Heterogeneous Missions Accessibility – Testbed

Sensor Planning Systems Interface

HMA-T SPS I/F

SYSTEM REQUIREMENTS DOCUMENT

Code : HMAT-SRD-0001-DMS
Issue : 1.1
Date : 04/02/2009

	Name	Function	Signature
Prepared by	Reuben Wright (DMS) Ph. Mérigot (Spot Image)	Project Engineers	
Reviewed by	Ricardo Moyano (DMS)	Review team	
Approved by	Ricardo Moyano (DMS)	Project Manager	
Signatures and approvals on original			

This page intentionally left blank

Document Information

Contract Data	
Contract Number:	780-2/C09 "Heterogeneous Missions Accessibility - Testbed. Phase 2"
Contract Issuer:	SPACEBEL

Internal Distribution		
Name	Unit	Copies
Ricardo Moyano	DMS	1
Reuben Wright	DMS	1
Internal Confidentiality Level (DMS-COV-POL05)		
Unclassified ☐	Restricted ☑	Confidential ☐

External Distribution		
Name	Organisation	Copies
Yves Coene	SPACEBEL	1
Pier Giorgio Marchetti	ESA/ESRIN	1

Archiving	
Word Processor:	MS Word 2000
File Name:	HMAT-SRD-0001-DMS-11.doc

Document Status Log

Issue	Change description	Date	Approved
1.0	First version of the document	02/10/2008	
1.1	Following RIDs discussed at Technical meeting (26/11/08) YC-39: cover YC-40: section 2.1 YC-41: section 2.3 YC-42: section 2.3 YC-43: section 3.1 YC-44: section 3.2.1.1 YC-45: table 3, sections 5.2.2, 5.2.5 YC-46: various requirements in section 5 YC-47: sections 2.3, 5.7	04/02/2009	

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

1.1. Purpose

This is the System Requirements Document (SRD) for the HMA-T project relative to the implementation of the SPS Profile for Earth Observation interface ([OGC 07-018]) applicable to two different SPS systems.

The system requirements focus on the mandatory operations of the interface which are related to sending programming requests for EO products to support access to data from heterogeneous systems dealing with derived data products from satellite based measurements of the earth's surface and environment.

The document contains:

- Functional analysis of the interface, including approaches taken to solve specific problems identified during this analysis
- System requirements for the application

1.2. Scope

This document is produced as part of the Technical Specification that shall be subject to review at PM1. Therefore, it is applicable to the project from PM1 onwards.

1.3. Document Structure

This document is organised as follows:

- Section 1 contains this introduction.
- Section 2 contains the list of applicable and reference documents, as well as the applicable standards to the project.
- Section 3 provides a general description of the ECSIM.
- Section 4 describes the initial analysis made on the system prior to the specification of requirements.
- Section 5 presents the ECSIM system requirements.
- Section 6 contains the traceability matrices between the system requirements and the requirements baseline.

1.4. Acronyms and Abbreviations

The acronyms and abbreviations used in this document are listed below:

Acronym	Meaning
AD	Architectural Design

Acronym	Meaning
AR	Acceptance Review
CCN	Contract Change Notice
CDR	Critical Design Review
CFI	Customer Furnished Item
CITE	Compliance & Interoperability Testing & Evaluation Initiative
CM	Configuration Management
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CR	Change Record
CTL	Compliance Testing Language
CVS	Concurrent Versions System
DB	Data Base
DDF	Design Definition File
DJF	Design Justification File
DMS	DEIMOS Space
ECSS	European Cooperation on Space Standardization
EO	Earth Observation
EO-DAIL	EO Data Access Integration Layer
EOEP	Earth Observation Envelope Programme
ESA	European Space Agency
ESTEC	European Space Technology Centre
FP	Final Presentation
FTP	File Transfer Protocol
GMES	Global Monitoring for Environment and Security
HMA	Heterogeneous Missions Accessibility
HMA-I	HMA Interoperability
HMA-T	HMA Testbed
HMI	Human Machine Interface
HW	Hardware
ICD	Interface Control Document
IDE	Integrated Development Environment
IPR	Intellectual Property Rights
IRD	Interface Requirements Document
ITT	Invitation to Tender
KOM	Kick-Off Meeting
N/A	Not Applicable
NCR	Non Conformance Report
OGC	Open Geospatial Consortium Inc.

Acronym	Meaning
OO	Object Oriented
PDR	Preliminary Design Review
QA	Quality Assurance
QR	Qualification Review
R&D	Research and Development
RB	Requirements Baseline
RID	Review Item Discrepancy
SAR	Synthetic Aperture Radar
SDE	Software Development Environment
SDP	Software Development Plan
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SoW	Statement of Work
SPOT	SPOT Image
SPR	Software Problem Report
SR	Software Requirement(s)
SRD	Software Requirements Document
SVV	Software Verification and Validation
SW	Software
TBC	To Be Confirmed
TBD	To Be Defined
TS	Technical Specification
UML	Unified Modelling Language
UR	User Requirement(s)
URD	User Requirements Document
WBS	Work Breakdown Structure
WP	Work Package
WPD	Work Package Description
XML	Extended Markup Language
XSD	XML Schema Definition
V&V	Verification & Validation
WBS	Work Breakdown Structure
WPD	Work Package Description

2. RELATED DOCUMENTS

2.1. Applicable Documents

The following table specifies the applicable documents that shall be complied with during project development.

Table 1: Applicable documents

Reference	Code	Title	Issue
[OGC 07-018]	OGC 07-018	OpenGIS Sensor Planning Service Application Profile for EO Sensors	2.0 draft 04/02/09
[OGC 09-000]	OGC 09-000	OpenGIS Sensor Planning Service Specification	2.0 06/01/09
[SOW]	SPB-HMA-T-SOW-002	Statement of Work For HMA Testbed (HMA-T) Phase 2	1.0
[ECSS-40-1B]	ECSS-E-40 Part 1B	Software – Part 1: Principles and requirements	28/11/2003

2.2. Reference Documents

The following table specifies the reference documents that shall be taken into account during project development.

Table 2: Reference documents

Reference	Code	Title	Issue
[RD UML]	ISBN 0-201-57168-4	The Unified Modelling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson.	-
[RD PMP]	HMAT-SPMP-0002-SPB	HMA-T Phase 2 - Software Project Management Plan	1.2
[RD Notifications]	WS-BaseNotification	Web Services Base Notification OASIS Standard, 1 October 2006	1.3

3. GENERAL DESCRIPTION

3.1. Project Background and Perspectives

HMA (Heterogeneous Accessibility Mission) and other projects are part of the infrastructure deployed in the perspective of the GMES program. Within this the Sensor Web Enablement (SWE) initiative is focused on developing standards to enable the discovery, exchange, and processing of sensor observations, as well as the tasking of sensor systems.

The functionality that OCG has targeted within a sensor web includes:

- Discovery of sensor systems, observations, and observation processes that meet our immediate needs
- Determination of a sensor's capabilities and quality of measurements
- Access to sensor parameters that automatically allow software to process and geolocate observations
- Retrieval of real-time or time-series observations and coverages in standard encodings
- Tasking of sensors to acquire observations of interest
- Subscription to and publishing of alerts to be issued by sensors or sensor services based upon certain criteria

Within the SWE group, the enablement of such sensor web service is pursued through the establishment of several standard interface definitions. The services are the following:

1. **Observations & Measurements (O&M)** – The general models and XML encodings for observations and measurements made using sensors.
2. **Sensor Model Language (SensorML)** – standard models and XML Schema for describing the processes within sensor and observation processing systems; provides information needed for discovery, georeferencing, and processing of observations, as well as tasking sensors and simulations.
3. **Sensor Observation Service (SOS)** – An open interface for a service by which a client can obtain observations and sensor and platform descriptions from one or more sensors.
4. **Sensor Planning Service (SPS)** – An open interface for a service by which a client can
   - determine the feasibility of collecting data from one or more sensors or models
   - submit collection requests to these sensors and configurable processes.
5. **Sensor Alert Service (SAS)** – An open interface for a web service for publishing of and subscribing to deliverable alerts from sensor or simulation systems.
6. **Web Notification Service (WNS)** – An open interface for a service by which a client may conduct asynchronous dialogues (message interchanges) with one or more other services.

SPOT Image has taken a leading role to define a SPS EO Profile and participating actively at OGC level on both the specification and the interoperability program. It is an active member of the new Sensor Planning Service revision working group and the initiator of the new SWE common revision working group. These new revision working groups aim to take into account all the effort made during the definition of the SPS EO profile specification.

The version of this profile specification we shall implement is draft 2.0.

3.2. Function and Purpose

The main purpose of the HMA-T project is the implementation and testing of the interfaces to Sensor Planning Services dedicated to the EO Sensor domain complying with specification OGC 07-018.

The implementation of two different interface sensor planning systems shall be carried out:

- SPOT IMAGE SPS EO – optical
- **Earth Explorer Mission Software CFI**. This CFI represents a specific COTS to the project and it is a collection of software functions performing accurate computations of mission related parameters for Earth Explorer missions. In particular, one set of functions is dedicated to computing time segments at which an Earth Explorer satellite, or one of its instruments is in view of various targets, including zones (defined as polygons or circles, on the earth ellipsoid or at a given altitude).

Both implementations shall be based on a common SPS EO library, representing a low-level module that deals with sweCommon operations.

3.2.1.1. SPS Interface Implementation

The operations belonging to this interface are aimed at determining the feasibility of an intended sensor planning request, for submitting such a request, for inquiring about the status of such a request, for updating or cancelling such a request, and for requesting information about further OGC Web services that provide access to the data collected by the requested task.

Figure 3-1 shows the UML context diagram including all operations of the interfaces that DEIMOS and SPOT Image shall implement for their respective SPS. The external entity “Client” refers to any software component that can invoke an operation from the SPS server.

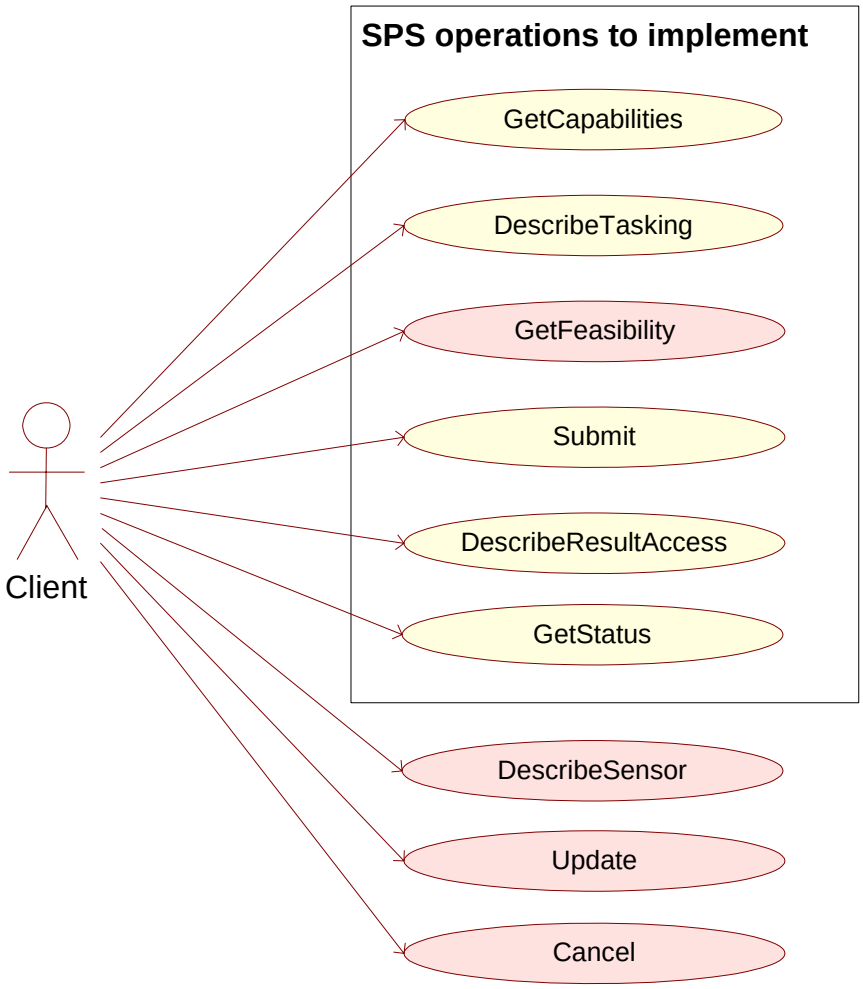

Figure 3-1: SPS interface use case diagram

The baseline work of the SPS EO application profile interface includes the mandatory operations of the [OGC 07-018] specification, plus GetFeasibility, that can be requested by a client and performed by both SPS servers. These operations are:

- `GetCapabilities`: This operation allows a client to request and receive service metadata (or Capabilities) documents that describe the abilities of the specific server implementation.
- `DescribeTasking`: This operation allows a client to request the information that is needed in order to send `GetFeasibility` and `Submit` requests. The response contains a description of the input and optionally the output parameters.
- `GetFeasibility`: This operation is to provide feedback to a client about the feasibility of a programming request.
- `Submit`: This operation submits the programming request.
- `DescribeResultAccess`: This operation allows a client to retrieve information how and where data that was produced by the sensor can be accessed. The server response may contain links to any

kind of data and not necessary through an OGC Web services nevertheless OGC Web services such as SOS, WMS, WFS or WCS are desirable.

- `GetStatus`: This operation is to provide information on a previously programming request.

All operations are synchronous and all except `GetCapabilities` are based on SOAP/WS-Addressing. `GetCapabilities` shall implement HTTP GET transfer using keyword-value pair (KVP) encoding.

3.2.1.2. SPS Interface Testing

The complete verification and activities shall be performed upon the interfaces development.

- **Unit tests** shall be designed, implemented and executed to test each individual software component composing the SPS and OP interfaces. Stubs shall be used in case of component interactions.
- **Integration tests** shall be designed, implemented and executed to mainly verify the interfaces between software components.
- Finally, **system tests** shall be designed, implemented and executed to validate the whole software.

It is to be noted, that normally system tests are also used for the acceptance, where a subset of tests is selected for the testing. System tests shall be carried out following the same approach as the unit and integration testing. Thus, test procedures shall be written in the same language as the SPS interface, and they shall test the involved operations by setting the input conditions, invoking the operation(s) subject to the test and verifying its response or outputs produced.

3.3. General Constraints

As mentioned above, the baseline includes only the implementation and testing of the mandatory (plus GetFeasibility) operations of the SPS specification. Therefore the following optional operations are not part of the scope of the work to be done:

- `DescribeSensor`
- `Cancel`
- `Update`

3.4. Environmental Considerations

For the **HMA-T SPS development** the following environment shall be used:

- Operating system: Linux.
- Development language: Java using the JNI library for the interaction with the Earth Explorer CFI (written in C).
- Essential libraries/software for the software development will be:
  - Apache Tomcat webserver
  - Axis2 SOAP framework

- XML libraries for reading/writing XML files

For **Configuration Management** the following software shall be used:

- ❑ CVS for CM tasks.

For **Project Management** the following software shall be used:

- ❑ Microsoft Project for planning and reporting of Project Management tasks.
- ❑ Microsoft Excel 2000 for reporting the software metrics to be included in Progress Reports

For **Documentation** the following software shall be used:

- ❑ Word 2000 to generate the documentation.
- ❑ Adobe Acrobat (v7) to generate the PDF format to deliver the documentation via electronic media.

4. FUNCTIONAL ANALYSIS

4.1. Introduction

This section presents the analysis made to create models that capture the essential requirements and characteristics of the SPS system. These models focus on *what* SPS components need to do, but leaves the details of *how* they will do it during the design phase.

4.2. Modelling Language

The Unified Modelling Language (UML) shall be used for describing the HMA-T model. UML is a visual modelling language (not a method) for software systems, which considers two different aspects:

- **Static structure** – this describes what types of objects are important for modelling the system and how they are related.
- **Dynamic structure** – this describes the lifecycles of these objects and how they collaborate together to deliver the required system functionality.

The HMA-T model shall be described using the following diagrams:

1. *Class diagrams* are used for describing the static view of the system. This document only identifies subsystems and the functionality allocated to them. Allocation of functions to classes is deferred to the design phase.
2. *Activity diagrams* are used for representing the dynamic behaviour of objects whose functionality is dominated by computations and data flows rather than states and events. Activity diagrams allow to model a process as a collection of activities and transition between those activities

The reader may find more detailed information on the Unified Modelling Language in [RD UML].

4.3. High-Level System Decomposition

4.3.1. HMA-T class diagram

Figure 4-1 shows the initial class diagram for HMA-T. There is a **Web Server** to handle web communications, a **SOAP Reader/Writer** and **SPS library** for extracting and creating messages, and an **SPS controller** to manage the interactions with an **internal SPS** and (if required) a **Notification Server**. The web communications are either handled directly over HTTP or as SOAP messages over HTTP. The two web communications components interface with any number of SPS Clients. The internal SPS system will be either for the SPOT optical sensor or the DEIMOS Radar sensor simulator.

The following diagram shows these components and their interactions. The boxed area is the SPS interface

The web server will be COTS, and the various components will be developed primarily in Java. Where there are defined C interfaces to an Internal SPS there will be JNI translators to provide a Java interface.

There will be two different SPS built – one for the SPOT sensor and one for the simulated Earth Explorer sensors. They will share a development approach and architecture, and the SPS Library.

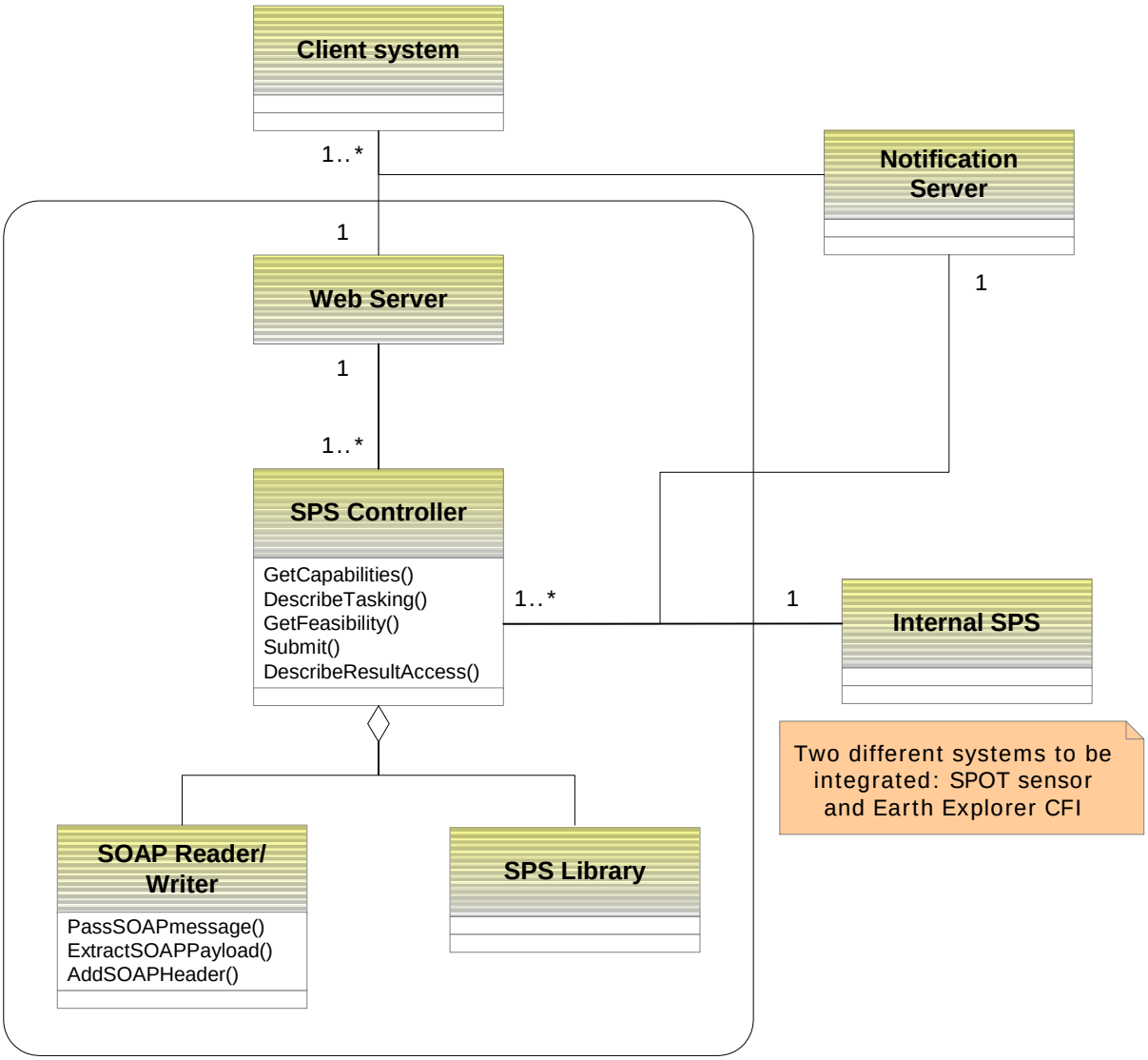

Figure 4-1: HMA-T class diagram

A more detailed description of each of the components identified above is presented in the table below.

Table 3: HMA-T initial class decomposition

Class	Description
Web server	This component shall be an Apache Tomcat web server, where the clients can connect to, and where the various components conforming the SPS implementation will be deployed.
SPS Controller	This is the core element that holds the logic of the system, in charge of the control and management of incoming requests from their reception up to the delivery of the appropriate responses. It is a web application deployed within the web server.
Notification server	This component, also deployed in a Apache Tomcat Web Server, is used to tell clients when the SPS has information for the client. The SPS controller

Class	Description
	can request the sending of a message to the clients. This will trigger the client to send a getStatus request to the SPS. It is only needed where processing of GetFeasibility or Submit messages is asynchronous.
SOAP Reader/Writer	This component is charged with the SOAP envelope extraction from incoming requests as well as appending the SOAP header to the built response prior to passing it to the web server for delivery to the client. It shall support version 1.1 of the SOAP specification.
SPS Library	The SPS library component is responsible for: ❑ Handling the extraction of data from SOAP message body elements, and write SOAP message bodies to pass back to the SOAP receiver component. ❑ Reading/writing sweCommon documents. ❑ Performing the deserialization/serialization of the SPS EO requests/responses. The data, once extracted, is available as Java classes, which are manipulated by the SPS_Controller component to manage the interaction with the SPS Systems.

4.4. Sequence Diagram

The following diagrams shows the messages exchanged by the architectural components in order to attend http/Get with KVP and http/Post with SOAP requests.

The Apache web server shall keep listening to requests coming from clients and shall route them to the SPS Controller component, which shall be responsible to attend to them. However, the first thing to deal with is the extraction of the SOAP header and serialise the payload contents, so that the data is managed through Java classes. These activities are carried out by the SOAP Reader Writer and SPS Library respectively. Afterwards, the SPS_Controller shall interact with the internal SPS exchanging information to satisfy the original request. As soon as this is done, the data is processed and translated into the XML messages as defined in [OGC 07-018], and finally, the appropriate identifiers and endpoint are appended to send the message back. As for the message reception, the serialization of Java classes into XML message is handled by the SPS Library and the second stage (adding header and passing to web server) by the SOAP Reader/Writer.

The examples below show:

a representative case for the basic HTTP GET exchange, where the XML response is sent immediately back to the client via the same HTTP connection.

a representative case for synchronous SOAP message exchange, where the response is sent immediately back to the client via the same HTTP connection.

a representative case for asynchronous message processing, where a response is sent immediately back to the client via the same HTTP connection but further activity generates a Notify message. This then triggers the client to send a getStatus to receive the outcome of the processing.

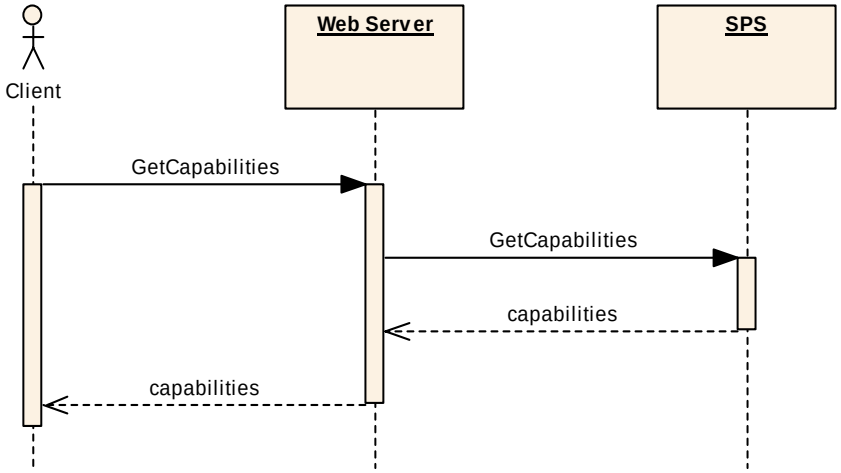

Figure 4-2: Sequence diagram for the http/GET GetCapabilities operation

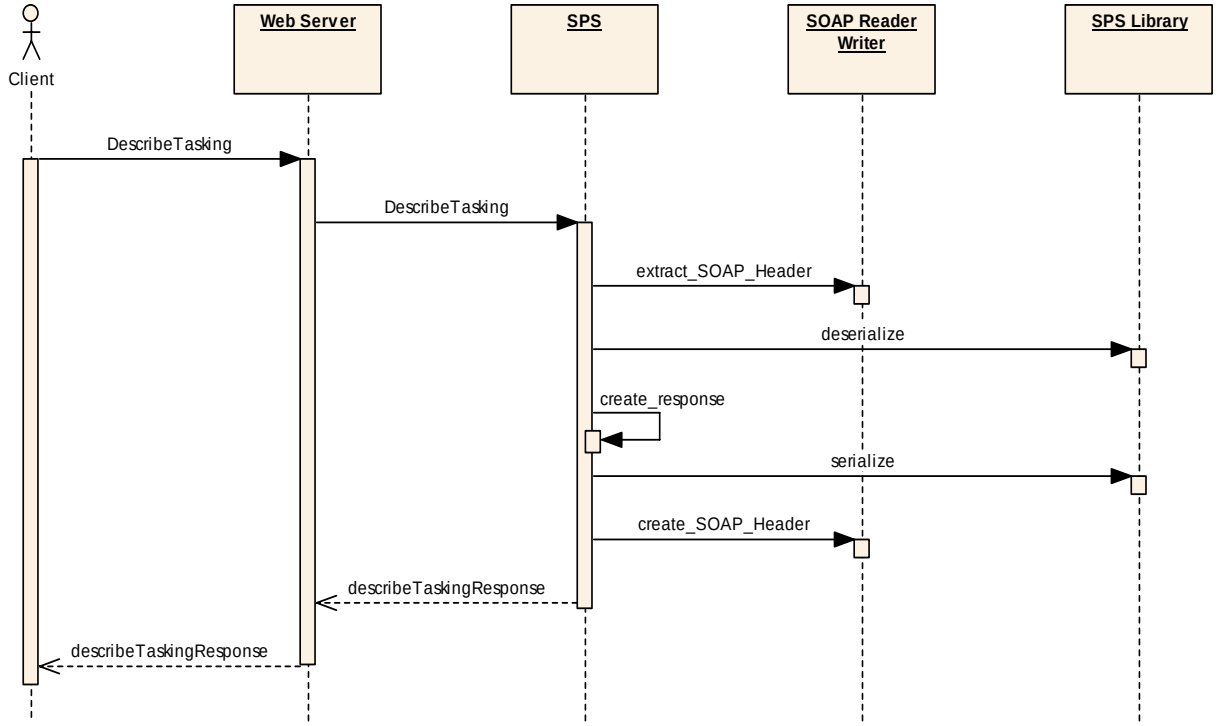

Figure 4-3: Sequence diagram for a HTTP POST operation

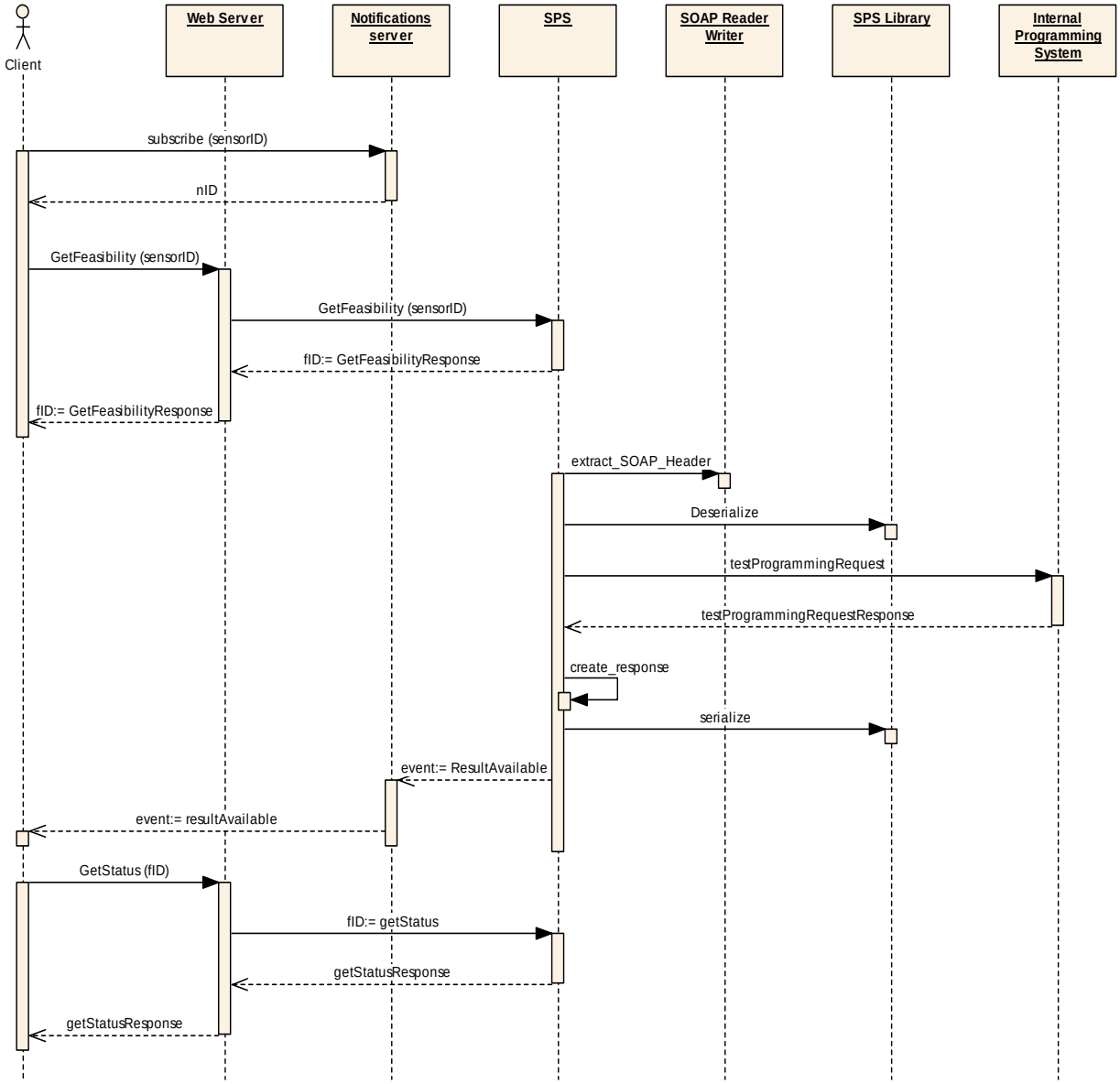

Figure 4-4: Sequence diagram for a HTTP POST operation with notifications

5. SYSTEM REQUIREMENTS SPECIFICATION

5.1. Requirement Naming Conventions

The following conventions will be used for the derived system requirement naming:

All system requirements will be named as

SR–XXX–NNNN/VER

where:

XXX represents the type of software requirement, which for HMA-T, are the following:

- **CON** – Software Configuration and Delivery Requirements
- **FUN** – Functional
- **IMP** – Design and Implementation Constraints
- **INS** – Installation
- **INT** – Interfaces
- **OPE** – Operational
- **PER** – Performance
- **RES** – Resources (Hardware and Software)
- **VVA** – Verification, Validation and Acceptance

NNNN Is a number providing an ordering within each requirement type. It starts at 0010 and two consecutive requirements are increased in 10 to allow the introduction of additional requirements in later versions of the document.

VER Is the issue of this document where the requirement was introduced or last changed.

Each requirement is presented in a tabular form constituted by four fields:

1. System requirement identifier
2. Validation method for the requirement:
   - **Test** (T) – Execution of the element under certain conditions to check the outputs corresponding to particular inputs;
   - **Inspection** (I) – Exhaustive evaluation of the code by manual reading;
   - **Analysis** (A) – Deduction method applied to documentation, code, test results, etc; it is partially or totally automated;
   - **Review** (R) – Review of project documentation.
3. Requirement text

5.2. Functional Requirements

5.2.1. General

SR-FUN-0010/1.0 T

SPS shall implement the following operations following the [OGC 07-018] specification:

- GetCapabilities
- DescribeTasking
- GetFeasibility
- Submit
- DescribeResultAccess
- GetStatus

SR-FUN-0020/1.0 T

SPS shall be developed and deployed twice (once by SPOT and once by DEIMOS) with some shared components. They shall use different underlying Programming Systems in order to carry out the GetFeasibility operation. They shall be providing different responses to the GetCapabilities and DescribeTasking requsests, reflecting the different internal systems.

SR-FUN-0030/1.0 T

The DEIMOS SPS shall have synchronous GetFeasibility processing.

SR-FUN-0040/1.0 T

The SPOT SPS shall have asynchronous GetFeasibility processing and a Notification Server.

5.2.2. Web server

SR-FUN-0110/1.0 R

SPS shall use the Apache/Tomcat Web server for hosting the SPS implementation.

SR-FUN-0120/1.0 I

The Web server shall be accessible for protocols as needed for the required operations – HTTP GET with information in Keyword-Value pairs and HTTP POST with information in SOAP messages. It shall support version 1.1 of the SOAP specification.

5.2.3. SPS Controller

SR-FUN-0210/1.0 T

SPS shall process the received messages in line with the [OGC 07-018] specification. This includes synchronous responses for each message type.

SR-FUN-0220/1.0 I

SPS shall populate the Java classes representing the SOAP responses, and pass these back to the SOAP Reader/Writer to be sent to the Client synchronously.

SR-FUN-0230/1.0 T

If necessary then the SPS shall send information to the Notification server to alert a client that a GetFeasibility request has been processed fully.

5.2.4. Notification Server

SR-FUN-0310/1.0 T

The Notification Server must support client subscriptions (cf. [RD Notifications]) to event types as defined in the capabilities of the SPS.

SR-FUN-0320/1.0 T

The Notification Server must contact the client to indicate that the GetFeasibility task is completed and that the client should send a getStatus message to request the result. This is triggered by the combination of the SPS Controller indicating that a planning task has completed and a client subscription including that task identifier.

5.2.5. SOAP Reader/Writer

SR-FUN-0410/1.0 I

SPS shall extract the SOAP headers from the incoming SPS requests. It shall support version 1.1 of the SOAP specification.

SR-FUN-0420/1.0 I

SPS shall append a SOAP header to the serialized information constituting a response to a given SPS request. It shall support version 1.1 of the SOAP specification.

SR-FUN-0430/1.0 I

SPS shall extract the SOAP body and shall pass it to the SPS Controller to trigger further processing. This extraction shall involve the creation and population of Java classes representing the same structures as appear in the SOAP input and output messages.

5.2.6. SPS Library

SR-FUN-0510/1.0 I

The implementation of the interfaces shall use the SPS library for deserialization of the incoming requests and the serialisation of the generated responses.

5.2.7. SPOT Internal Programming System

SR-FUN-0610/1.0 I

This SPS shall use a library that performs SPOT 2,4,5 feasibility analysis. This library performs the analysis by using the following criteria:

- orbitology
- climatology

The satellite workload is not taken into account.

5.2.8. DEIMOS Ground Segment Internal Planning System

SR-FUN-0710/1.0 I

This SPS shall use the Earth Explorer CFI as a COTS Satellite Programming System for the implementation. It shall perform the analysis immediately, so that a response can be sent back to the client synchronously.

SR-FUN-0720/1.0 I

The implementation of the interface related to the Earth Explorer CFI shall use the JNI library as a bridge between the C and Java languages.

SR-FUN-0730/1.0 I

The Earth Explorer CFI shall be configured to simulate the Sentinel 1 mission. This simulation shall not be precise, but shall allow a reasonable set of parameters to be used in the messages passed between the Client systems and the SPS.

5.3. Performance Requirements

SR-PER-0010/1.0 T

SPS shall be capable of being deployed on a single server (for each implementation).

SR-PER-0020/1.0 T

SPS shall respond rapidly enough to all message receptions that sensible time-out parameters can be set for any client software. 60 seconds is the anticipated value, although this may be changed during testing if necessary.
This is the benchmark when a single message is being processed at once. In case of multiple clients connecting at the same moment it is permissible (though not expected) that one should be given a time out by the SPS in the understanding that the client will retry later.

5.4. Interface Requirements

SR-INT-0010/1.0 T

Spot Image SPS shall be deployed in Spot Image facilities. TBD whether the DEIMOS SPS shall be deployed on servers within DEIMOS facilities or at ESRIN. In any case they shall all be accessible through the Internet by an operator using the SSE if a compatible client exists within the SSE.

5.5. Operational Requirements

SR-OPE-0010/1.0 R

SPS is not intended to be an operational system. The developed systems shall be maintained only on a best-effort basis.

5.6. Resources Requirements

SR-RES-0010/1.0 T

DEIMOS SPS shall be deployed on a server running Linux and the Apache Tomcat web application server.

SR-RES-0020/1.0 T

Spot Image SPS shall be deployed on a server running Windows Server 2003 and the Apache Tomcat web application server.

5.7. Design and Implementation Constraints

SR-IMP-0010/1.0 R

The SPS development process shall be based on [ECSS-40-1B].

SR-IMP-0020/1.0 R

The SPS requirement analysis shall be performed using UML, including the production of use cases and

interaction diagrams for the various message types

SR-IMP-0030/1.0 R

The SPS design shall be performed using UML

5.8. Software Configuration and Delivery Requirements

SR-CON-0010/1.0 R

The configuration aspects applicable for the project shall be defined in the Software Configuration Management Plan, included in [RD PMP]

5.9. Adaptation and Installation Requirements

SR-INS-0010/1.0 R

For any software components delivered for deployment at ESRIN instructions shall be produced regarding how to install the software. This could include information for the interface implementation and/or the SPS back-end component.

5.10. Verification, Validation and Acceptance Requirements

SR-VVA-0010/1.0 R

In order to support the demonstration activities in the Acceptance Review a System Test Plan shall be written and system testing shall be carried out at DEIMOS and SPOT resulting in Validation Testing Reports. These tests shall be based on, and shall reference, this Software Requirements Document and the SPS EO Application profile [OGC 07-018]. While these prototypes may be useful for developing SPS EO CITE tests there is not CITE testing anticipated within this project.