

Annex D

to Tender Specifications

**Work procedures for
Requests for Services
and Project Delivery**

Workflow (indicative) for Request for Services

The table below outlines an indicative workflow that is to be followed for Service requests concerning new software delivery or existing software upgrades. Each specific Request for Services (RFS) launched under the FWC will provide workflow information that will be binding for the specific RFS.

Activity		Actor	Task completion time (est. Calendar Days)		Notes/Typical Deliverables (for specific contracts)
			Time & Means (TM)	Fixed Deliverable & Timing (FDT)	
1	Request for proposals	EMSA	2-10	15	Specification of the tasks to be executed including user requirements, type of deliverables and/or time and means required.
2	Provision of offer	contractor	2-5	15	The offer for specific contracts will specify: <ul style="list-style-type: none"> • The total price of the offer • The detailed prices per activity/person day offered under the specific contract (only for the TM contracts). • The description of the responsibilities of each team member and their CVs. • the specific methodology for the performance of the tasks requested (only for FDT).
3	Go/No-Go Evaluation	EMSA	1-5	10	
4	Provision of analysis / design document	contractor	2-10	20	Use cases / Functional specification / Software design description / System dimensioning document.
5	Go/no-go Decision for implementation or request for revision of design	EMSA	1-5	1-5	N/A
6	Implementation	contractor	1-20	15-TBD (depending on the nature/ size of the RFS)	Depending on the nature of contracted activity, implementation could be separate in several stages. One of these stages could be prototyping.
7	FAT ¹	contractor	1-5	3-10	FAT report/ Software Release Note
8	Corrections- FAT	contractor	1-5	3-5	
9	Preview- Pre-SAT	EMSA or third party	1-3	3-5	All the artifacts shall be reported in TeamForge
10	Corrections- Pre-SAT	contractor	1-3	2-5	Analysis results are recorded in

¹ Refer to the next chapter for detailed requirements related to the FAT

Activity		Actor	Task completion time (est. Calendar Days)		Notes/Typical Deliverables (for specific contracts)
			Time & Means (TM)	Fixed Deliverable & Timing (FDT)	
	and software delivery				TeamForge. Software (executables and source code) delivered in TeamForge. Release notes and updated ICM is attached.
11	SAT – Functional tests	EMSA and/or third party	1-5	3-15	SAT includes regression testing All the artifacts shall be reported in TeamForge
12	SAT – performance tests	EMSA and/or third party	1-3	1-3	SAT may include load / stress/ security tests. All the artifacts shall be reported in TeamForge
13	Development Contractor Analysis	contractor	1	1-2	Analysis results are recorded in TeamForge
14	Corrections SAT	contractor	1-3	2-5	Software (executables and source code) delivered in TeamForge. Release notes and updated ICM is attached.
15	SAT 2 nd Run	EMSA and/or third party	1-5	3-10	SAT may include regression/ load/ stress and security testing All the artifacts shall be reported in TeamForge
16	Development Contractor Analysis	contractor	1	1-2	Analysis results are recorded in TeamForge
17	Corrections SAT	contractor	1-3	2-5	Software (executables and source code) delivered in TeamForge. Release notes and updated ICM is attached.
18	SAT 3 rd Run	EMSA and/or third party	1-5	3-5	SAT may include regression/ load/ stress and security testing All the artifacts shall be reported in TeamForge
19	Development Contractor Analysis	contractor	1	1	Analysis results are recorded in TeamForge
20	Corrections SAT	contractor	1-3	2	Corrections will include at minimum the blocking artifacts. Software (executables and source code) delivered in TeamForge. Release notes and updated ICM is attached.
21	Stabilisation period as per EMSA Release and deployment procedures	EMSA	14	14	
22	Go/no-go Decision for launching in production environment	EMSA	1-2	0-3	Release notes/ User manual updates

Conditions of execution

Contract phases and deliverables

Kick-off meeting

The meeting should take usually within 1 calendar day from contract signature (at maximum, one week after the contract's signature). It could be executed, if requested by EMSA, via a teleconference.

Before the kick-off meeting the contractor will deliver a detailed execution plan for the contract and a draft of the Software Requirements Specification document (SRS)

During the kick-off shall be agreed the final project implementation plan for the contract and also agreed potential changes on the SRS.

Design

The purpose of this phase is to design the software, including functionalities, business rules, data model, user interface, system interfaces, architecture, and database model.

The design documentation to be delivered by the contractor will include the documentation that shall be foreseen in the RFS. If no specific reference is to be made in the RFS, in line with the FWC tender requirements, it is anticipated that the contractor will deliver, at minimum, the documents mentioned in the table below.

Table 1 Design documentation

Document	Notes
SRS (Functional Software requirements Specification).	Templates should be agreed with EMSA. In the absent of specific information in the RFS the content specification as per the FWC tender specifications article 4.2 shall apply.
SDS (System Design Specification)	
STP (System Test Plan)	The STP should be segregated to several docs/ booklets to distinguish very clearly the test plans concerning each distinct software component or service under testing. The document should include all the required information as per the "test cases specification" document mentioned in AnnexFAppendix C_Service_Validation_Verification_Testing_Procedure.
UM (User Manual)	Template should be agreed with EMSA. In the absent of specific information in the RFS the content specification as per the FWC tender specifications article 4.2 shall apply.

Drafts of design documentation shall be delivered for review at a date to be specified in the RFS (usually at least 1 week before the delivery date of the design phase).

EMSA will review the design documentation delivered by the contractor. It will provide the contractor with its comments and/or reservations within a time period to be specified in the RFS (maximum two weeks of the date of delivery). The contractor will be required to revise the design documentation to address EMSA's comments and/or reservations. The revised design documentation shall be submitted to EMSA within a timeframe established by EMSA.

The design phase will be considered concluded when the contractor and EMSA reach an agreement on the design documentation and finalised versions have been delivered to the Agency.

Development and tests

The purpose of this phase is to develop the software components that are to be contracted according to industry best practices, the design documentation as well as undertaking the necessary testing and correction to ensure that the deliverables meet the requirements and are in line with the design documentation.

The deliverables shall include as a minimum:

- a. Software source code,
- b. Software binary,
- c. On request from EMSA: Virtual machine containing the software,
- d. Factory Acceptance Test (FAT) reports and any updates of the Software Test Plan (STP),
- e. Updated version of the Installation and Configuration Manual (ICM) including installation sequence, configurations, etc.,
- f. Applicable scripts:
 - Database scripts,
 - Configuration and deployment scripts to perform the weblogic server installation. These should use WLST and properties files that can be edited by EMSA depending on the installation environment. JDBC data source configurations should be delivered in a separate script,
 - Scripts for data migration, if applicable for the RFS,
- g. Update of the User Interface Manuals, if applicable for the RFS.
- h. The “Master” test plan for testing the specific release.

Contractors should note that all the documents utilised for configuration , release and deployment (refer to the list below) should conform with the EMSA Release and deployment procedures and indicative content templates.

Table 2 Release and Deployment documentation

Document	Notes
STP (System Test Plan) update), if changed during the software implementation phase	The STP should be segregated to several docs/ booklets to distinguish very clearly the test plans concerning each distinct software component or service under testing. The document should include all the required information as per the “test cases specification” document mentioned in AnnexFAppendix C_Service_Validation_Verification_Testing_Procedure.
ICM (Installation & Configuration Manual);	Templates should be agreed with EMSA. In the absent of specific information in the RFS the content specification as per the FWC tender specifications article 4.2 shall apply.
Release Master Test Plan	The document should include all the required information for the “Test plan” mentioned in AnnexFAppendix C_Service_Validation_Verification_Testing_Procedure

A draft of documents (d) shall be delivered for review at least 3 weeks before the delivery date of the phase. The final version of the Software Test Plan (STP) and the FAT report must be delivered with the first software delivery.

A draft of the ICM shall be submitted to EMSA at a date to be specified in the RFS (usually 3 weeks before the planned date for the first deployment in test environment accompanied with a final version which normally should be submitted 1 week before that deployment).

The delivery must be driven by release and not by contracts. This means that all the functionalities and bug corrections that go to production in the same release must be delivered at the same time and independently of the

contracts that they are bound for. Instructions in this respect will be included in each RFS concerning software deliveries concerning a specific release.

The first software delivery, used for executing the first tests must include all the functionalities and bug corrections that go to production on the planned release.

Delivery is considered concluded when a successful installation of the software has been executed on EMSA's acceptance environments using the software source code delivered by the contractor.

Schedule

The project schedule and detailed is to be provided by the contractor in the offer and agreed with EMSA at the kick-off meeting. The schedule in the offer must at least meet the milestone dates and the workflow for implementation, testing at contractor site, site acceptance testing that is going to be specified in an RFS.

Test by the Contractor

The following specific requirements are applicable:

1. Before the contractor formally delivers software to EMSA for the acceptance procedure, it shall ensure that all tests required by the development cycle have been successfully completed.

For this purpose, the Contractor:

- a) Should conduct internally a Test Readiness Review² (TRR);
 - b) Shall conduct a Factory Acceptance Test (FAT)³;
 - c) Shall provide EMSA with access to a software “preview” site to track that all the changes made in the SSN web interface meet the agreed specifications.
2. EMSA staff could be present at the FAT to obtain evidence of the successful completion of the activity. Only after EMSA has accepted the results of the FAT (based on the FAT report) is the contractor allowed to deliver the software for pre-SAT and SAT.
 3. The FAT shall be executed in accordance with the Software Test Plan agreed with EMSA. In this respect the STP should be delivered to EMSA at the planned finish of the design phase (as stated above) and an update be delivered at a date to be specified in the RFS before the delivery of the software.
 4. During the FAT, the contractor shall perform all the installation steps as detailed in the ICM for the release(s) being delivered.
 5. The FAT report shall:
 - a) Describe, and justify the suitability of, the characteristics and scale of the FAT environment.
 - b) Describe all the issues found and reported by EMSA during the preview of the software and indicate if they have been corrected.

² Test Readiness Review I (TRR I) is a formal review, conducted by the Program Manager (PM) appointed by the contractor, signifying the Component Validation and Integration portion of the system or system component under development is complete and recommends that the system/component shall move into the Factory Acceptance Testing. The results of the TRR will demonstrate that each individual component and the system where the components belong are developed or configured in accordance with the approved design and function properly to meet specified requirements.

³ The main objective of the FAT is to confirm that the software implemented meet the agreed design specification (functional/ non-functional) and contract requirements, so it could be delivered for installation at EMSA.

- c) Include proofs that full regression tests of SSN components affected by the delivery have been conducted.
 - d) Describe all the aspects of the delivery that are major or blocking (refer to the definitions in the section “Acceptance procedure”).
6. With the FAT reports, the contractor will provide **all the tests scripts used to automate test cases along with instructions enabling EMSA to re-use the scripts.**

Scope of tests by the contractor:

During the FAT, the system should be sufficiently tested (proper implementation of business rules / functional requirements, performance, security of transactions, load, etc.) before being delivered to EMSA for the acceptance tests.

The goals related to testing of system functions are:

1. Conformance with business rules/functional requirements,
2. Completeness,
3. Correctness,
4. Avoidance of regression errors (impacts to functions of the application that should not be affected by the contracted work).

The non-functional goals of the overall testing procedure are the average response time of the system to a request for information and the security of transactions.

The STP should make clear references to the test cases/scenarios that will be executed during the FAT. In this respect the following table provides the minimum requirements with respect to the test goals mentioned above.

Table 3 Minimum requirements regarding the Factory Acceptance Test by the Contractor

Quality Requirement	Quality Criterion	Metric Threshold	Threshold
Completeness	Coverage of requirements	Percentage of functional requirements listed in the RFS covered by the STP (at minimum one test case, as well as additional test cases if necessary for fully testing the applicability of the requirement)	100%
Completeness	Coverage of business rules	Percentage of business rules, as defined during the design phase, covered by the STP (at minimum one test case, or more)	100%
Completeness	Test Coverage for database tier	Percentage of statements covered in unit or integration test for database tier	>65%
Completeness	Test Coverage for business logic presentation tier	Percentage of statements covered in unit or integration test for business logic and presentation tier	>65%
Correctness	Blocking Issues/FAT	Blocking issues identified in FAT cycle 1	Less than 3
		Blocking issues identified in FAT cycle 2	No blocking issues

Quality Requirement	Quality Criterion	Metric Threshold	Threshold
Correctness	FAT cycles	Number of attempts to pass FAT criteria	No regression impacts
Performance	Response Time	Average response time	As many as required to eliminate all blocking issues (Min 2 attempts)

Referring to the thresholds related to the first four rows of the table above, the percentages mentioned in the “Threshold” column represent the amount of statements covered for each distinct module of the SSN system during the Unit and Integration tests. The Unit or Integration tests to be conducted (of functional or non-functional nature) shall be included in the test plan. The software approval work-flow at contractor site could envision the following three steps (steps a and c below are mandatory, step b optional):

- a) Unit and integration tests during software development,
- b) Unit and integration tests during the Test Readiness Review (TRR),
- c) Unit and integration tests during the Factory acceptance test (FAT).

The values in the table above show the rate of completeness of tests before the start of the FAT. The amount of tests and their nature has to be approved by EMSA and will be described within the test plan document.

Test environment for software pre-view

For the purpose of the software preview as introduced in the previous chapter, the contractor shall provide EMSA with access to a test environment set up and maintained by the contractor which includes:

- All the software components of the SSN upgraded or altered during the course of this contract; as well as
- Other components and system interfaces that shall be integrated and used in the same production release as the software delivered.

The system configuration will allow testing of the SSN System Interface using the STP. Furthermore the system will emulate realistically “external” systems interacting with the software components implemented and/ or upgraded by the RFS.

This test environment should be normally maintained (except if otherwise is specified in the RFS) in operation between the date scheduled for the start of FATs until the completion date of the contract linked to the RFS. Important to note in this respect is that the contract duration will normally include 12 months for bug corrections under warranty. The system will be initially used for executing the FATs and subsequently for testing patches and hotfixes that are to be delivered against bug reports of EMSA during the pre-SAT, the SATs and the 12-month warranty period of the software.

The test environment and the way of “simulating” external systems should be described in the offer for an RFS in broad lines. The specifications and configuration should be further detailed during the design phase of the system.

Acceptance procedure

For each delivery, EMSA will provide a formal indication of the acceptance, conditional acceptance or rejection of the delivery to the contractor.

The acceptance procedure will start when the software is available and running in EMSA's test & quality environments.

EMSA will verify that:

- All issues detected in any previous acceptance procedures have been corrected,
- The software conforms with the requirements and with the design specifications. Furthermore in case of software deliveries based on Java, EMSA will verify the source code quality using SonarQube. In this respect contractor should note the checking rules in the Appendix A of this Annex. All projects (those delivery new code base “from scratch” and those upgrading existing code) will be submitted to the rules identified for the “Quality Gate” in the Appendix. For existent projects an adoption plan may be established and agreed with EMSA. However, “**Blocking issues**” shall not be accepted.
- The existing components which are not impacted by this contract still conform to their specifications (no regression impacts),
- Implementation best practices have been followed,
- The binaries resulting from the software build in-house are correct and can be used for installing the application in EMSA environments (pre-production and production) and once installed achieve the desired results.

EMSA will classify any issues identified in three different categories reflecting their impact and severity:

1. Blocking issues:

Structural problems or serious issues (functional or technical) considered as limitations of the implementation with very high probability of interfering with the expected result. The contractor will be obliged to correct/execute all issues considered in this category,

2. Major issues:

Problems or issues that do not conform to the requirements or specifications or best practices or considered to be the wrong approach to obtain the result, but for each one of them a workaround or a correction is available. The contractor will be obliged to correct/execute all issues considered in this category,

3. Minor issues:

Changes considered to be a better solution but without a deep impact on the quality of the system. The correction/execution of the issues under this category will be decided on a case by case basis.

Each issue will be identified and described by EMSA and sent to the contractor. All issues will be registered in TeamForge. Appropriate access to TeamForge will be established for the contractor. The contractor is requested to track and monitor the treatment of each issue sent by EMSA. The acceptance tests and the classification of the issues will be determined in collaboration between EMSA and the contractor.

The outcome of the acceptance procedure is positive if no issue is found by EMSA. If issues are found by EMSA during the acceptance procedure, the contractor is requested to immediately correct them and the acceptance procedure restarts from the date of the delivery of the corrected deliverable.

EMSA can decide to conditionally accept the deliverable when some issues remain uncorrected and are not blocking. The condition for that acceptance is that a date for the correction of the remaining issues is defined by the contractor and agreed with EMSA. EMSA will take the decision to conditionally accept the product after evaluation of each remaining issue.

Appendix A to the Annex D of the Tender Specifications

Initial Quality Gate for Java Projects

SonarQube is used for quality checking. EMSA uses a special Java Quality Gate based on “Sonar way with Findbugs” quality profile.

Fail to pass this Quality Gate will imply the rejections of the version being delivered.

All projects (new ones and existent ones) will be submitted to the Quality Gate. For existent projects an adoption plan may be established and agreed with EMSA. However, “**Blocking issues**” shall not be accepted.

The following rules summarize the initial Quality Gate:

- **Blocking issues**

- Deliveries containing blocking issues shall not be accepted. The following issues are considered blocking:
 - Avoid Decimal Literals In Big Decimal Constructor
 - Avoid Print Stack Trace
 - Big Integer Instantiation
 - Broken Null Check
 - Class defines equal(Object); should it be equals(Object)?
 - Class defines hashCode(); should it be hashCode()?
 - Class defines toString(); should it be toString()?
 - Correctness - A known null value is checked to see if it is an instance of a type
 - Correctness - close() invoked on a value that is always null
 - Correctness - equals method always returns false
 - Correctness - equals method always returns true
 - Correctness - equals(...) used to compare incompatible arrays
 - Correctness - Impossible cast
 - Correctness - Impossible downcast
 - Correctness - Impossible downcast of toArray() result
 - Correctness - Null value is guaranteed to be dereferenced
 - Equals Hash Code
 - Integer Instantiation
 - Multithreaded correctness - Call to static Calendar
 - Multithreaded correctness - Call to static DateFormat
 - Performance - Maps and sets of URLs can be performance hogs
 - Performance - The equals and hashCode methods of URL are blocking
 - Preserve Stack Trace
 - Security - Hardcoded constant database password
 - String Instantiation
 - String To String
 - System Println
 - Bad practice - Method invokes System.exit(...)
 - Unused Private Field
 - Useless Operation On Immutable

- **Critical issues**

- Might be accepted but
 - have to be justified
 - A correction target date or version as to be defined

Depending on the situation, justification can be:

- Justified only for a period of time and correction target date/version defined or...

- Justified and permanently accepted.
 - Critical issues cannot increase from version to version
- **Major issue**
 - Major issues cannot increase from version to version
- **Cyclomatic complexity**
 - Max is set to 25
 - Cyclomatic complexity index over 25 might be accepted but:
 - have to be justified
 - refactoring have to be planned to a next version or date
 - Depending on the situation, the justification can be:
 - Justified only for a period of time and correction target date/version defined or...
 - Justified and permanently accepted.
 - Cyclomatic complexity index cannot increase from version to version:
- **Duplications**
 - Duplications shall be lower than 7.5%
 - Duplications cannot increase from version to version
 - Auto-generated classes/code can be ignored
- **Documentation and Comments**
 - Code comments should be greater than 20%
 - Public API Documentation should be greater than 50%
 - Both indicators cannot decrease from version to version
 - EMSA shall manually and randomly assess source code files to validate the quality and usefulness of the documentation and comments.
- **Unit Tests Coverage**
 - New starting projects shall have a minimum of 25% test coverage
 - For existent project, a minimum increase of 5% per major version is required.
 - Test coverage cannot decrease from version to version

Quality Gate shall be mandatory for all projects. As for any other project task, a Quality Gate will consume effort and time. Contractors are encouraged to adopt continuous and rigorous quality checking measures during the development process and submit each version to EMSA Quality Gate before delivery.