E2E mission performance Simulators for Space Science missions

Executive Summary

Code : SS-E2ES-DME-TN-011
Issue : 2.0
Date : 08/01/2016

	Name	Function	Signature
Prepared by	José Barbosa	Project Engineer	
	José Julio Ramos	Project Engineer	
	Steve Guest	Project Engineer	
	José Lopez Moreno	Project Engineer	
Reviewed by	José Barbosa	Project PA Team	
Approved by	José Barbosa	Project Manager	
Signatures and approvals on original			

DEIMOS Engenharia, S.A.
Av. D. João II, Lote 1.17.01 Edifício Torre Zen 10º
1998-023 Lisboa, PORTUGAL
Tel.: +351 21 893 30 10 / Fax: +351 21 896 90 99
E-mail: deimos@deimos.com.pt

This page intentionally left blank

Document Information

Contract Data	
Contract Number:	**4000112220/14/NL/MV**
Contract Issuer:	ESTEC

Internal Distribution		
Name	Unit	Copies
José Antonio González	Head of the Earth Observation Business Unit	1
Antonio Gutierrez	Ground Segment Business Unit Director	1
Internal Confidentiality Level (DMG-COV-POL05)		
Unclassified ☑	Restricted ☐	Confidential ☐

External Distribution		
Name	Organisation	Copies
Joachim Fuchs	ESA	1

Archiving	
Word Processor:	MS Word 2007 (Word 97-2003 Compatibility Mode)
File Name:	SS-E2ES-DME-TN-011-Executive-Summary-R20-E.docx

Document Status Log

Issue	Section	Change Description	Date
1.0	All	Final Delivery Version	15-Dec-2015
2.0	All	Revised, summarized, version	08-Jan-2016

Table of Contents

This page intentionally left blank **2**
Document Information **3**
Document Status Log **4**
Table of Contents **5**
1. Introduction **7**
1.1. Purpose and Scope **7**
1.2. Applicable Documents **7**
1.3. Reference Documents **8**
2. SS MIssions Categorization and Commonalities **9**
2.1. Space science missions' categorization **9**
2.1.1. Mission Type Categorization Classes 9
2.1.2. Instrument Categorisation Classes 11
2.1.3. SS Mission Categorization 12
2.1.4. Data Product Levels 12
3. Reference Architecture Definition **14**
3.1. Requirements for the SS-E2ES Reference Architecture **14**
3.2. Reference Architecture Definition **16**
4. Building Blocks Specification **21**
5. Mission Selection and RA Evaluation Criteria **25**
5.1. Mission Selection and Comparison **25**
5.2. Mission Comparison and SS mission type coverage **25**
6. ExoMars and Solar Orbiter Requirements Specifications **27**
6.1. ExoMars High Level Concept **27**
6.1.1. Simulator Context 27
6.1.2. Stakeholders Definition 27
6.2. Solar Orbiter High Level Concept **29**
6.2.1. Stakeholders Definition 29
6.3. ExoMars and Solar Orbiter E2E Simulator Software Requirements **30**
7. ExoMars and Solar Orbiter Architecture Design **31**
7.1. Solar Orbiter Architecture Design Specification **32**
8. RA Application Evaluation and Roadmap **34**
8.1. Evaluation exercise definition **34**
8.2. Assessment of Reference Architecture Application **34**
8.3. Roadmap Proposals **35**
8.3.1. In-Situ Instruments 35
8.3.2. Harmonisation of SS and EO architectures 35
8.3.3. Survey of Libraries 35
8.3.4. Programming Languages in Space Science 36

8.3.5. Data Products 36
8.3.6. Tools and Frameworks 37
8.3.7. Use the architecture for a real project 37
9. Conclusions 38

1. INTRODUCTION

1.1. Purpose and Scope

This document summarizes the conclusions of the Space Science End-to-End Simulators Reference Architecture project. It will present a summary of the main conclusions achieved in each of the studies phase and provide an overall general conclusion to the project.

The document is structured in the following sections:

- Introduction, this Section
- In Section 2 we present a summary of the Categorization and Commonalities Technical Notes (TN-001 and TN-001a)
- In Section 3 we present a summary of the Requirements and Reference Architecture exercise (from TN-002)
- In Section 4 we summarize the contents of the Building Blocks Specification document (TN-003)
- In Section 5 we show how two SS missions were selected for RA application and how the evaluation criteria were defined (TN-004)
- In Section 6 we show how the Requirements Specification was done for both selected missions (TN-007 and TN-009)
- In Section 7 we summarize the Specific Architectures for ExoMars and Solar Orbiter (TNs 008 and 010)
- In Section 8 we present the summary of TN-011, containing the Evaluation of the RA application and Roadmap recommendations
- Section 9 presents the overall conclusion for the SS-E2ES activity

1.2. Applicable Documents

Ref.	Code	Title	Issue
[SOW]	Appendix 1 to AO/1-7814/14/NL/MV	Statement of Work for the E2E mission performance Simulators for Space Science missions (SS-E2ES)	1.0
[Proposal]	SS-E2ES-DMS-PRL-2014-892	Proposal to ESA for the E2E Mission Performance Simulators for Space Science Missions	-
[Q00A]	ECSS-Q-00A	ECSS Space Product Assurance – Policy and Principles	Issue A
[Q20A]	ECSS-Q-20	ECSS Space Product Assurance – Quality Assurance	
[Q20-09A]	ECSS-Q-20-09A	ECSS Space Product Assurance – Non conformance control system	
[Q80C]	ECSS-Q-ST-80C	ECSS Space Product Assurance – Software Product Assurance	6 March 2009
[E40C]	ECSS-E-ST-40C	ECSS Space Engineering (Software)	6 March 2009

1.3. Reference Documents

Ref.	Code	Title	Issue
[RD.1]	GMV-ARCHEO-E2E-FR-001	ARCHEO Final Report	1.0
[RD.2]	N/A	The Concept of Reference Architectures, Systems Engineering Vol. 13.	No.1, 2010
[RD.3]	N/A	"Introducing the European Space Agency Architectural Framework for Space-based Systems of Systems Engineering" by Daniele Gianni, Niklas Lindman*, Joachim Fuchs and Robert Suzic - European Space Agency	N/A
[RD.4]	SS-E2ES-DME-TN-001	Space Science Missions and Elements Categorization	1.2
[RD.5]	SS-E2ES-DME-TN-001a	Space Science Missions Commonalities	1.1
[RD.6]	SS-E2ES-DME-TN-002	Requirements and Reference Architecture	1.4
[RD.7]	SS-E2ES-DME-TN-003	Generic Building Blocks General Specification	1.4
[RD.8]	SS-E2ES-DME-TN-004	Evaluation Method and Mission Selection	1.1
[RD.9]	SS-E2ES-DME-TN-007	ExoMars E2E Simulator Software Requirements Specification	1.0
[RD.10]	SS-E2ES-DME-TN-008	ExoMars E2ES - Simulator Architecture Design Document	1.0
[RD.11]	SS-E2ES-DME-TN-009	Solar Orbiter E2E Simulator Software Requirements Specification	1.0
[RD.12]	SS-E2ES-DME-TN-010	Solar Orbiter E2ES - Simulator Architecture Design Document	1.0
[RD.12]	SS-E2ES-DME-TN-005	RA Application Evaluation and Roadmap	1.0

2. SS MISSIONS CATEGORIZATION AND COMMONALITIES

2.1. Space science missions' categorization

The term "Space Science" encompasses the study of the state and evolution of the Solar System and the Universe. Even considering that SS missions share many characteristics with EO missions, they possess specific differential characteristics:

- There are a wider variety of science objectives. From distant galaxies to solar system bodies, and from electromagnetic to gravitational waves, the objectives of the SS missions are any detectable phenomena that can provide any information on the Universe and Solar system evolution.
- The SS missions, devised towards each corresponding objectives, tend to provide a wider variety of missions, including instruments, SC platforms, trajectories, etc. Correspondingly, any aspect of the mission designs tends to be more innovative, requiring more ad-hoc development and, in some cases, preventing the re-use of elements which could reduce mission costs.
- SS missions can be more challenging than EO missions in terms of schedule, cost and operations. For example:
    - Solar System exploration missions are designed for specific launch windows, which makes the complete mission preparatory work useless in case the window is missed
    - The distance to the Earth reduces the bandwidth available for download
    - There are communications black-out periods during conjunctions and oppositions

Even with the previously mentioned differences between SS and EO missions, there are plenty of commonalities which can be identified. For that we have to categorize them, similar to what was done in [RD.1]. The categorization can be based on two main properties of the mission: the overall mission configuration and the instrument types on-board each mission. In the next two Sections we will show the main axes for finding the main criteria in each property set.

2.1.1. Mission Type Categorization Classes

We can define several criteria to define mission configurations. In the following list, the main criteria considered for categorization are shown. For a full analysis, please refer to [RD.4]

- **Type and purpose of the mission**. This is the most important criteria, in the sense that it drives the overall mission design. We can mention the following categories:
    - Astronomy
        - In this class, beyond the classical Astronomy missions, we included also the Astrophysics and Fundamental physics missions, since they are concerned mostly with the observation of deep space
    - Exoplanet
        - Missions concerned with finding exoplanets, such as EchO, PLATO and CHEOPS.
    - Planetary observation
        - These are the missions that resemble the most the EO-type missions. They might include landers but these will be excluded from our categorization, unless they contain some sort of remote sensing instruments.
    - Solar system exploration
        - Missions to explore other solar system bodies: comets, moons, asteroids, such as Rosetta.
    - Solar

        - Sun observation: corona, magnetic field, solar wind, etc.
    - Demonstration
        - These are missions such as Proba-2 and Proba-3, that contain a series of state-of-the-art instruments and demonstration technologies
    - Astrometry (GAIA, successor to Hipparcos)
        - We selected this class for GAIA, since it is a fundamentally different mission from all the ones considered, not only in terms of objectives but also in terms of Data Processing, Archiving and Distribution. It is a clear outlier and a good example of a mission for which it would have been very difficult to adapt a generic Reference Architecture
- **Type of orbit**. It is obviously related to the scientific objective of the missions.
    - Earth-bound orbit, looking up, frequently used by astrophysics missions
        - Low Earth orbits (LEO), high elliptical orbits (HEO)
    - Lissajous orbit around the Lagrangian points of the Sun-Earth system, most commonly, L2.
    - Sun-centered orbit (excluding Lissajous)
        - For sun observation
        - For body fly-bys
    - Body-bound orbits, which observe the target body in a similar way as EO missions observe the Earth
        - Planetary-bound
        - Moon or comet-bound
- **Observation geometry/scanning method**. It will depend on the scientific objective and on the selected instruments.
    - 3-axis stabilised
    - Spinner
    - Lander
- **Data product format**
    - FITS, typically used for Astronomy and Solar missions
    - PDS is a format mostly used for Planetary missions
- **Reference coordinate system** (this is part of constructing the scene, e.g. galactic for Planck)

The previous mission types are obviously interconnected. For example, Lissajous orbits around the L2 point of the Sun-Earth system are used for astrophysics missions due to the stable environment, while the L1 point is a privileged location for Sun observation from the ecliptic plane.

Even within the same criterion, SS missions can have different categories depending on the mission phase. For example, a comet orbiter mission will have a Sun-centred phase, with designed asteroid fly-bys, as in Rosetta.

From the previous classification, we can already identify the following missions as having similarities with EO missions in terms of orbit or observation geometry:

- Astronomy missions on LEO orbit show similar orbit dynamics as EO missions
- Solar System planet-bound exploration missions resemble EO missions in terms of orbit geometry (obviously, with the central gravity corresponding to the target body), and in terms of instruments

2.1.2. Instrument Categorisation Classes

In terms of Instrument Type criteria to categorize the missions, the obvious problem is that the variety of instruments designed for space exploration is very high. During the course of the study, the following criteria were considered for instrument categorization:

- **Detector Type**, this category is the most important in defining the type of L1 processing. The most common instruments in SS are as follows:
  - Imagers
    - CCDs (used not only for imaging but also for photometry)
    - Bolometers (filled bolometers, micro-bolometers, etc.)
    - Active Pixel Sensors
    - Interferometric Imaging Radiometers
  - Spectrometers (this class is not completely independent from the previous one, an instrument can be an imaging spectrometer)
    - Different spectrometer types: heterodyne, FFT etc.
  - Polarimeters
  - Coronagraphs
  - Radiometers
  - Radar sounders
  - Altimeter
- **Waveband**. There is, of course, a very strong correlation with the detector types listed in the Detector Type category:
  - Gamma-rays
  - X-Rays
  - Ultra-Violet
  - Visible
  - Infrared
  - Submillimetre
  - Microwave
- **Passive or active** instruments
  - Passive instruments detect electromagnetic radiation emitted or reflected by the events being observed.
  - Active instruments themselves emit radiation, which is directed toward the target to be investigated. The radiation reflected from that target is detected and measured by the instrument. This technique is most commonly used in altimeters, typically using laser or radar pulses.
  - In astronomy missions, instruments are always passive. For missions observing solar system bodies, they can be either active or passive.
- **Type of detection**:
  - Interferometric
  - Direct detection
- **Target** of the measurement
  - Planetary

        - Surface (Land, Water and/or Ice)
        - Atmosphere
        - Sub-surface
    - Radiation (or Deep Space)
        - Astronomy missions studying radiation localized sources such as stars, pulsars, galaxies, x-ray sources, etc.
        - Missions studying extended sources, such as the Cosmic Background radiation (Planck mission)
        - Sun radiation
    - Gravity and magnetic fields
    - Radiation-atmospheric interaction (Cluster)
- **Type of retrieval products**
    - Images
    - Spectra
    - Spectral cubes
    - Light curves
- **Scanning geometry**
    - Independent of platform
    - Determined by platform attitude

Categories can, of course, be nested. As an example, we can categorize the Instrument Type within each mission by Type.

2.1.3. SS Mission Categorization

Using the categories outlined in the previous sections, a database was populated for all the missions under analysis, the SS Mission Database. Each mission was classified in each of the categories and the results analysed. This allowed for a selection of the most relevant categories and elimination of the less useful or redundant ones.

The most relevant category found was the Mission Type (Astronomy, Exoplanet, Planetary, etc.), which is very strongly correlated with Scene Models, Geometrical Models and Orbital Parameterization. There was also a strong correlation with the Product Format: in general, all Astronomy Missions use the FITS format while all the Planetary Missions use the PDS format.

As for the categories related to the mission instrumentation, the most relevant were:

- **Mission Type**
- **Instrument Type** (Imager, Spectrometer, etc.)
- **Waveband Class**
- **Detector technology**, both in type and composition (very strongly correlated with Waveband)

A very strong correlation was observed between these last two categories, of course: CCD imagers are used for measuring the visible waveband while antennas are used to probe the microwave spectrum. Several queries were performed to the database, in search of the clearest way to organize the Mission Data and the results reported in the full version of this TN, [RD.4].

2.1.4. Data Product Levels

Space science missions typically define *levels* of data processing to describe to what extent the data has been processed. There is no standard way of describing these levels for ESA missions, let alone in the

world community. However, there is some degree of commonality. Almost all missions define their levels with a numeric rather than a descriptive name, and usually in the range 0-3.

The CODMAC standards were developed by a National Research Council Committee on Data Management and Computation (CODMAC) in the 1970s. It appears to be most prevalent in planetary missions. For our purposes we define levels in a more "Space Science ESA" way, although as already mentioned, there is no standard. Rough "NASA" equivalents are indicated (please note that the ESA EO – but not SS - levels are very similar to the NASA levels)

Ref Level	NASA level	Summary	Description
Telemetry	Packet Data	Raw telemetry	Telemetry packets. May be packaged into a file format e.g. ESOC's DDS.
0	Level-0	Raw data	Raw data reformatted from telemetry into a more generally useful format.
0.5	Level 1-A	Raw data in physical units	Data converted into physical units e.g. volts. Often this process will be reversible, but may not be, at least not without loss of accuracy.
1	Level 1-B	Calibrated data	Data calibrated to remove instrumental effects. (This is the goal; in practice some effects may not be removable before the next level).
2	Level 1-C	Science data	Fully calibrated and scientifically useful data products, e.g. images, spectra, spectral cubes.
3	Level 2	Derived data	Further science data products extracted from, or by combining, the Level-2 data. Examples are mosaiced maps, line lists, point source lists.

These reference levels are the definitions as referred to in the commonality analysis.

As for the format of the data itself, this study will refrain from recommending a new, more generic, format. Not only because the format itself does not affect the Reference Architecture or Building Blocks (at most just the naming of the Product Input/Output Library Blocks) but because there are two very different formats already existing and in use for many years for Astronomy (FITS) and Planetary (PDS) missions.

3. REFERENCE ARCHITECTURE DEFINITION

3.1. Requirements for the SS-E2ES Reference Architecture

The requirements in the [SoW] were applicable to the E2ES simulator and Simulation Framework, regardless of the development model used for the simulator (either using or not using the Reference Architecture concept). However, the requirements on the simulator should be obtained by inheriting the requirements on the Reference Architecture itself, together with other requirements not captured in the Reference Architecture and/or specific to the mission.

The updated Reference Architecture requirements were laid down following these guidelines:

- Generalization of the SS-E2ES SoW requirements (bottom-up approach)
- Analysis of mission and instrument commonalities, performed in the Categorization and Commonalities TNs ([RD.4] and [RD.5])
- Experience in E2ES development
- Practical philosophy: common practices, future re-usability and usefulness.

The requirements in the [SoW] were organized in the following types:

- **E2ES Requirements**
  - Functional requirements
  - Design requirements
  - Interface requirements
  - Performance requirements
- **E2ES Simulation Framework Requirements**
  - General requirements
  - Modules and interface requirements
  - Control requirements
  - Data storage and visualization requirements

After careful analysis, this classification was considered in need of some improvement, starting with the fact that there were no specific requirements for the Reference Architecture itself.

In fact, it is useful to separate the requirements according to who is going to read and apply them. There are 4 main targets for the Requirements to be defined in this activity:

- **Reference Architecture:** There should be a separate group of requirements applicable to the Reference Architecture to be designed as a part of this activity. They should include the definition of Reference Architecture, the SoW objectives and main tasks and, lastly, the Requirements for E2E Simulators that are so generic that they can be part of the Reference Architecture definition itself. These RA Requirements will directly impact the current activity but will not be an input for the Space Science Simulator Teams. They will be used for the evaluation of the final Reference Architecture. These requirements affect implicitly the Specific Architecture, through the design of Reference Architecture itself.

- **Specific Architecture:** This is the Architecture for each Space Science End-to-End Simulator, to be designed based on the Reference Architecture. There should be a set of Requirements applicable to the Architectures to be designed by the Space Science Teams. This set of requirements is one of the most important outputs of this activity and it will be the "instruction manual" on how to apply the Reference Architecture to each Mission needs. These Requirements are intended to be used more as a checklist than as contractually hard requirements and their

objective is to help simplify work. They will also be used in this activity for the evaluation of the Specific Architectures to be designed for the Eculid and ExoMars missions.

- **Software Framework:** This will be the Framework to be used to support the Simulator execution. The requirements in the SoW for this activity will be used as a basis for a recommendation for the Software Framework to be used for the Space Science E2ES. However, the output will be no more than a recommendation of a Software Framework. This set of requirements should also be transmitted to the Space Science Teams in case that wish to evaluate other Software Frameworks – then the requirements can be again used as a checklist, supporting the assessment of any Software Framework under analysis.

- **Software Implementation**: After the Specific Architecture for each mission is completed and approved, it must be implemented into code by the Space Science E2ES teams. We will draft also a series of requirements that will help these teams making the transition from Architecture to Implementation (or paper to code). These requirements will be an output of this activity and they are intended to be used again as a checklist.

In the next figure we show how the Requirements designed for each of the four targets will be applicable:

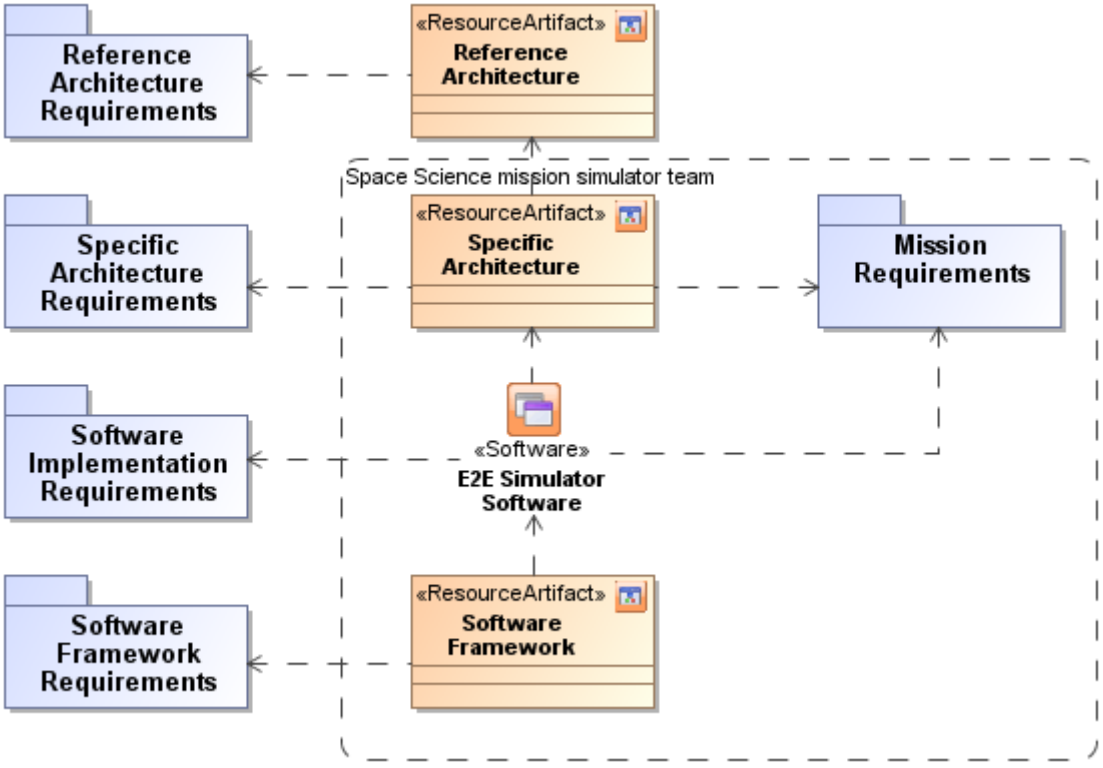

Figure 3-1: SS-E2ES Requirements Applicability

Outside the limited area are shown the activities that will be performed by the SS-E2ES Reference Architecture team. The team will define the four levels of requirements and the Reference Architecture. Please note that the Reference Architecture Requirements apply directly to the work performed in this project, and are not an input for the team implementing the simulator. Obviously, they have an implicit impact on the specific architecture of each mission.

The SS Mission Teams activities are within the limited area. The SS Teams will have to implement a Specific Architecture based on the Reference Architecture and on the Specific Architecture Requirements. They will then have to Implement a E2E Simulator based on their Specific Architecture and the Software

Implementation Requirements. Finally, they will have to select a Software Framework either by accepting the Reference Architecture team recommendation or by applying the Software Framework Requirements to select one.

From the above diagram it is clear that the inputs for the SS Mission Teams are the Reference Architecture itself and the Specific Architecture, Implementation and Software Framework requirements.

3.2. Reference Architecture Definition

For this activity, we chose to adapt ESA-AF, an Architectural Framework developed by ESA, to our particular needs. We don't need to cover all the framework intricate details to meet the defined requirements.

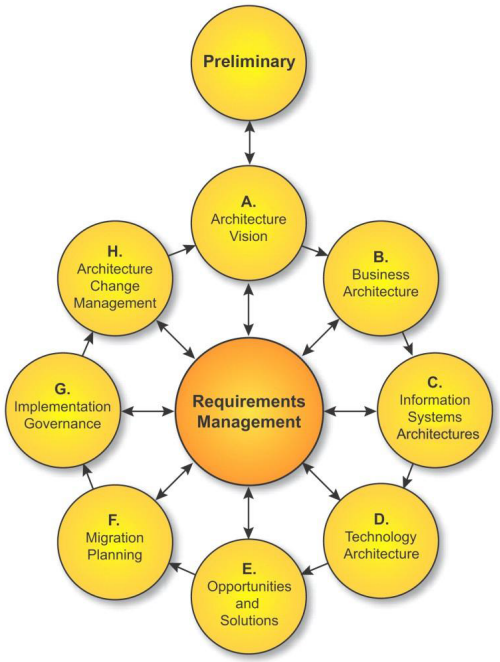

Figure 3-2: TOGAF lifecycle

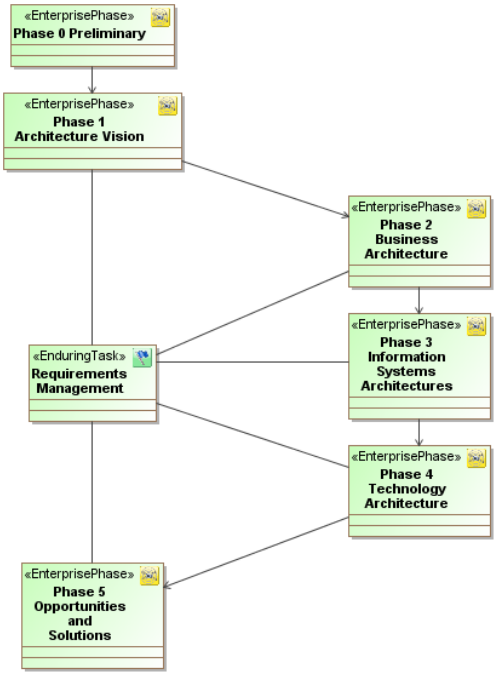

Figure 3-3: Adapted SS-E2ES lifecycle

The business process, the lifecycle this enterprise architecture follows is an adaptation of the one defined in the TOGAF standard, shown in Figure 3-2. This process has been adapted to the real needs of this project, thus it follows the steps shown in Figure 3-3:

0. Preliminary
    - This first step is devoted to define the framework and detailed methodologies to be used to develop the enterprise architecture.
1. Simulator Context. The objectives of Phase 1 are to
    a. To define the scope of, and to identify and prioritize the components of the Reference Architecture effort
    b. To define the relevant stakeholders, and their concerns and objectives.
    c. To define the key business requirements to be addressed in this architecture effort, and the constraints that must be dealt with.
    d. To plan the incremental deployment of the simulator capabilities (operational activities) all over the development phases of a Space Science mission.

These objectives are supported by the use of ESA-AF Operational Views (OV).

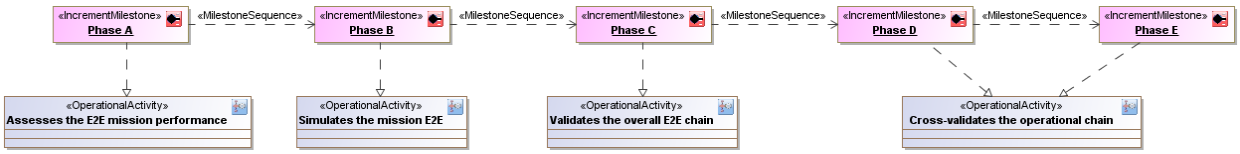

Figure 3-4: Capability phasing (StV-3 Capability Phasing)

In order to achieve these steps we have performed a series of surveys and analysis of the currently available architectures for E2E simulators and technologies:

a. Survey and analysis of existing individual architectures for E2E Space Science Simulators, in order to capture their essence and find generalizations and commonalities, that could be part of our Reference Architecture.
b. Survey of the available technologies (simulators, models, libraries and repositories).

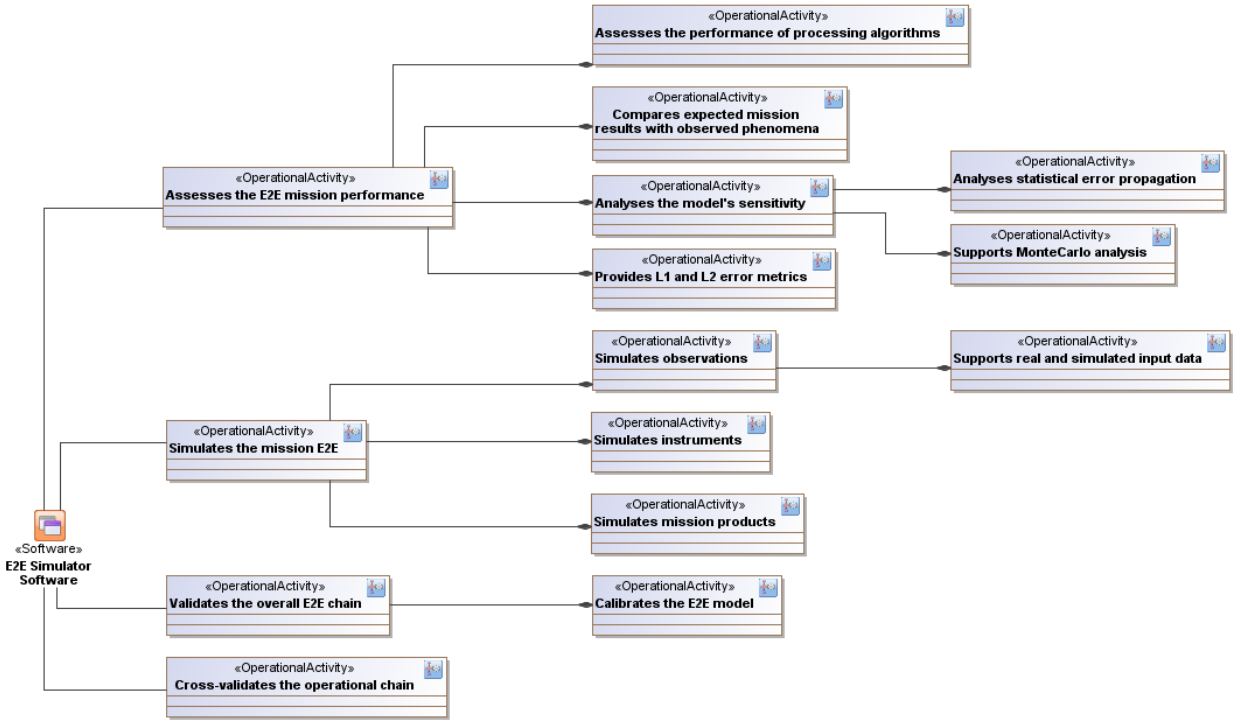

Figure 3-5: E2ES capabilities (OV-6a Operational Rules Model)

2. Simulator Overall Architecture. The objectives of Phase 2 are to:
   a. To describe the main processes part of an E2ES, i.e. to introduce the Reference Architecture with two views: the E2ES context, a more detailed view of the simulation chain (in vertical layout or in a loop layout).
   b. To select the relevant architecture viewpoints that will enable the architect to demonstrate how the stakeholder concerns are addressed in the Reference Architecture

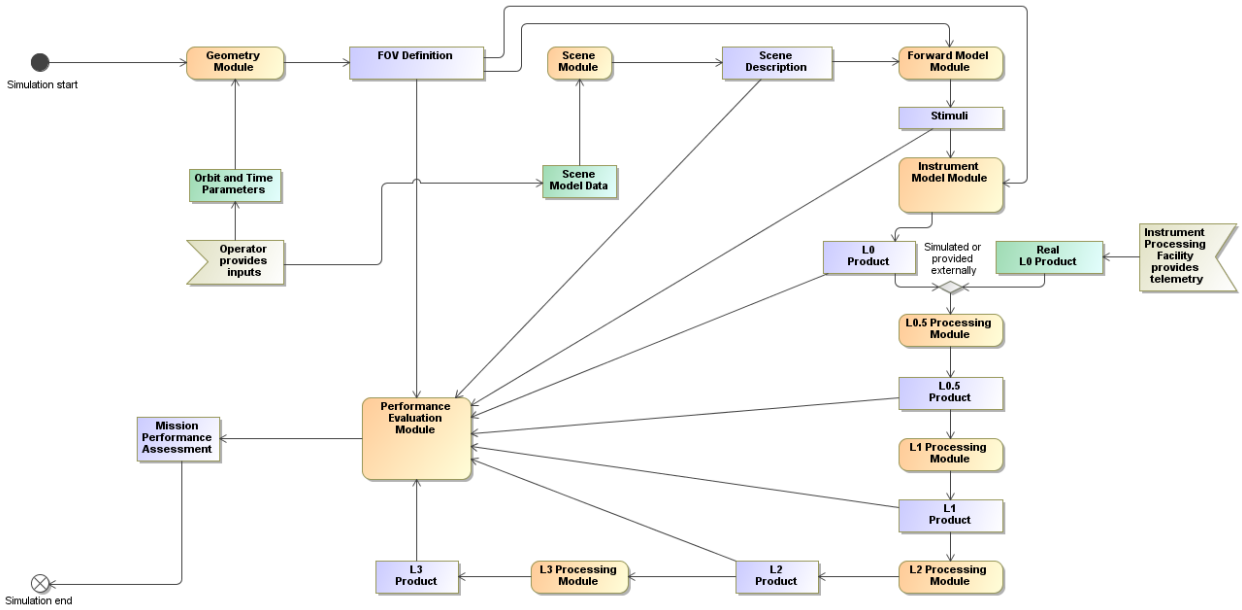

Figure 3-6: Overall architecture - Loop layout

3. Simulator Architecture Specification
    a. The objective of Phase 3 is to develop Target Architectures covering both of the Data, Business and Application Systems domains.

This objective is supported by the use of ESA-AF System Views and the following steps:

a. High-Level Architecture Design - To define the high-level modules (building blocks) for end-to-end simulators. According to the SoW, this analysis has been based on the results of the Archeo-E2E project for Earth Observation missions, but also findings from the previous analysis of individual architectures and, of course, the results of Task 1 "SS missions & instruments categorization".
   Within this activity, the Reference Architecture describes three important aspects of the E2E simulator:
   - Mission Categorization - Listing several ways of categorizing Space Science missions.
   - Building Blocks - Defining the high-level modules (building blocks) and main data flows on an E2ES.
   - Combination of Building Blocks - Giving some examples on how to combine the building blocks for creating different simulation chains for several mission categories.

b. Data Specification - To define the major types and sources of data necessary to support the E2ES, including products, ADFs and model configuration parameters.
   More in detail:
   - Data Products - Defining the major data products (final or intermediate) involved in the E2ES chain.
   - Configuration Data - Defining the configuration parameters altering the behaviour of the E2ES chain.
   - Auxiliary Data - Defining the Auxiliary Data Files supporting the E2ES chain.

c. Building Blocks Architecture Design - To describe the system structures, in this case end-to-end simulator building blocks, defining models on different granularity levels for each structure.

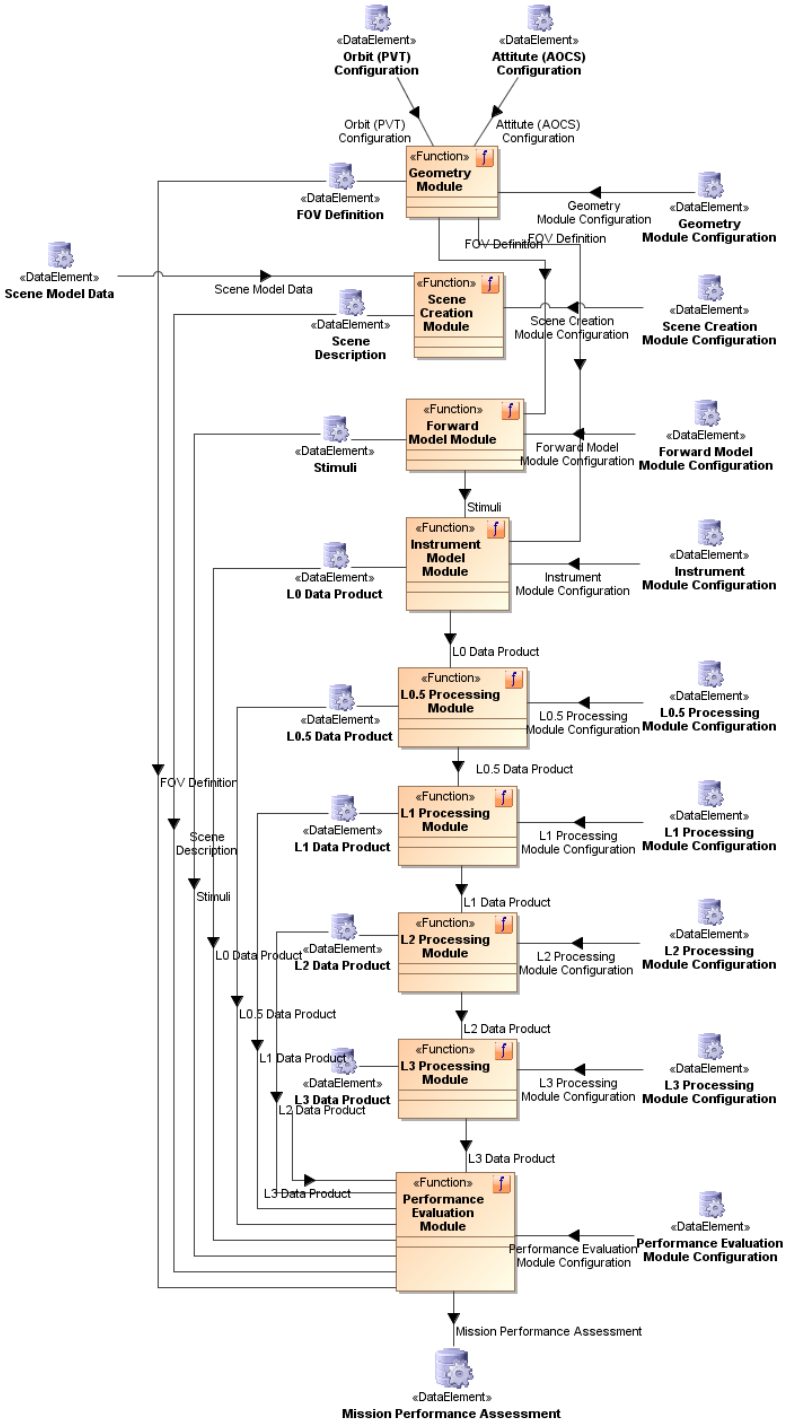

Figure 3-7: Building Blocks (SV-1a Resources Interaction Specification)

4. Candidate Solutions
    a. The objective of Phase 4 is to develop a Technology Architecture that will form the basis of the following implementation work.

This objective is supported by the use of ESA-AF Technology Views and the following steps:

a. Analysis of available technologies for the eventual implementation of instances of the reference architecture, including simulation frameworks, processing libraries and model repositories.

5. Reference Architecture Instantiation. The objectives of Phase 5 are to:
   a. Evaluate and select among the implementation options identified in the development of the various Target Architectures.
   b. Identify the strategic parameters for change, and the top-level work packages or projects to be undertaken in moving from the current environment to the target.
   c. Generate an overall implementation strategy and a detailed Implementation Plan.

In order to support these objectives, two detailed implementation plans will be created for two demonstration missions: Exomars TGO and Solar Orbiter.

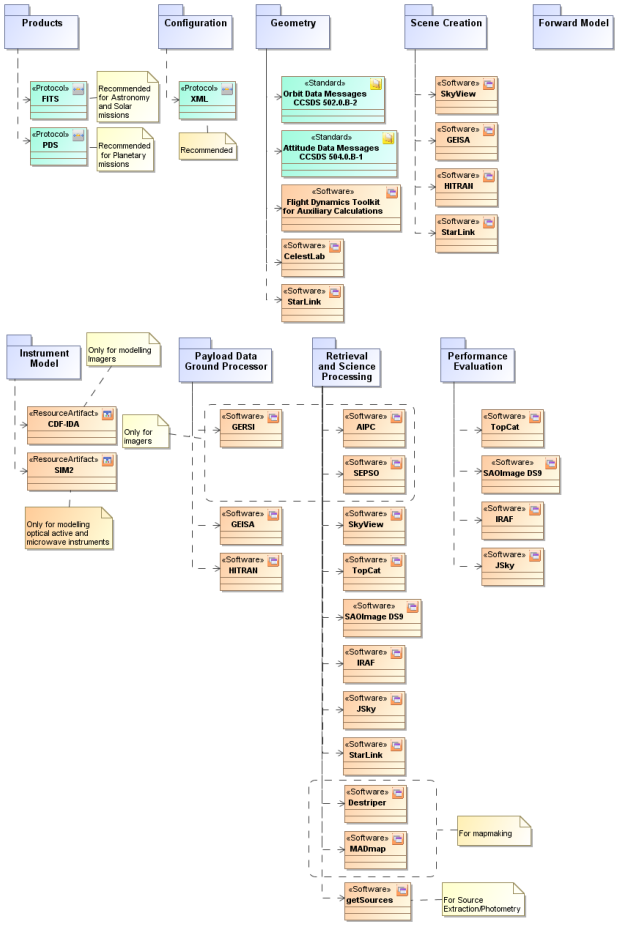

Figure 3-8: Processing libraries

Again, to make sure the selected EAF really helps in its role of describing the SS-E2ES Reference Architecture, this consortium has followed some lessons learnt by ESA teams in similar projects, as these ones:

1. Use as clear and clean as possible view descriptions.
2. Hide meta-modelling complexity and do not over-explain the framework (stakeholders want robust solutions to the problems and are typically not interested in framework details).
3. Use as few as possible different types of (meta-modelling) artefacts.

4. BUILDING BLOCKS SPECIFICATION

The Building Blocks document is very large and detailed. In the next table is a summary of the document contents, showing all the Building Blocks that can be found inside. Navigating from the larger to the more detailed categories, the desired module Building Block example should be found quickly. Please note that for all major classes, the most generic Building Blocks are described in an initial section:

Table 4-1: Document Map

Processing Level	Category	Building Block	
Geometry Simulation	Generic	Orbit Simulator	
		Attitude Simulator	
		Instrument Pointing Simulator	
Scene Creation	Generic	Generic Blocks	
	Astronomy Mission	Astroscene Module	
	Exoplanet Mission	Exoplanet Astroscene	
		Black Body Emissions Calculator	
		Planetary Spectral Emission Module	
		Exoplanet Model	
		Exoplanet Orbital Model	
		Stellar Flux Calculator	
	Solar Mission	Stellar Emission Model	
Forward Model	Generic	Scene Interaction Geometry	
		Stimuli Generation	
	Planetary Mission	Atmosphere Model	
		Surface Model	
		Planetary Emission Calculation	
		Sky Maps	
Instrument Model	Generic	Optics Building Block	
		L0 Formatter	
	Antenna	Radar Altimeters	Swath Size
			Resolution
		Radar Sounder	Vertical Resolution
			Attenuation
			Along and Across Track Horizontal Resolution
		Imaging Radiometers	
		Sounding Radiometers	
	Photodiodes	LIDAR	Cross-Talk Factors computation
		Laser Altimeters	Echo Pulse width Estimation
			Background Photons
	CCD	Imagers	Preprocessing

			Detector
			Signal Chain
	Active Pixel Sensor	Generic Blocks	
	Bolometer	Generic Blocks	
	Heterodyne Mixers	Generic Blocks	
	Micro-channel Plate	Generic Blocks	
	Scintillators	Generic Blocks	
	Semiconductor Array	Generic Blocks	
L0 to L0.5 Processing	Generic	Unpack Telemetry	
		Decompression	
		Sorting	
		Repackaging	
		Add Auxiliary Data	
		Unit Conversion	
		Time Correction/Conversion	
		Masking	
		Data Extraction & Quality Control	
		Measurement Pre-Processing	
		Time Domain Integration	
	Imagers	Generic Blocks	
	Radiometers	Generic Blocks	
	Spectrometers	Generic Blocks	
	Sounder	Generic Blocks	
	Particle Detector	Charged Particle Discrimination	
		Efficiency Saturation Effect Correction	
	Coronograph	Generic Blocks	
	Altimeters	Generic Blocks	
L0.5 to L1 Processing	Generic	Cosmic Ray Removal / Deglitching	
		Integration Ramps Reconstruction	
		Flux Calibration	
		Baseline Subtraction	
		Pointing Errors (Jitter) Compensation	
		Dark Current Subtraction	
		Crosstalk	
		Linearity	
		Velocity Correction	
		Non-Linearity Correction	
		Thermal Drift Corrections	
		Demodulation	
		Radiometric Calibration	

		Data Compression	
	Imagers	Detector Modulation Transfer Function	
		Flat Field Correction	
		Vignetting Removal	
		CCD Fixed Pattern Noise Removal	
		Point Spread Function (PSF) Calculation	
		Flux Density Conversion	
		Electrical Filter Response Correction	
	Spectrometers	Phase Correction	
		Fourier Transform Spectrometer Phase Correction	
		Telescope Emission Calculation	
		Transient Correction	
	Radiometers	Generic Functions	
	Sounder	Radar Sounder - Signal-to-Noise Ratio and Signal-to-Clutter Ratio	
		Compute Calibrated Signal	
	Particle Detector	Generic Functions	
		Sporadic Solar Event Correction	
		Galactic Cosmic Ray Removal	
		Temperature Gradient Correction	
		Altitude Variation Correction	
	Altimeters	Radar Altimeter	Received average power
			Signal to Noise Ratio
		Laser Altimeter	Signal to Noise Ratio
			Probability of False Alarm and Probability of Detection
			Range Gate
			Vertical Ranging Resolution
			Horizontal Ranging Resolution
L1 to L2 Processing	Generic	Projection	
		Mapmaking	
		Weighted Averaging	
		Regridding	
		Denoising	
		Deconvolution	
		Pixel Flagging	
	Imagers	Long Term / Persistent Transient Correction	
		Baseline Removal / Destriping	
	Spectrometers	Interferogram to Spectral Domain Conversion	
		Apodization	
		Out of Band Power Removal	

		Spectral Rebinning	
L2 to L3 Processing	Generic	Source Extraction	
		Mosaicking	Resampling
			Co-Registration
			Tie Point Selection
			Tie Point Matching
			WCS Projection
			Deregistration Estimation Model
			Pixel Classification
Performance Evaluation	Generic	Generic Blocks	

5. MISSION SELECTION AND RA EVALUATION CRITERIA

5.1. Mission Selection and Comparison

Three missions were analysed in order to evaluate the benefits of applying the reference architecture in the development of the corresponding end-to-end simulator. The selected missions for analysis were ExoMars, Eculid and Solar Orbiter.

5.2. Mission Comparison and SS mission type coverage

After collecting the full information about each of the three missions, a comparison was done, in terms of the SS Categories outlined in previous documents ([RD.4] and [RD.5]).

Table 5-1: Classification of Selected Missions in the most relevant SS Categories

Category	ExoMars TGO	Euclid	Solar orbiter (In-Situ / Remote Sensing)
Mission Type	Planetary with 9 months transit time	Astronomy	Solar with 2 years transit time
Instrument Type	**NOMAD**: Semiconductor Array (IR) and CCD (UV) **CaSSIS**: CCD **ACS**: Semiconductor Array with two different substrates (HgCdTe, PbCdSe) **FREND**: Scintillator + He-3 Counter	**VIS**: CCD **NISP**: CCD	**EPD**: Semiconductor Array **MAG**: Magnetometer **RPW**: Antenna **SWA**: Electrostatic Analyser **EUI**: APS **METIS**: APS **PHI**: APS **SoloHI**: APS **SPICE**: APS **STIX**: Collimator
Detector Type	**NOMAD**: 2x Imaging Spectrometer **ACS**: 2x Echelle Spectrometer + Dual Band Fourier Spectrometer **CaSSIS**: Imager **FREND**: Particle Detector	**VIS**: Imager **NISP**: Imager & Photometer	**EPD**: Particle Detector / Neutron **MAG**: Magnetometer **RPW**: Plasma/Radio Wave **SWA**: Plasma Analyser **EUI**: Imager **METIS**: Imaging Spectrometer /Coronagraph **PHI**: Imager & Polarimeter **SoloHI**: Imager **SPICE**: Imaging Spectrometer **STIX:** Imaging X-Ray Spectrometer

Waveband	**NOMAD**: IR (2.2-4.3 μm) and VIS/UV (0.2-0.65 μm) **ACS-NIR**: NIR (0.7-1.7 μm) **ACS-MIR**: MIR (2.3-4.6 μm) **ACS-TIR**: TIR and FIR (1.7-17 μm and 1.7-4 μm) **CaSSIS**: Four band filters: pan-chromatic (centred at 650 nm), blue-green (475 nm), IR (950 nm) and NIR (850 nm) **FREND**: Detector 1: ^{3}He counter for epithermal neutrons (0.4 eV-500 keV) Detector 2: stylbene scintillator crystal for fast neutrons (0.5-10 MeV)	**VIS**: VIS (0.55 μm to 0.9 μm) **NISP**: NIR 3 broad band filters (Y, J, H) on a wheel, covering the band from 1.0 to 2.0 μm 4 grisms on a wheel to read redshift data, which is in the range 0.7-2.0 μm	**EPD**: N/A **MAG**: N/A **RPW**: Radio **SWA**: Radio **EUI**: High Resolution Imagers (HRI): two sensors, one centred at Lyman-α and the other at 17.4 nm (extreme UV). Full-Sun Imager: a single sensor with two interchangeable filters, at 17.4 and 30.4 nm (extreme UV). **METIS**: broad-band (visible at 500-600 nm) and narrow-band (UV at 121.6 nm and EUV at 30.4 nm) **PHI**: Visibile **SoloHI**: Visible **SPICE**: Extreme UV 70.2-79.2 nm, 97.2-105.0 nm and 48.5-52.5 nm **STIX**: X-ray

From the table above, the Missions best suited for the application of the SS-E2ES Reference Architecture are **ExoMars TGO** and **Solar Orbiter Remote Sensing Instruments**. The selection of two contrasting mission types means that the Reference Architecture will be applied to quite different, but realistic, purposes. The selection is also the one that maximizes the range of Detector and Instrument types, covering all possibilites from in-situ particle detectors to X-ray spectrometers.

The two missions include:

- ❑ The possibility of extending the RA of the ExoMars TGO to include the Lander in future iterations of this study and the possibility to extend the Solar Orbiter RA to include the in-situ instruments
- ❑ A total 5 different Instrument Types
- ❑ A total of 9 Detector Types
- ❑ Measuring properties of atmosphere, surface and sub-surface planetary features, and of deep space.
- ❑ Detection of electromagnetic radiation (visible and infrared on both Solar Orbiter and ExoMars, additionally UV in the case of ExoMars's NOMAD instrument), particles (Fine Resolution Epithermal Neutron Detector), electric fields (MicroARES), magnetic fields (MAG), X-rays (STIX), plasma and radio waves (RPW and SWA), etc…
- ❑ Performing imaging, spectrometry, coronography and photometry
- ❑ Making measurements that depend on the spacecraft's position (ExoMars's Mars mapping), orbital position (Solar Orbiter observation modes), and more complicated geometry such as the relationship between the TGO, Mars and Sun (ACS solar occultation measurements)
- ❑ Single-instrument results and examples of data from multiple instruments being combined e.g. CaSSIS providing the geological and dynamical context for sources or sinks of trace gases detected by NOMAD and ACS, or combining several Solar Orbiter measurements from its 10 instruments
- ❑ Different limitations on the data, including spacecraft pointing stability, telescope focus, instrument signal-to-noise ratios

6. EXOMARS AND SOLAR ORBITER REQUIREMENTS SPECIFICATIONS

6.1. ExoMars High Level Concept

6.1.1. Simulator Context

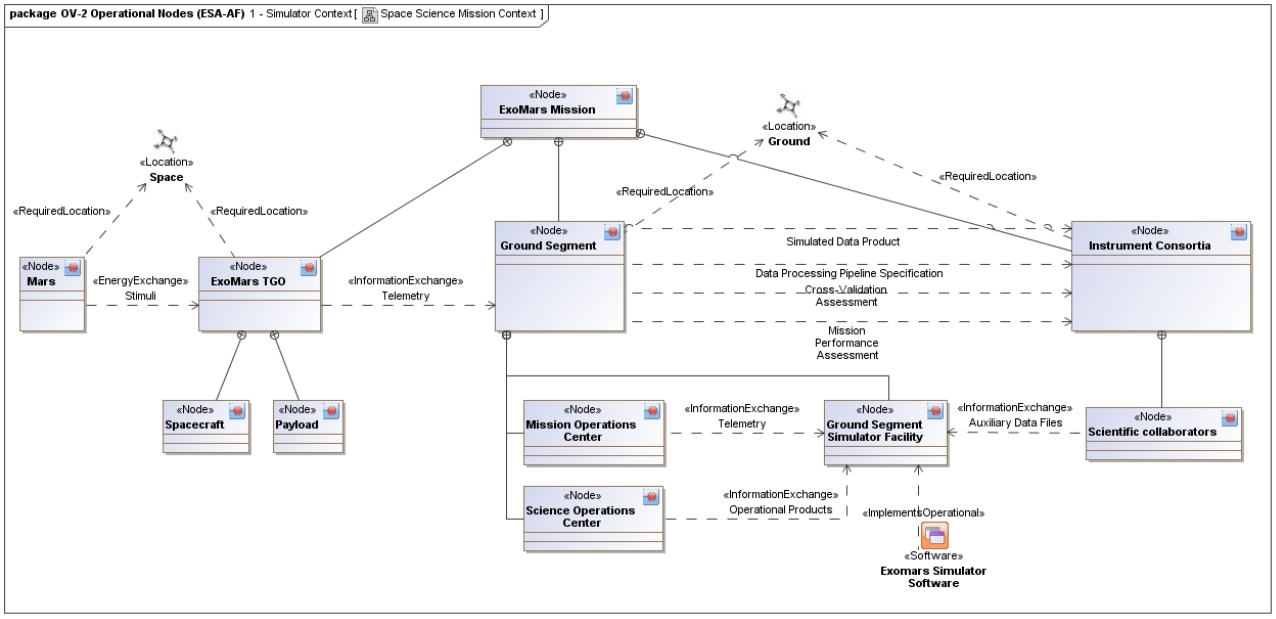

Figure 6-1: ExoMars E2E Simulator location in the overall mission context

The E2E Simulator will be included in the Ground Segment of the ExoMars mission and it will be operated mostly in ESA's Ground Segment facility. It is expected that a version will also be running at the Principal Instrument Teams (e.g. BIRA – the Belgium Institute for Space Aeronomy).

- The Space Segment is composed by the flying platform, ExoMars TGO, which is observing Mars with its instruments.
- The Ground Segment is where all the Teams and Processing centers are located and working.
- There are two main Processing Centers
  - Mission Operations Center (MOC), responsible for monitoring and commanding the platform, computing Flight Dynamic and receiving and packaging L0 data
  - The Science Operations Center (SOC), where the L0 data is processed into L2 and higher levels, archived and disseminated to the Users
- Typically placed as part of the SOC, the E2E Simulator Software produces processing model definitions, cross-validated assesments, simulated products and Mission Performance Assessments to in support to the Instrument Consortia Teams and Scientific Collaborators in general.

6.1.2. Stakeholders Definition

These are the relevant stakeholders, their concerns and objectives, including the products, which will be generated in order to support the development of the E2E simulator software:

- ESA, as sponsor of the Space Science Mission
- ROSCOSMOS, as producer of one of the instruments
- ESTEC, as "subsidiary" of ESA, acting as "customer" of the SS-E2ES project and all future E2ES projects.
- ExoMars E2ES industrial consortium, supporting ESTEC in
    - The development of the the Architecture of of the E2ES, using the requirements laid out in this document
    - The implementation of the E2E based on the Architecture
    - Testing, Verification and Validation of the E2ES
    - Maintenance and support of the E2ES throughout the lifetime of the mission
    - Using the Software Framework, external CFI proposed by ESA, as specified in the "SFG", "SFI", "SFC" and "SFD" requirements.
- ExoMars mission teams, who will be using E2E simulator software. The main users of this software would be mission scientists, the instrument project manager and their teams and the quality control engineers.

All these stakeholders needs and expectations have to be taken into account when defining the requirements and care must be taken to make sure that all are covered by the final Requirements List.

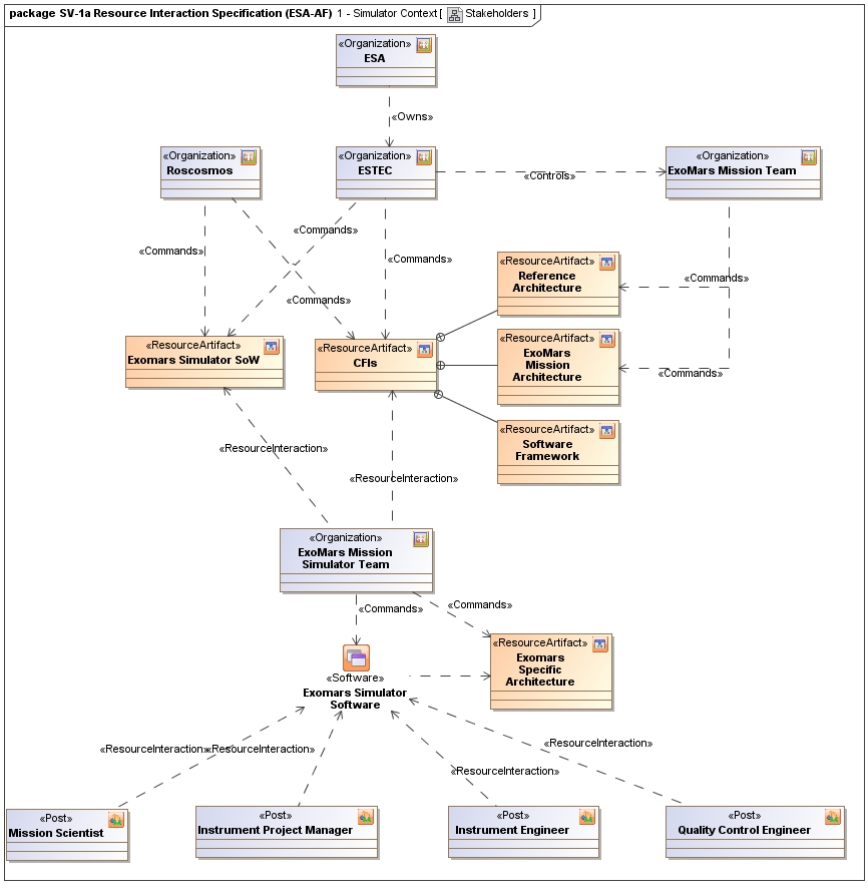

Figure 6-2: : Main ExoMars mission stakeholders

6.2. Solar Orbiter High Level Concept

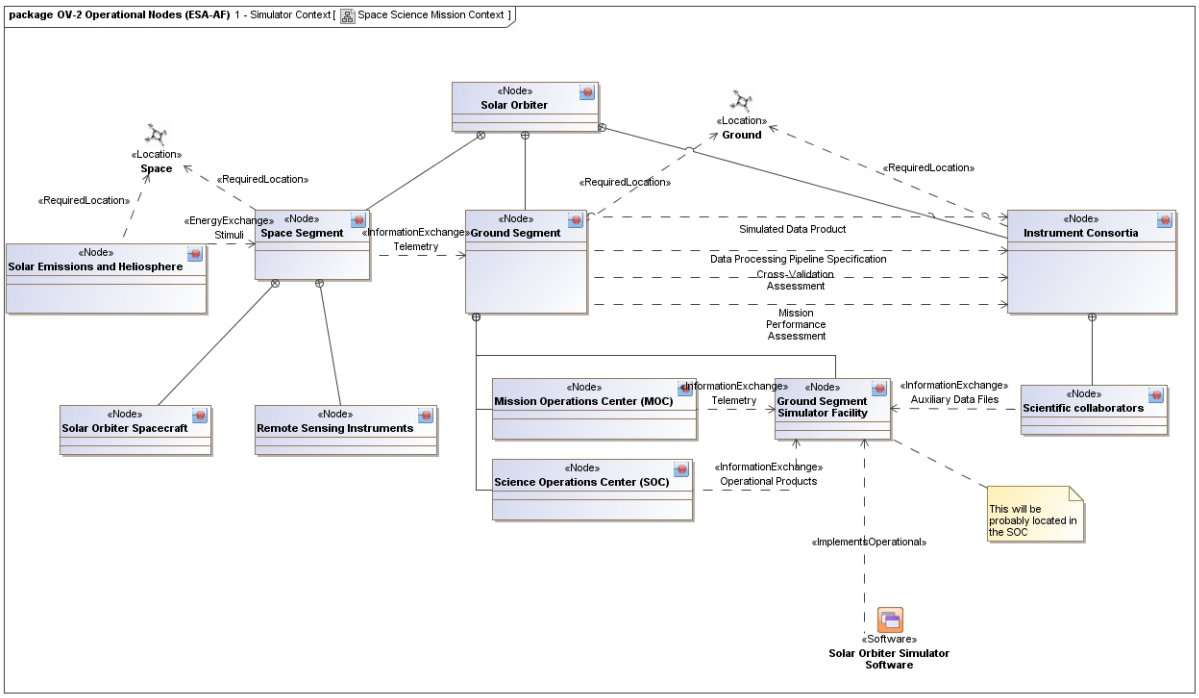

Figure 6-3: Solar Orbiter E2E Simulator location in the overall mission context

The E2E Simulator will be included in the Ground Segment of the Solar Orbiter mission and it will be operated mostly in ESA's Ground Segment facility. It is expected that a version will also be running at the Principal Instrument Teams (e.g. the Max-Planck-Institut für Sonnensystemforschung).

- The Space Segment is composed by the flying platform, Solar Orbiter TGO, which is observing Mars with its instruments.
- The Ground Segment is where all the Teams and Processing centers are located and working.
- There are two main Processing Centers
  - Mission Operations Center (MOC), responsible for monitoring and commanding the platform, computing Flight Dynamic and receiving and packaging L0 data
  - The Science Operations Center (SOC), where the L0 data is processed into L2 and higher levels, archived and disseminated to the Users
- Typically placed as part of the SOC, the E2E Simulator Software produces processing model definitions, cross-validated assessments, simulated products and Mission Performance Assessments to in support to the Instrument Consortia Teams and Scientific Collaborators in general.

6.2.1. Stakeholders Definition

These are the relevant stakeholders, their concerns and objectives, including the products which will be generated in order to support the development of the E2E simulator software:

- ESA, as sponsor of the Space Science Mission
- ESTEC, as "subsidiary" of ESA, acting as "customer" of the SS-E2ES project and all future E2ES projects.
- Solar Orbiter E2ES industrial consortium, supporting ESTEC in

- The development of the the Architecture of of the E2ES, using the requirements laid out in this document
- The implementation of the E2E based on the Architecture
- Testing, Verification and Validation of the E2ES
- Maintenance and support of the E2ES throughout the lifetime of the mission
- Using the Software Framework, external CFI proposed by ESA, as specified in the "SFG", "SFI", "SFC" and "SFD" requirements.

- Solar Orbiter mission teams, who will be using E2E simulator software. The main users of this software would solar scientists, the project architect and his team and the instrument, calibration and validation scientists.

All these stakeholders needs and expectations have to be taken into account when defining the requirements and care must be taken to make sure that all are covered by the final Requirements List.

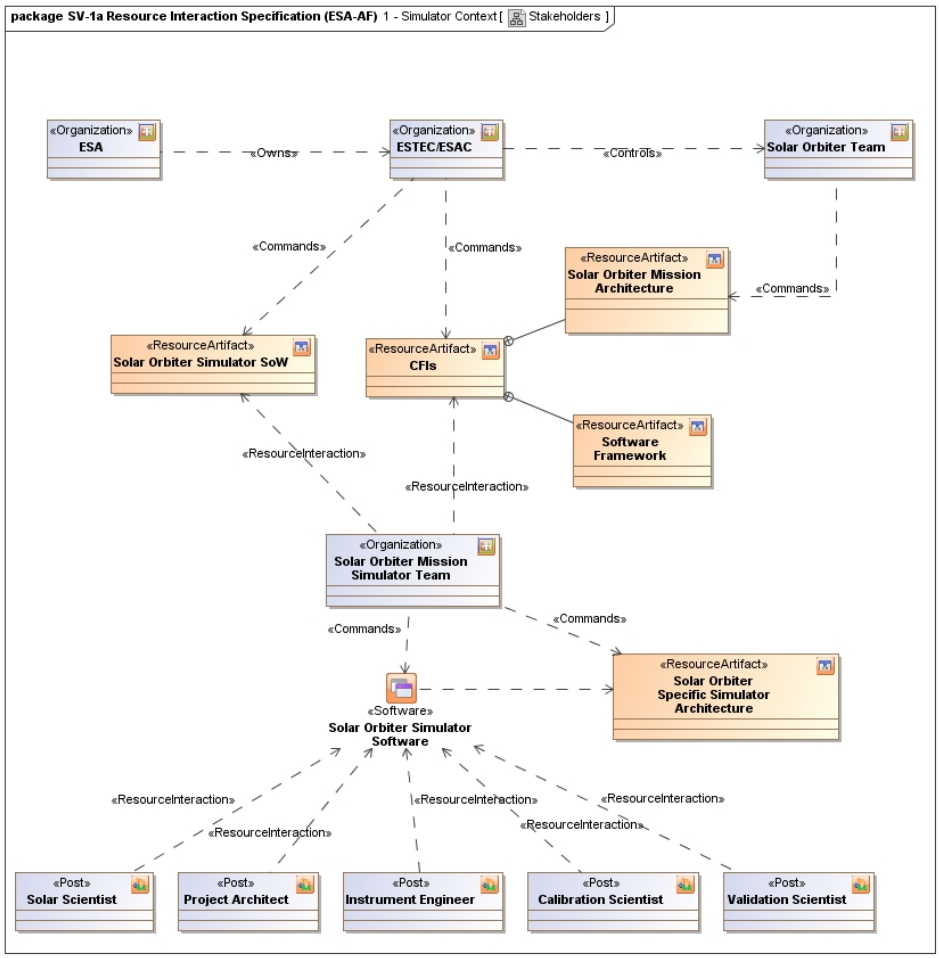

Figure 6-4: Main Solar Orbiter mission stakeholders

6.3. ExoMars and Solar Orbiter E2E Simulator Software Requirements

For the full set of requirements defined for each mission, please refer to [RD.7] and [RD.9].

7. EXOMARS AND SOLAR ORBITER ARCHITECTURE DESIGN

For each of the instruments of the mission, the Building Blocks were described for the processing of the data from L0 to L2. According to the Product Levels defined Section 2, the Data Processing part of the ExoMars E2E simulator will have the following processing levels:

- **Level 0.5 Processing - From Level 0 to Level 0.5 data**: Will process Level 0 data with into engineering units. Often this process will be reversible, but may not be, at least not without loss of accuracy.
- **Level 1 Processing - From Level 0.5 to Level 1 data**: Will process telemetry and science data into calibrated data, in physical units. Output of the Level-1 Processing modules is calibrated to remove instrumental effects. Satelilte pointing and orbital status is also part of the output of this Processing.
- **Level 2 Processing - From Level 1 to Level 2 data**: Will process L1 data into L2 Science data. Output of the Level-2 retrieval modules, containing fully calibrated and scientifically useful data products with derived physical parameters, e.g. images, spectra, spectral cubes.

The blocks defined above will have to be applied in the definition of the pipeline for each of the ExoMars TGO instruments. Although the instruments' data will be used to perform some cross-related activities (such as computing co-alignment after arrival in Mars orbit and after separation) none of these activities will be part of the nominal Processing Pipelines, and as such, no crossing of the data is shown in the next diagram. Please note how different channels of the same overall instruments are processed separately. In NOMAD processing, for example, the LNO and SO sub-instruments are based on Semiconductor Array detectors, the UVIS, a much smaller instrument has a CCD as sensing layer. For ACS, the MIR and NIR channels are standard spectrometers while the TIR is a Fourier Transform Spectrometer, that needs a separate processing pipeline.

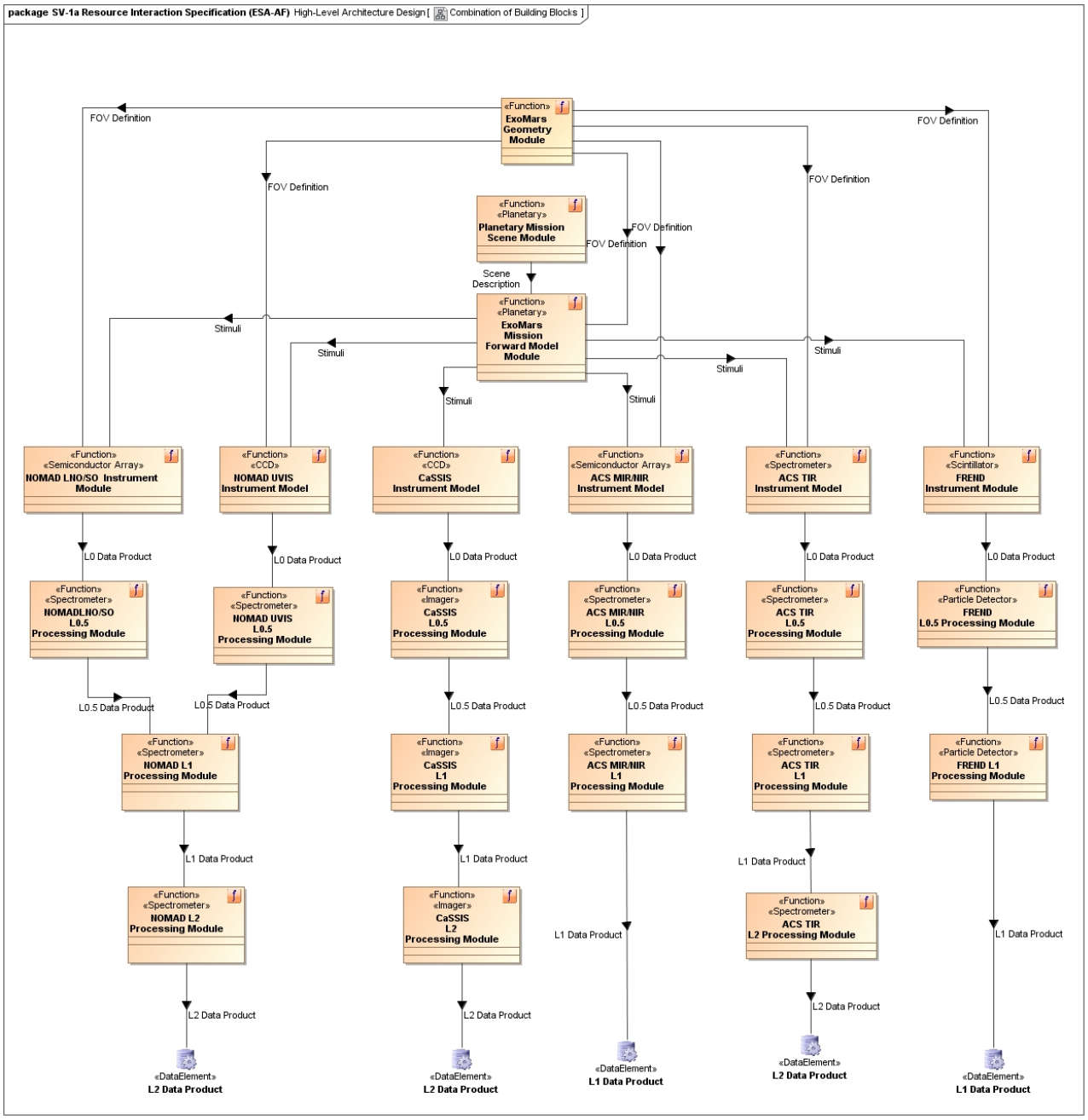

Figure 7-1: Detailed Pipeline definition for all ExoMars TGO instruments

7.1. Solar Orbiter Architecture Design Specification

The Processing Levels defined for the ExoMars mission are directly applicable to the processing of the Solar Orbiter Mission instruments.

Note that in this definition, the Geometry Module is common to all the instruments – as well as the Scene Module and the Forward Model Module. While the first generalization is certainly advisable (all the instruments share the same platform, after all), the generalization of the physical simulations is a bit less warranted since there might be models only applicable to some of the instruments and which are interesting to keep in separate modules. This decision can be revisited at a later time as more information about the Scene Simulation algorithms for the Solar Orbiter mission becomes available.

Note that not all of the instruments have a L2 Processing Module, as defined in their specific pipeline design documents, available at the time of writing this ADD.

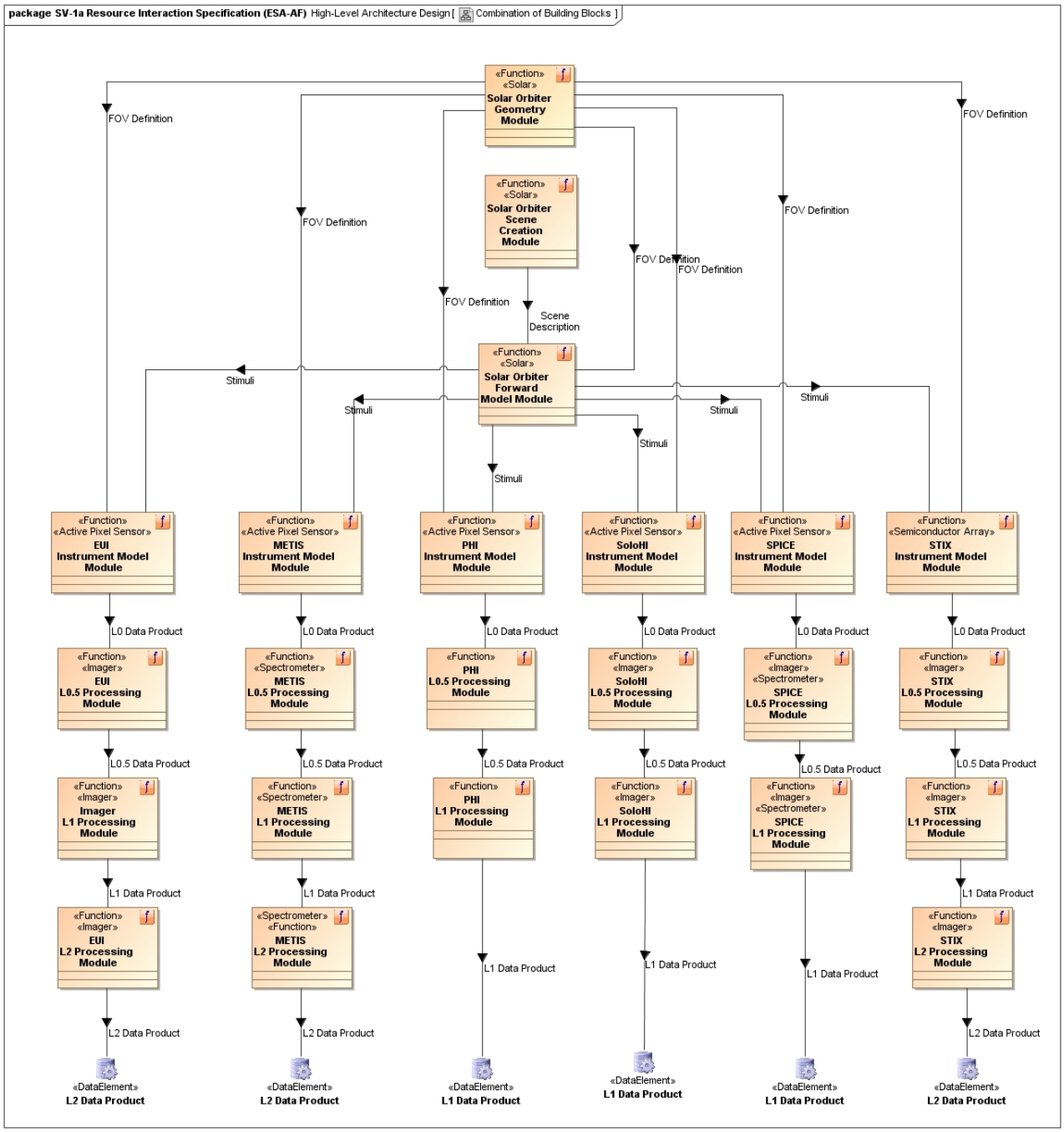

Figure 7-2: Detailed Pipeline definition for all Solar Orbiter instruments

8. RA APPLICATION EVALUATION AND ROADMAP

8.1. Evaluation exercise definition

The advantages of the RA were assessed in the scope of this contract. In order to have a qualitative estimation of the advantage (and, when possible, quantitative), evaluation criteria were defined in [RD.8] and agreed with ESA.

The evaluation criteria covered the following different aspects:

- Efficiency of the design process.
- Potential reuse of building blocks between different simulator of SS missions, and from EO missions.
- Comparison of the proposed development concept with previous developed simulators.
- Efficiency of development and validation process.
- Quality of results.

Estimates were made comparing the effort to develop an end-to-end simulator on an ad-hoc basis with the effort that would be needed to perform the same task once the RA is available.

8.2. Assessment of Reference Architecture Application

We found that there are several advantages gained by a Reference Architecture, principally in standardisation and reuse.

- Standardisation of terminology. Different missions and instruments may use different terms for the same thing, or even worse, the same term for different things. It is hoped that providing a reference architecture will help to promote standard terminology.
- Standardisation of requirements. There are a set of requirements that will be applicable to all mission simulators. The reference architecture will reduce the effort to identify them and help to avoid missing important ones.
- Standardisation of design. The same fundamental design can be applied to all missions. Software architectural design is difficult and a simulator, at least in the early stages, may well be implemented by scientists rather than professional software engineers. A solid and proven design will be of great benefit here. Moreover, skills acquired will be transferrable to other missions as the design remains familiar.
- Standardisation of interfaces. The interfaces between simulation stages would be defined by the RA. The format and structure of the exchanged files would be also provided, meaning no time would be needed to design them.
- Standardisation of implementations. Some modules may have a ready-made implementation and be ready to use by appropriately setting their configuration parameters to tailor their behavior to the mission. Some modules may not be implemented in the operational RA, but having their design ready, plus some building blocks available for reuse providing part of its functionality, and supporting libraries that significantly ease the implementation, would greatly reduce the effort to implement them. While it must be understood that there is always likely to be some tailoring needed for mission specifics, reuse of standard building blocks and libraries has great potential to stimulate productivity and significantly reduce the cost of development.

All of these advantages were very apparent when applying the Reference Architecture to the two selected missions.

For blocks with high degree of re-usability, like the Geometry Module or the L0 to L0.5 Processing Module, the RA was essentially untouched in the conversion to both specific architectures. Of course, for each of the instruments and detectors the implementation of the code itself will vary. But the objective of

the architecture was to make sure that such specific development is done inside a standardized set of Modules with standardized interfaces and names.

For some other blocks, like Scene Creation and Level 2 processing, it is much more difficult to generalize and our approach was to leave the description of such blocks at a very high level. Nevertheless, their inclusion on the overall architecture and the strict definition of their objectives and roles, together with the interfaces, makes it possible to assign easily the work for future implementation teams.

Overall, in our opinion, and from the very generic evaluation in the above table, we estimate that the existence of a Reference Architecture might have savings of up to 50% in overall effort. Savings will increase with as more Specific Architectures are done based on it, providing feedback, more Building Block specifications and even source code to a eventual central repository at ESA.

8.3. Roadmap Proposals

This section describes further developments for this activity that have been identified.

8.3.1. In-Situ Instruments

The statement of work for this study explicitly stated that it should concentrate on remote sensing missions. Although certain missions such as Solar Orbiter include both remote sensing and in-situ instruments in their payload, we have considered only remote sensing instruments in our categorization, commonality, and subsequent reference architecture exercises. However we have noted in our documents where we have identified differences in these missions and instruments, for example:

- Use of the CDF data format by the solar physics in-situ and solar terrestrial physics communities.
- Formation flying of the four Cluster spacecraft.
- Planetary and minor solar-system body rovers and landers.

We recommend that the reference architecture be reviewed and extended as necessary to additionally cover in-situ measurements. In fact, this is required for complete coverage of our chosen evaluation missions.

- The science goals of Solar Orbiter require combinations of data from both remote-sensing and in-situ instruments.
- The ExoMars rover can then be considered in addition to the orbiter.

We do not anticipate that this extension will invalidate the defined reference architecture in any way, and expect that differences will be relatively minor.

8.3.2. Harmonisation of SS and EO architectures

This study is a follow-up of the ARCHEO report [RD.1] to define a reference architecture for end-to-end simulators in Earth Observation missions. In the main it has proven possible to apply the same overall architecture to space science missions. However, some differences have emerged, such as the decoupling of the physics of the scene from the light arriving at the instrument by means of the forward model module. Given the overall similarities, it seems worthwhile to harmonise the two architectures into a single one. The EO reference architecture could be included in the ESA-AF model developed for this contract.

8.3.3. Survey of Libraries

This study defines building blocks in terms of architectural elements. Ideally the user would be presented with actual building block implementations to be chained together and configured for a particular mission. This can be considered the ultimate goal, but is likely to be a challenging one.

In the course of the study we identified a preliminary set of libraries as candidates for use in building block implementations. However, to identify the best candidates in each case is a large piece of work. Several factors would add complexity to this exercise.

- There are a lot of libraries in use by the various space science communities and it is a major effort to look at them all in detail. Involvement and buy-in of the science communities would be a significant help.

- Algorithms have to be sufficiently generic to be applicable to categories of missions and instruments rather than individual ones. A number of them are likely to be almost but not quite generic and it would need to be looked at whether it is possible, practical and worthwhile to rework them to be fully generic.
- Some algorithms will have several implementations and the "best" will need to be identified based on various criteria such as interfaces. performance, portability, scope for parallelisation etc.
- Some building blocks have alternative implementation algorithms and recommendations should be made regarding in which circumstances a particular algorithm is suitable or not.
- It is likely that in some cases no suitable candidate will exist. In these cases an estimated cost to develop it from scratch would be useful.
- Programming language considerations. See below.

This library survey could be a very useful thing in itself, even in the absence of a set of reference building block implementations. No single resource currently exists to aid the simulator or pipeline developer to find existing code or algorithms that they might be able to reuse. The existence of such a resource (web site, code repository) in itself could encourage the community to produce reusable algorithms to be made available there.

8.3.4. Programming Languages in Space Science

To some extent the architectural design is insulated from the programming language of the implementations due to using files containing data products as interfaces between modules. This is based on the requirements for the simulation framework. There is nothing to stop files being used as interfaces at sub-modular levels, and in many cases it will make a lot of sense. However, it seems likely that at least for performance reasons, and especially with lower level building blocks, that calls will have to made using APIs rather than file interfaces. In this case the programming language does matter and several implementations may be considered based on the choice of programming language.

Moreover, different languages may dominate in different communities, and many scientists are not easily persuadable to use something they are unfamiliar with. The Solar community uses IDL almost exclusively. IDL is also used extensively in astronomy, but there Python is growing in popularity, and Java is widely used in tools and APIs. The great majority of flight dynamics software, which could be useful for the geometry module, is still in FORTRAN.

To some extent it is also possible to define APIs independently of the underlying language, for example to call C or FORTRAN routines from IDL or Python, but this is probably tricky in the general case and may have portability issues.

8.3.5. Data Products

8.3.5.1. Formats

The categorization exercise identified only two data product formats in use by space science remote sensing communities: PDS for planetary missions and FITS for everything else. To widen the scope of a generic end-to-end simulator architecture, additional formats could be supported:

- CDF for solar in-situ and solar terrestrial physics.
- netCDF for Earth observation.

Other data formats, such as HDF, do exist, and while we identified no cases in our survey, it would be prudent to consider that they could be needed at a future date.

8.3.5.2. Data model

It may be beneficial to define a generic data product model that is independent of any specific format. Then the support for a particular format becomes an import/export operation. For example the Herschel Space Observatory ground segment defined a Java-based model and API for this reason.

The design of such of a model must however still make some consideration for the actual supported formats, since some are quite hierarchical in nature and others such as FITS are somewhat "flatter". It should be noted that the Herschel model was designed to easily map to a FITS format.

The first step in any such data product modelling should be to analyse whether the benefit outweighs the cost when compared against the alternatives.

8.3.6. Tools and Frameworks

8.3.6.1. Reference Architecture User Tool

While creating the RA and the website and while reviewing its contents, the team discussed possible expansions and improvements to the RA as a tool for users.

The model could be dynamic and able to provide a generic RA according to user inputs. A simple use case would run something like this:

- Enter mission name, type, orbit type, attitude type etc.
- Add instrument
- Enter instrument name, waveband, type, detector type

We saw this as a web-based application, although a standalone GUI application is also possible. The set of forms proposed initially as part of this study could potentially be re-used as templates for this application.

This could be a nice modelling tool in itself, but could also go further into the realm of Model-Based Engineering by generate scaffolding code out of the Information Systems Architecture (data and applications) to a target language of choice (C++, Java, Python, etc.).

8.3.6.2. Automated Test Framework

The standardization of module design and interfaces may well result in the possibility of a generic framework to support automated integration testing. In principle all end-to-end simulators will run the same modules in the same sequence. This could be run either as part of continuous integration system, or at set periods e.g. nightly if it takes too long for each build.

8.3.6.3. Categorization Database

As part of the categorization and commonalities exercise for space science missions, a database was created so that different views of the data could be dynamically extracted. Finding this information for some missions and instruments proved to be surprisingly difficult. If this database was expanded and maintained, then this information may prove useful for the general space science community. This could be achieved by an online front-end for querying the database.

8.3.7. Use the architecture for a real project

The fundamental test of whether a reference architecture is correct and useful in a practical sense is to apply it to a real-world project. Here the goal is to go beyond a design and produce an actual working simulator. In the first place this probably should be for a relatively small and non-complex mission. This would likely result in further refinement of the architecture.

9. CONCLUSIONS

Over the course of this study we have convinced ourselves of the benefits of a reference architecture in particular and of an end-to-end simulator in general. Our arguments have mostly been technical ones, but the reality is that it can only be sold to the space science community by successfully making the case that it improves the science return of the mission. This can be achieved by:

- Optimising scientific performance.
- Saving time and money on pipeline development and testing.

We believe that the reference architecture can help to deliver both.

There is no doubt that end-to-end simulators are useful for space science missions in phase 0/A, and this is where the benefits of a reusable architecture and building blocks are clearest. However, a problem remains in securing adequate funding for it. The reference architecture provides a standard design with a solid software engineering base, thus reducing costs and is of great benefit in itself. Additional gains are achieved simply through standardization of terminology.

Furthermore, and as demonstrated, the availability of a reference architecture will save significant costs in the definition, detailed design and implementation of an end-to-end simulator. The eventual provision of a standard set of reusable building block implementations is more ambitious, but if it can be achieved it will surely go a long way towards making such simulators even more affordable throughout the space science community.

End of Document