the behaviour of the application under a specific expected load.

[3] source OWASP: https://www.owasp.org/index.php/Testing:_Introduction_and_objectives

- Stress testing which looks for errors produced by low resources or competition for resources. This testing is normally used to break the application.
- Volume testing, also known as scalability testing, which subjects the software to larger and larger amounts of data to determine its point of failure.

The Contractor will provide EP with different scenarios, depending on the business needs and the features of the application, using the PerformanceTestRequest document provided with the delivery.

The supplier is responsible of providing SQL scripts to populate the database with test data that will be used during the performance tests, and SQL scripts to clean the database after these tests. These scripts must allow EP to launch the tests as many times as they need.

4.8. User Acceptance testing

User Acceptance Testing, which is a black box testing, will give the internal client (the business users) the opportunity to verify the system functionality and usability prior to the system being moved to production. The User Acceptance Test will be the responsibility of the internal client; however, it will be conducted with full support from the EP's project team and the Contractor. The project team and the Contractor will help the client to develop the acceptance criteria as well.

The formal acceptance of any new work package as a part of a system can only be made once EP has validated the functionality expected and described in the set of functional and technical requirements.

5. TRANSITION

At the end of the contract the Contractor may be asked to perform the transition of all necessary knowledge and documentation enabling the European Parliament to get the similar kind of services from the new Contractor or to be done by itself. Additional handover details may be defined in CSOW or in other appendixes.

In addition to the application(s) and documentation already provided with the different deliveries of the project, properly updated with the potential changes that may have occurred during the rollout period, the following documentation shall be included in the transition documentation:

- Transition document list, describing all the components provided.
- Warranty period reports, including all the issues reported, their situation (open or closed) and the ongoing activities.
- Open issues log.

ANNEX I.11

ITS14 - SUBTASK FORM

To be used for Quoted Time and Means Orders during contract execution

SUBTASK FORM	**Request Date:**	Contract signing date
	SUBTASK no:	1
	Specific Contract reference:	ITS14-L3-R1/PE-ITEC-DES-CONCEPT-PROJECTS/CS021
Short Description:		Trilogue Table Functional Analysis and Solution Architecture, Oct 2016
Institution:		European Parliament
Responsible for the realisation of the SUBTASK:		Olivier Leboeuf

Description of work:

- Provide EP with near-site Brussels, PROJECTS Functional Analysis and Solution Architecture services during 01/10/2016 - 31/10/2016. The work is done for:
    1. the Trilogue Table Editor project
    2. the e-Parliament program (managed by one program manager official)
- The services:
    1. Functional Analysis services
        - Gathering the requirements, writing specifications, following-up the development activities from the functionality point of view, functionality testing, usability review and designs etc.
        - Service availability required for this service: "*Both AM and PM*"
    2. Solution Architecture services
        - Translating business-level requirements to system-level requirements, application architecture and end-to-end solution design
        - Service availability required for this service: "*Both AM and PM*"
- Other work to be done:
    1. Work progress reporting (in EP's Jira tool)
    2. Meetings, see CSOW for additional details.
    3. Service management and the related deliverables (during 01/10/2016 - 15/11/2016, see below)

Expected deliverables and dates of delivery (unless otherwise specified, the dead-line for all deliverables is 31/10/2016)**:**

1. Functional Analysis related deliverables:
    - Draft and partial Requirements document (functional and non-functional)
    - Draft and partial Functional specifications
    - Draft and partial Solution and Functional Architecture document
2. Solution Architecture services related deliverables
    - Draft and partial Solution and Operational Architecture document
    - Draft and partial DTA (Technical Architecture Document)
    - Source code and related documentation (stubs/mock-ups, proof of concept, prototypes, ...)
3. Service Management related deliverables:
    - Service Enhancement Plan (only if the SLA Report provided by the EP contains at least one service where EP's overall satisfaction is not "Good" - dead-line 15/11/2016
    - Delivery Note - dead-line 15/11/2016

SUBTASK FORM	**Request Date:**	Contract signing date
	SUBTASK Number:	1
	Specific Contract reference:	ITS14-L3-R1/PE-ITEC-DES-CONCEPT-PROJECTS/CS021

WORKLOAD ESTIMATES					
PROFILE	**PRICE/HOUR**	**ON-SITE/ OFF-SITE/ NEAR SITE**	**PLACE OF WORK**	**NUMBER OF HOURS**	**TOTAL PRICE**
PANL-SL3		Near-site	Brussels		
PDVS-SL3		Near-site	Brussels		
Total				h	€

Planned starting date of work:	
Planned delivery date of work:	

DEFINITION OF WORK PACKAGE AND WORKLOAD ESTIMATES		
	CONTRACTOR	PARLIAMENT
RESPONSIBLE PERSON:		Olivier Leboeuf
Starting date of work:		
DATE AND SIGNATURE FOR AGREEMENT OF WORKLOAD DEFINITION AND ESTIMATES, ALLOWING WORK TO START:		

SUBTASK FORM	**Request Date:**	Contract signing date
	SUBTASK Number:	1
	Specific Contract reference:	ITS14-L3-R1/PE-ITEC-DES-CONCEPT-PROJECTS/CS021

PROVISIONAL ACCEPTANCE OF WORK		
	CONTRACTOR	PARLIAMENT
PERSON RESPONSIBLE:		Olivier Leboeuf
DATE OF DELIVERY:		
DATE AND SIGNATURE DENOTING DELIVERY AND RECEPTION OF FORESEEN DELIVERABLES (PROVISIONAL ACCEPTANCE):		

FINAL ACCEPTANCE OF WORK	
	EUROPEAN PARLIAMENT
PERSON RESPONSIBLE:	Olivier Leboeuf
DATE OF ACCEPTANCE:	
DATE AND SIGNATURE DENOTING FINAL ACCEPTANCE OF THE DELIVERIES:	

Reminder: WARRANTY (Article I.10 of the Framework Contract):

1. *For each product which is the subject of the services, the Contractor shall grant a warranty period of **six months** from the date on which **the final acceptance** report referred to in **Article I.9.2** is drawn up.*
2. *The Contractor shall be required to remedy errors or malfunctions of which he is notified by the European Parliament during the warranty period within **30 calendar days** of such notification.*