estec

European Space Research
and Technology Centre
Keplerlaan 1
2201 AZ Noordwijk
The Netherlands
T +31 (0)71 565 6565
F +31 (0)71 565 6040
www.esa.int

DOCUMENT

Generic E2ES (E2E Performance Simulator) and L1/L2 Processor Requirement Document and Inputs to SoW

Prepared by	EOP-PEP team
Reference	PE-TN-ESA-GS-402
Issue	2
Revision	0
Date of Issue	16-12-2022
Status	Approved/Applicable
Document Type	Requirement Specification
Distribution	

APPROVAL

Title Generic E2ES (E2E Performance Simulator) and L1/L2 Processor Requirement Document and Inputs to SoW	
Issue 2	Revision 0
Author	Date 16-12-2022
EOP-PEP team	
Checked by	Date 16-12-2022
M. Zundo	
Approved by	Date 16-12-2022
M. Zundo	

CHANGE LOG

Reason for change	Issue	Revision	Date
First issue	1	0	02-10-2014
Maintenance update following release of openSF 3.5	1	1	14-06-2016
Revision of text incorporating developments in E2E	1	2	17-07-2018
Revision of requirements incorporating developments in E2E	1	3	29-11-2018
Revision of all requirements	2	0	30-11-2022

CHANGE RECORD

Issue 1	Revision 0		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
First issue	02-10-2014	all	all

Issue 1	Revision 1		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Release of openSF	14-06-2016	all	all
Clarification on active instrument	14-06-2016	all	all

Issue 1	Revision 1		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Update of baseline sw tools	14-06-2016	all	all
Issue 1	**Revision** 2		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Incorporating E2E developments since last revision into text	17-07-2018	all	all
Issue 1	**Revision** 3		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Review and revision of requirements based on new E2E developments and removal of duplicates	01-05-2020	all	all
Alignment of req. for L1PP and L2PP	01-05-2020	all	sec. 6.6 and 6.7
Clean-up and update of Acronyms and Definition	01-05-2020	all	sec. 1.1 and 1.2
Clarified interfaces between SGM, ISM and GM	01-05-2020	all	sec. 3.2.2
Issue 2	**Revision** 0		
Reason for change	**Date**	**Pages**	**Paragraph(s)**
Review of all requirements based on internal discussions	16-12-2022	all	all

This document is a generic template at the level of an ECSS Technical Specification addressing system and software aspects (NOT algorithms) as well as inputs to the corresponding Statement of Work.

It must to be **reviewed** and **customised** for each mission paying attention to the text in Yellow

Please change <MISSION-X> to your mission.

1 SCOPE

This document defines the requirements and deliverables for the <MISSION-X> End-to-End Performance Simulator (E2ES) applicable to the <MISSION-X> industrial team during the <MISSION-X> Project Phases 0, A, B, C, D, and during support to Phase E1.

These requirements address both user and implementation aspects to the level of ECSS-E-ST-40C SSS (Software System Specification), IRD (Interface Requirement Document) and TS (Technical Requirement Specification) and inform the SOW (Statement Of Work), including identifying the inputs, outputs, and deliverables associated with procurement.

The nomenclature used herein is specific to the E2ES and differs slightly from the ECSS documentation. However, in case of difference the correspondence is given.

1.1 Acronyms

The acronyms below are used within the scope of this activity and within this document.

AD	Applicable Document
ADD	Architecture Design Document
ANC	Ancillary Data
AOCS	Attitude and Orbital Control System
APID	Application Process ID (CCSDS)
AR	Acceptance Review
ATBD	Algorithm Theoretical Baseline Document
BOA	Bottom Of the Atmosphere
BRK	Breakpoint
AUX	Auxiliary Data
CAL	Calibration Data
CCDB	Characterisation and Calibration Data Base

CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
CKD	Calibration Key Data
DDF	Design Definition File
DDVP	Design, Development and Validation Plan
DJF	Design Justification File
DPM	Detailed Processing Model document
DU	Data Unit
E2ES	End-to-End mission performance Simulator
EOCFI	Earth Observation CFI libraries
EM	External Module
FTP	File Transfer Protocol
GM	Geometry Module
GNSS	Global Navigation Satellite System
GPP	Ground Processing Prototype (*defined for completeness, use L1PP or L2PP instead*)
GS	Ground Segment
HKTM	Housekeeping Telemetry
HMI	Human Machine Interface
(I)CCDB	(Instrument) Characterisation and Calibration Data Base
ICD	Interface Control Document
I/F	Interface
IMA	Item Made Available
IODS	Input/Output Data Specification
IOCR	In-Orbit Commissioning Review
IPF	Instrument Processor Facility
IRF	Instrument Response Function
ISM	Instrument Simulation Module
ISP	Instrument Source Packets (formatted as CCSDS Space Packets)[1]
KD	Key Data
KDA	Key Data Archive
KOM	Kick Off Meeting
L0P	Level 0 Processor
L1	Level 1 product
L1 E2ES	End-to-End Simulator up to Level 1
L1OP	L1 Operational Processor
L1PP	L1 Processing Prototype
L2	Level 2 product

[1] The term "*Source*" in the definition of ISP is due to legacy reason and does not have any semantic. The ISP acronym identifies the instrument generated data units.

L2 E2ES	End-to-End Simulator up to L2
L2OP	L2 Operational Processor
L2PP	L2 Prototype Processor
L3	Level 3 product
LGPL	GNU Lesser General Public License
MAG	Mission Advisory Group
MRD	Mission Requirements Document
MSP	Maintenance and Support Plan
MSR	Maintenance and Support Report
NC	Non-Compliance
NTP	Network Time Protocol
ODGM	On-board Data Generation Module
openSF	Simulation Framework
PAM	Performance Assessment Module
PAM L1/PAM1	Performance Assessment at Level 1
PAM L2/PAM2	Performance Assessment at Level 2
PDD	Product Definition Document
PDGS	Payload Data Ground Segment
PDR	Preliminary Design Review
PGM	Level-1 Product Generation Module
PKP	Performance Key Point
PVT	Position Velocity Time
RAW	RAW Data
RB	Requirements Baseline
RD	Reference Document
(S)CCDB	(Satellite) Characterisation and Calibration Data Base
SGM	Scene Generation Module
SOW	Statement Of Work
SP	(CCSDS) Space Packet
SRF	SW Reuse File
SRN	SW Release Notes
SS	Space Segment
SVVP	System Verification and Validation Plan
SUM	Software User Manual
TDS	Test Data Set
TN	Technical Note
TOA	Top of Atmosphere
TS	Technical Specification
V&V	Verification and Validation

1.2 Definitions

This section contains a set of definitions and resources relevant to the understanding of the requirements and of the process described in this document. It is not a comprehensive list of terminology used in the E2E domain.

1.2.1 *Data Definition*

Observation data (OBS)	Data Units output of the Instrument and formatted as CCSDS Space Packets exactly as generated by the Instrument and containing the measurements both in normal and calibration modes.
Ancillary data (ANC)	CCSDS Space Packet Data generated on-board in support of the observation data, both by the instrument and the platform, such as, navigation, temperature, Housekeeping Telemetry (HKTM), timing data and configuration. When generated by the instrument they are called "*instrument ancillary*" when by the platform "*platform ancillary*".
Raw data (RAW)	Sequence of concatenated Instrument and Ancillary Space Packets as are transmitted on the space to ground RF link with no header and no annotation.
Level 0 product	Level 0 data files in the same format at the actual GS (Ground Segment header + concatenation of CCSDS Space Packets).
Level 1 product	Level 1 data files in the same format at the actual GS and as generated by L1PP. (*Note: Product content is expected to follow [CEOSHB]*).
Level 2 product	Level 2 data files in the same format at the actual GS and as generated by L2PP. (*Note: Product content is expected to follow [CEOSHB]*).
Auxiliary data (AUX)	Data needed, by the ground segment, to perform ground processing and not part of the measurement data set. This auxiliary data (static or dynamic) is in format of files formatted as in the real GS to be used for configuration of the processor or as input to the processors (e.g. DEM, Land classification map, RTM

	lookup table, Orbit Files, Instrument Characterisation, Meteorological data, Offset tables, calibration coefficient, focal plane definition, etc.). Some auxiliary data can originate from offline calibration activities (see definition of calibration products). Within the E2S they are supplied as part of the simulation scenario as an input e.g. to the scene generator module SGM, to the instrument simulation module ISM and to the Level 1 and Level 2 Processor Prototypes.
Calibration Products (CAL)	Data files (products) generated in the ground segment or by the L1PP during the processing of instrument data and used in the Ground Segment or in the L1 and higher-level processing. They can be either dynamic (CAL) or static (CAL/AUX). • Dynamic/on-line calibration data (CAL) are produced and applied automatically by ground processors based on measurements or calibration data. • Static calibration data (can be called either CAL or AUX) are produced off-line based on measurement, long term trending, analysis, manual setting etc. and applied to the processing chain as configuration items (e.g., dead pixel, instrument alignment, etc.).
Breakpoint (BRK)	Data files optionally produced by L1 and L2 Processing modules containing any intermediate result useful for diagnostic, debug and troubleshooting.
NOMINAL Position and Attitude Data	The PVT and Quaternion corresponding to the mission orbit and attitude.
REAL Position and Attitude Data	The PVT and Quaternion corresponding to the physical orbit ad attitude of the platform. It is obtained from the NOMINAL one plus errors due to its actual implementation (e.g. orbit control, AOCS performance).
MEASURED Position and Attitude Data	The PVT and Quaternion as measured on-board via GNSS and AOCS. It is the on-board and on-ground knowledge of position, velocity and orientation of the satellite and is obtained from the REAL attitude plus AOCS measurement error.

RECONSTRUCTED Position and Attitude Data	The PVT and Quaternion corresponding to the Mission orbit and attitude as computed on-ground making use of the MEASURED one plus any other optional correction.

1.2.2 *Other Definitions*

Term	Definition
ATBD	In context of Processor Prototype development, the ATBD is the document describing the algorithms to be implemented. It addresses the overall processing flow within the processor, all the steps required, all functions and all parameters used. The ATBD describes for each function systematically: the input, the output and the mathematical algorithm to be used, and any special processing.
IODS	In context of E2ES and L1/L2 Processor Prototype development the IODS works as a complete and coherent list of all the input and output data (files) and their content (OBS, AUX, CAL, etc) addressing therefore both the interface aspects (ICD-like) as well as the content and format ones of each file and allows algorithm described in ATBD to refer to both parameters as well as physical realisation of data (files). It can refer to other documents as necessary to avoid duplications (e.g. Packet format specification within Level 0). The specification of the Operational Data Products and their detailed formats comes later in the development and will be based on the IODS but considering also operational aspects (e.g. metadata and or product splitting for optimisation). Since the L1 output Products are the input to the L2 Processor the common aspects of the two IODS needs to be coordinated.
SCCDB/ICCDB/CCDB/CKD/KDA	These acronyms identify, in different missions, the set of data, parameters and coefficients related to design or calibration (or both) that are needed to process the on-board measurements. These data can be obtained in full or in part as result of the design process, of on-ground measurements, of on-ground calibrations and in-flight calibrations (either relevant for on-line or off-line calibration processing). In the GS these data are assigned to data types CAL or AUX (see 3.3.1) .

Term	Definition
On-line and off-line calibration	See section 3.3.1.
Dynamic and static calibration	See section 3.3.1.
Module	Each of the software components that perform a defined function in the context of a E2ES chain. Within the context of the architecture defined by the [E2EGICD] they are constituted by software executables with file-based input and output.
Orchestration	Refers to the process external to the module of invoking at the right time, in the right order and with the right inputs the modules/executables to produce the required output. *Orchestration* can be limited to the one-off selection of AUX or CAL data which are valid for a specific observation or to the sequential invocation of processing and simulation modules and build-up of all required inputs according to a scenario that changes with time.
Job-order	In context of Data Processor orchestration, a *job-order* is a file listing all the necessary inputs (data and configuration) needed for a specific processing run.
Time-based scenario	In the context of E2E simulation as described in [E2EGICD] a *time-based scenario* (supported by a corresponding time-based orchestration) is a time-ordered list of time segments for which different simulation conditions applies. These conditions can either be different operating mode of the instrument as well as different type of targets. The orchestration will execute/simulate the time-segment in sequence according to their order.

1.2.3 *Resources*

Term	Definition
openSF	Orchestrating framework SW compliant to the E2E Generic Interface ICD and freely available as executable and source code at opensf.esa.int according to ESA Community licence Type 3.
ESA Community License - Permissive	Permissive Software License applicable to ESA Member States (http://opensf.esa.int/index.php/docs-and-mission-data/licensing-documents).

2 DOCUMENTS

2.1 Applicable documents

The documents applicable to the required <MISSION-X> activities and deliverables are formally defined in the contract. For ease of reference the contractual appendices/applicable documents are repeated herewith. In case of conflicting, incomplete, missing or ambiguous requirements the contractor shall bring these to the attention of the customer for formal resolution.

AD	Title	Reference	Issue
[SOW]	Statement of Work for the <MISSION-X> (SOW)		
[SSRD]	<MISSION-X> Satellite System Requirement Document (SSRD)		
[MRD]	<MISSION-X> Mission Requirements Document (MRD)		
[ECSST]	Tailoring of ECSS Standards for <MISSION-X>		
[DRL]	<MISSION-X> Document Requirements List (DRL)		
[EOFFSTD]	Earth Observation File Format Standard and associated <MISSION-X> tailoring	PE-TN-ESA-GS-0001	latest
[HXML]	Handbook for EO XML and Binary Schemas	PE-TN-ESA-GS-0121	latest
[RWLO]	EO generic RAW and L0 specification	PE-TN-ESA-GS-586	latest
[CFIFS]	Earth Observation Mission Software File Format Specification	PE-ID-ESA-GS-584	latest
[OPENSF]	OpenSF Documentation at http://eop-cfi.esa.int/index.php/opensf		latest
[E2EGICD]	ESA generic E2E simulator Interface Control Document	PE-ID-ESA-GS-0464	latest
[PDEF]	<MISSION-X> Product Definition		
[S2GICD]	Space to Ground ICD		
[L1IODS]	Level 1 Input/Output Data Specification		
[L2IODS]	Level 2 Input/Output Data Specification		
[L1ATBD]	Level 1 Algorithm Theoretical Baseline Definition		
[L2ATBD]	Level 2 Algorithm Theoretical Baseline Definition		
[OFLCATBD]	Offline Calibration Algorithm Theoretical Baseline Definition		
[ESACLP]	ESA Community License - Permissive: http://eop-cfi.esa.int/index.php/docs-and-mission-data/licensing-documents		

2.2 Normative documents

ID	Title	Reference	Published
[ECSS40]	Space Engineering Software	ECSS-E-ST-40C	latest applicable
[ECSS80]	Space Product Assurance	ECSS-Q-ST-80C	latest applicable

2.3 Reference documents

RD	Title	Reference	Issue
[CFIUM]	Earth Observation Mission Software CFI UM	http://eop-cfi.esa.int	latest
[AGILE]	ECSS 40 Agile Development Handbook	ECSS-E-HB-40-01A	latest applicable
RD03	<MISSION-X> Technical Specification		
RD04	<MISSION-X> Interfaces Control Document (ICD)		
RD05	<MISSION-X> Architecture Design Document (ADD)		
RD06	<MISSION-X> System Validation Plan (SVP)		
RD07	<MISSION-X> Mission Performance Analysis Plan (MPAP)		
[CEOSHB]	CEOS Handbook on Product definition: http://ceos.org/document_management/Working_Groups/WGISS/Documents/WGISS_CEOS-Interoperability-Handbook_Feb2008.pdf	CEOS Interoperability Handbook	1.1

3 CONTEXT AND PURPOSE OF THE END-TO-END PERFORMANCE SIMULATOR AND LEVEL 1 PROCESSORS

3.1 The E2E performance Simulator

The E2ES is a complete *end-to-end* chain of software modules representing both the Space Segment and the Ground Segment of the mission and *generically* built as shown in Fig. 3-1.

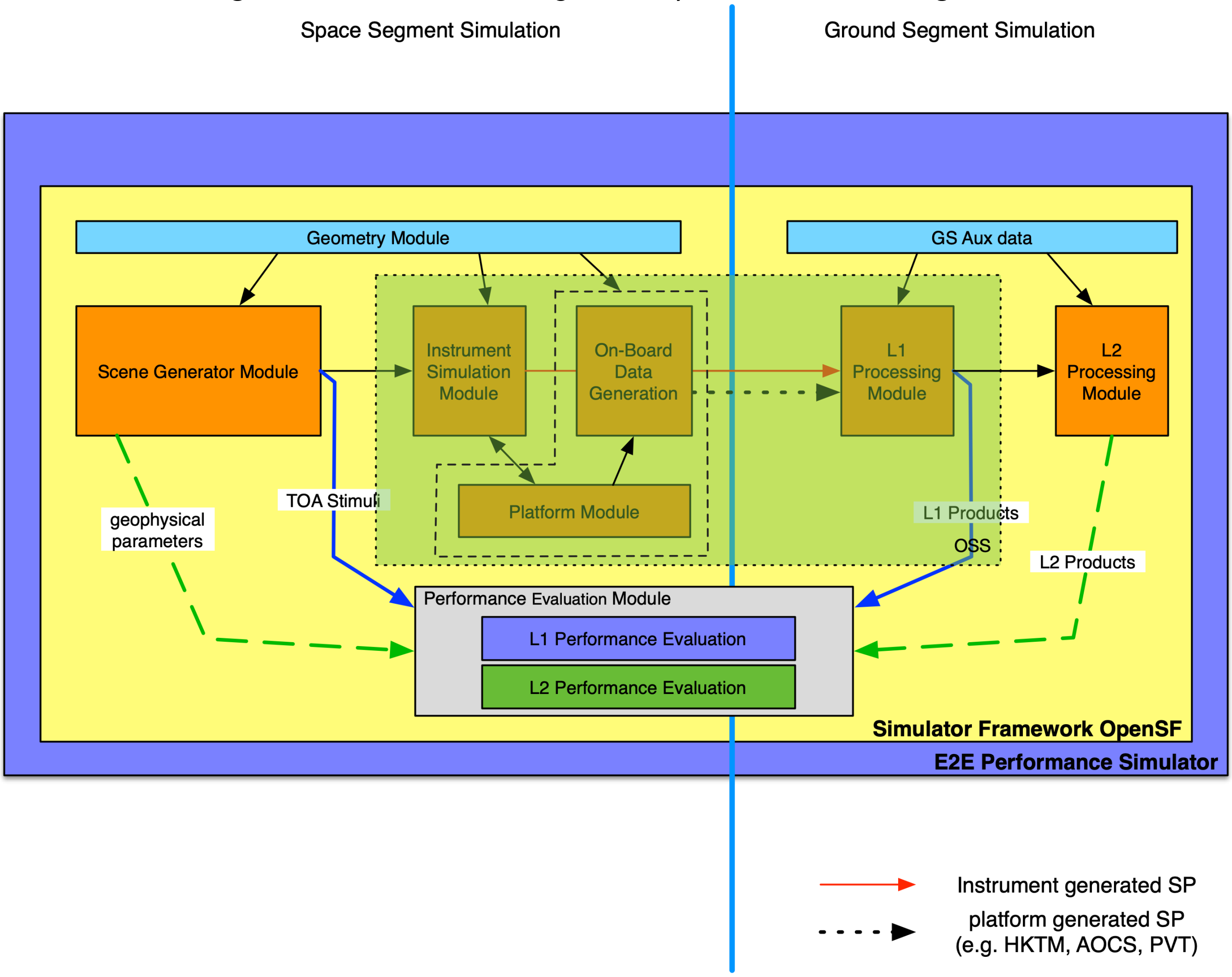

Fig. 3-1 E2E Performance Simulator (E2ES)

This chain allows simulating the complete process and flow from a simulated scene (the reference scene) to the L1 and L2 data products. Comparison of the output with the input after introducing to the process noise, measurement errors, different instrument models, and different L1 and L2 algorithms characterizes the system performance and its sensitivity to these variable parameters.

Comparison of the retrieved geophysical quantities in the L2 products with the geophysical parameters which were input to the Scene Generator characterizes the full end-to-end system and may be termed the L2 E2ES. If the comparison is between the L1 data products and the output of the Scene Generator (stimuli), the name L1 E2ES is appropriate.

In the early phases of a mission, the E2ES supports the definition and the verification of the Space Segment requirements; in later phases it is used as an offline Test Data Generator for the Ground Segment and as prototype for the ground processing but also to support the assessment of the mission objectives at L1 and L2.

To ensure that the E2ES is fully representative it is mandatory that there is no direct interface or data sharing between the Space Segment simulation and the Ground Segment simulation and that any data exchange is performed via simulated TM files and static AUX data.

The E2ES will be made available to various scientific actors supporting <MISSION-X>, in order to obtain contributions for further improvement of the E2ES.

The L1PP and L2PP are software tools that evolve and serve multiple purposes over the mission project phases because they will be used:

a) to support the E2E assessment up to L2 of the mission performance pre-launch
b) to support the development and verification of the Space Segment
c) to support development of the Ground Segment
d) to support commissioning and exploitation

The typical evolution of these items in the different phases of the project is shown in Fig 3-2.

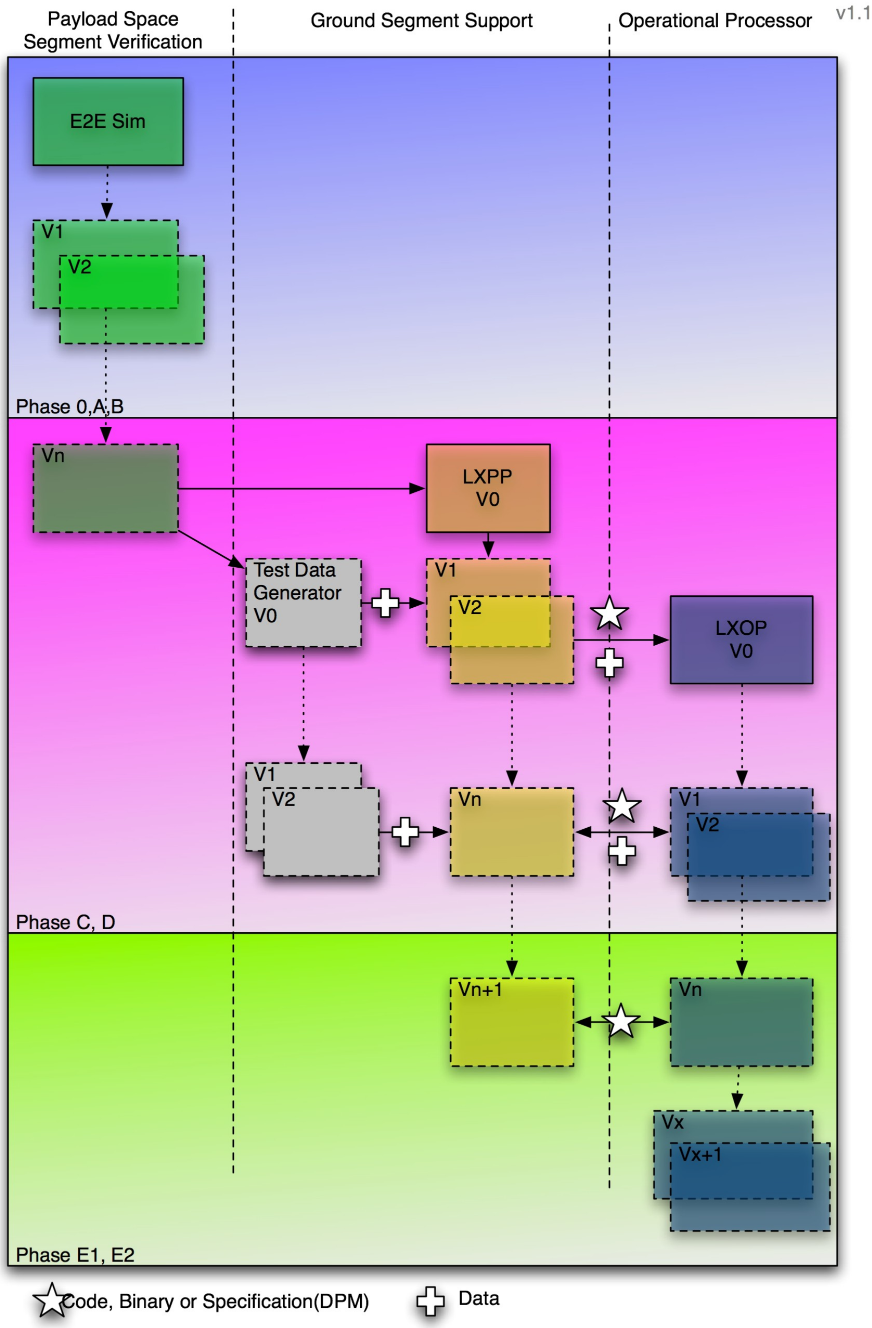

Fig. 3-2: Typical evolution of the E2ES and L1PP

3.1.1 *Simulation Infrastructure*

The E2ES development requires the use of a simulation infrastructure, providing basic functionalities for the setup and execution of the simulation (Fig. 3-1). OpenSF is a generic simulation framework product developed, maintained and used internally by ESA, which can be used for this purpose and is available free of charge.

The OpenSF framework provides added-value functionalities to scientific simulations, substituting complex and rigid scripts for the execution of user-defined models. Its architecture is laid out in the Fig. 3-3. With OpenSF, engineering models and product exploitation tools can be plugged in the same system platform using a well-defined integration process. Sensitivity analysis and automatic error iteration on modules parameters is included off-the -shelf with minimal req. on the modules. Libraries exist to ease integration of models developed in C/C++, Fortran 90, IDL and MATLAB. The software comes complete with databases, a help system, and a complete set of user documents.

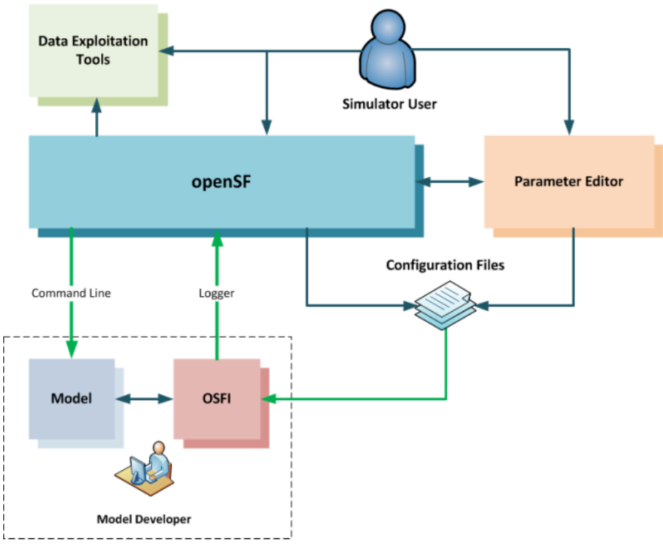

Figure 3-3 OpenSF high-level architecture

License-wise, OpenSF uses a flexible licensing scheme that allows integrating it in any kind of third-party developments. The core library is distributed under the ESA Permissive Community License Type 3 (BSD like), [ESACLP].

OpenSF is a multi-platform system with minimal hardware and software requirements. The technologies used are: Java(TM) 2 Runtime Environment, and XML. It is supported on Windows XP, Linux and Mac OS X operating systems.

OpenSF provides both a GUI and a command line interface for interacting with the E2ES. The OpenSF documentation and software can be downloaded (after user registration) from the link [https://eop-cfi.esa.int/index.php/opensf].

3.2 The Space Segment Simulation

3.2.1 *The Geometry Module*

The Geometry Module (GM) computes all information related to the observation geometry, e.g., Orbit Propagation (PVT), Attitude Determination (Quaternions), Field of View, and Coverage areas, grids needed for re-sampling and provides them to the other modules of the space segment simulations. Its calculation makes use of the Earth Observation Mission Software CFI library [CFIUM].

The GM implements the orbit propagation and attitude calculation either using an internal model or by ingestion of externally generated Orbit and Attitude data (e.g., NOMINAL, REAL or externally simulated) providing an abstraction layer between the actual source of data and the rest of the modules.

Any necessary geometrical calculations, e.g., Sun/star position, satellite position, instrument pointing, line of sight, occultation, eclipses, visibility limits needed, for example, to select the visible stimuli or to compute the impact of Sun illumination on detectors, thermal variation impacting or deforming the instrument geometry, are computed with functionality provided via the common GM.

The reference data flow used between the GM and the other modules for the NOMINAL, REAL, MEASURED and RECONSTRUCTED position and attitude is shown in Fig. 3-4 below. (See the definitions in Section 1.2).

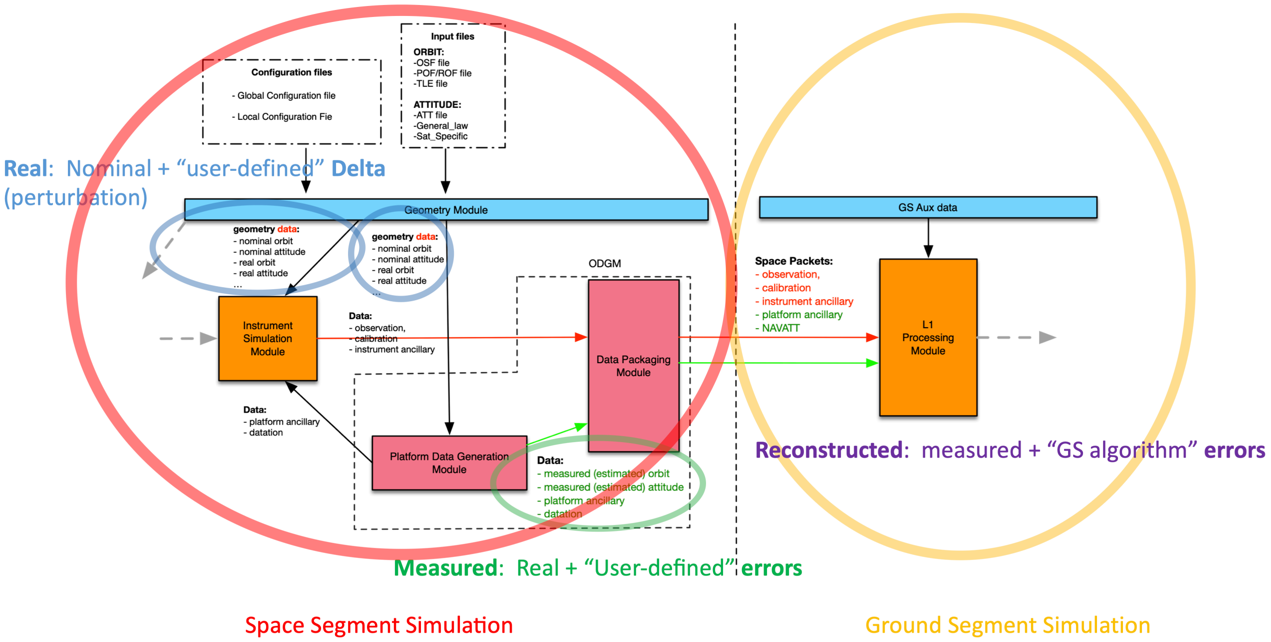

Fig. 3-4 Reference Position and Attitude data flow

3.2.2 *The Scene Generator*

The *Scene Generator* Module (SGM) performs two functions:

1. Ingest the geophysical model of the scene to be observed, for example temperature, sea roughness, dielectric constant, wind speed, altitude, salinity, CO2 concentration, aerosol size, etc on a ground-referenced grid independent of the observation geometry. In Phase B/C, the SGM will also include where required a model for the simulation of the stimuli used during on-ground characterisation as well as of other in-flight complex calibration targets (e.g., Transponders, Celestial bodies like the Moon, the Sun or the Deep sky, Surface targets like Dome C).

2. Implement the forward model for the observed scene: generate the stimuli at TOA to the instrument at any one time (for example radiance) or (for active instruments) the relevant backscattering matrices in a generic and adequately oversampled grid.[2]

---

[2] The ideal architecture (Fig. 3-5) within the E2ES entails the TOA stimuli scene being computed, to the maximum extent, on a grid independent from the position and attitude of the observing system and leaving to the ISM the task to sample it at the appropriate resolution for the relevant line of sight, thus facilitating the run of the simulation in different geometry conditions without the need of re-running every time computationally expensive SGM simulations, however for processing performance reason this is not always possible.

These functions make use of auxiliary geophysical information, and use information computed by the GM, to produce the (set of) time-based instrument scenes which will be the input to the Instrument Simulation Module (ISM).

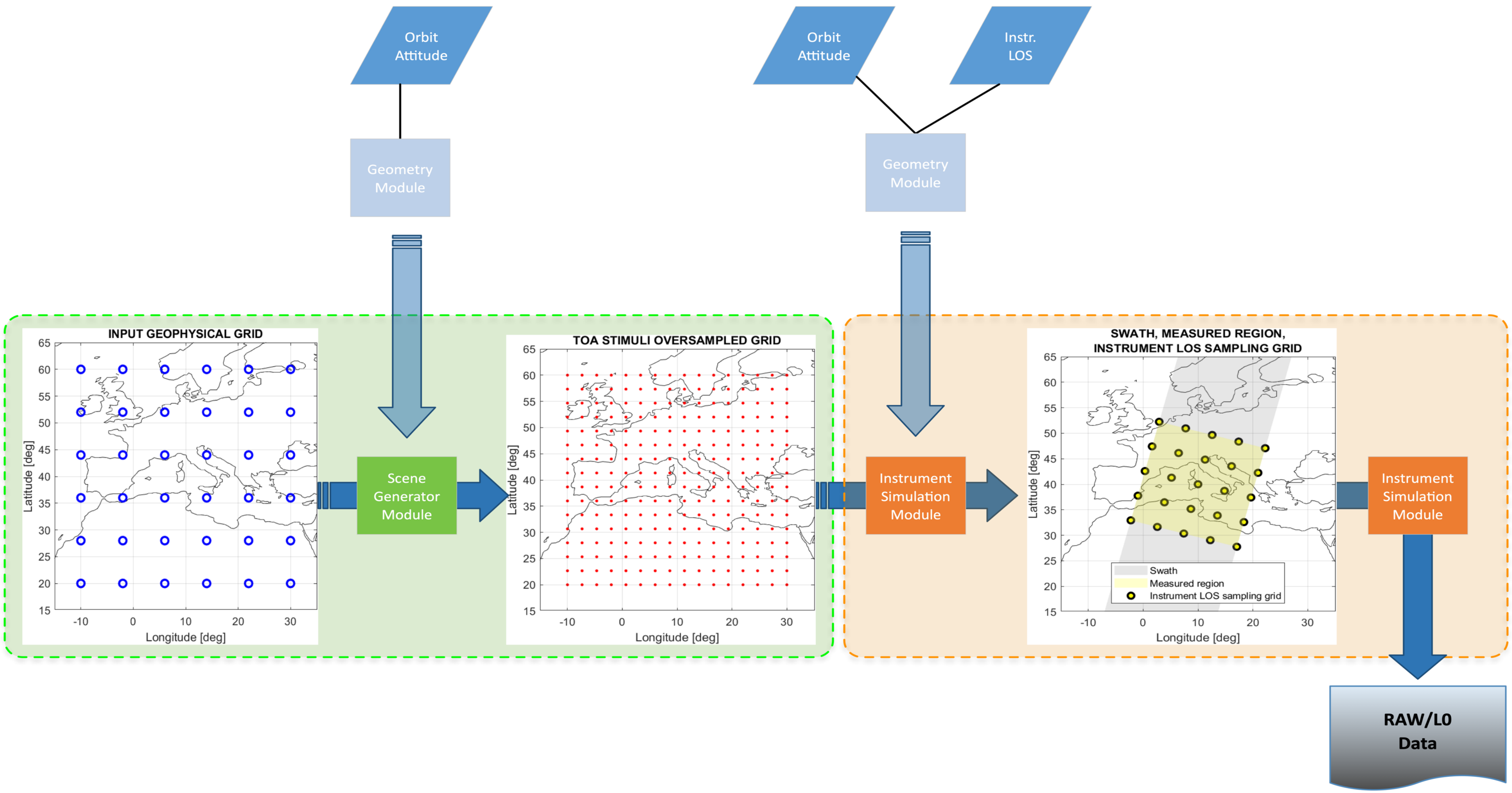

Fig. 3-5 Reference architecture of SGM interfaces

The SGM can also include an appropriate modelling of the atmosphere, and in the case where user-defined errors can be introduced (understood as additional environmental effects, for example clouds or a modulation of the geophysical target) these shall also be computed and applied in the scene generator to the generated TOA stimuli.

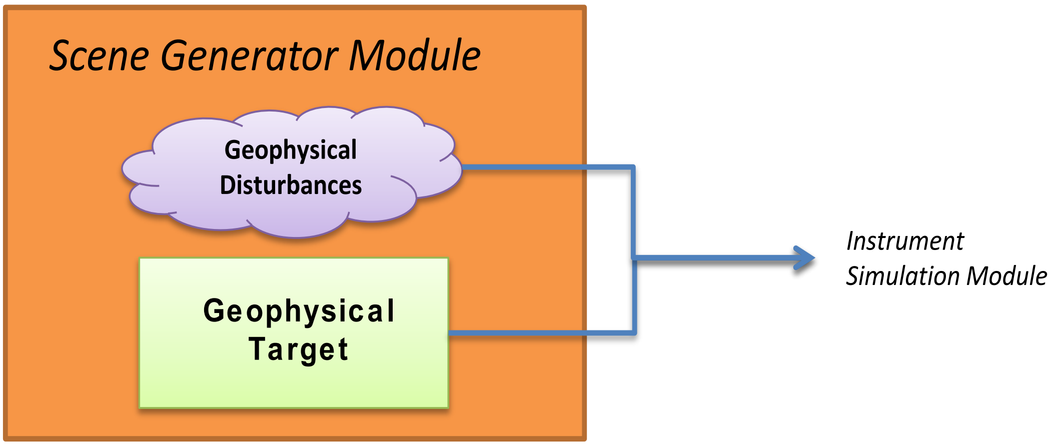

Fig. 3-6 Scene generator

Additionally, there are geophysical disturbances representing effects external to the instrument that modify the computed stimuli. For example, an active instrument such as Synthetic Aperture Radar can be affected by the Ionosphere while targeting the Geophysical ground backscattering; a Lidar instrument measuring wind speed can be affected by Earth motion while a passive spectrophotometer affected by atmosphere's absorption.

Any necessary geometrical calculations, e.g. Sun/star position, satellite position, line of sight, spatial visibility ranges or directions in which the scene has to be generated, are computed by the common Geometry Module.

3.2.3 *The Instrument Simulation*

The Instrument Simulation Module (ISM) computes the transfer function of the instrument by implementing a model of the instrument with the required detail for the relevant observing conditions. The transfer function, applied to the input scene computed by the SGM, samples the TOA (or the external calibration target) stimuli scene in the appropriate resolution and geometry, produces output representing the measurement and ancillary instrument data as they are generated on-board the satellite.

The instrument modelling makes use of constructive, design, engineering, or pre-flight calibrated values, which need to be used also during the data processing; these data are contained in dedicated AUX and CAL data files.

Additionally, any user-defined errors (e.g., Gaussian measurement noise, biases, drifts, vibrations, harmonic oscillation, thermoelasticity, mis-pointing, etc.) are injected in this module to simulate their effect on the measurement and produce the resulting on-board generated data.

Any necessary geometrical calculations, e.g., Sun/star position, satellite position, instrument pointing, line of sight, occultation, eclipses, visibility limits needed, for example, to select the visible stimuli or to compute the impact of Sun illumination on detectors, thermal variation impacting or deforming the instrument geometry, are computed by the common Geometry Module.

To maintain this dataflow concept in the more complex case of an active instrument it is envisaged that the input stimuli scene represents an interaction between the target and the signal transmitted by the instrument. In this case the Instrument Simulation Module computes the convolution of the transmitted signal with the (passive) scene created by the Scene Generator Module and applies the geophysical disturbances. The modified signal is then detected in the receiver and instrument packets are generated. The information flow of an active instrument simulator is presented in Fig. 3-7.

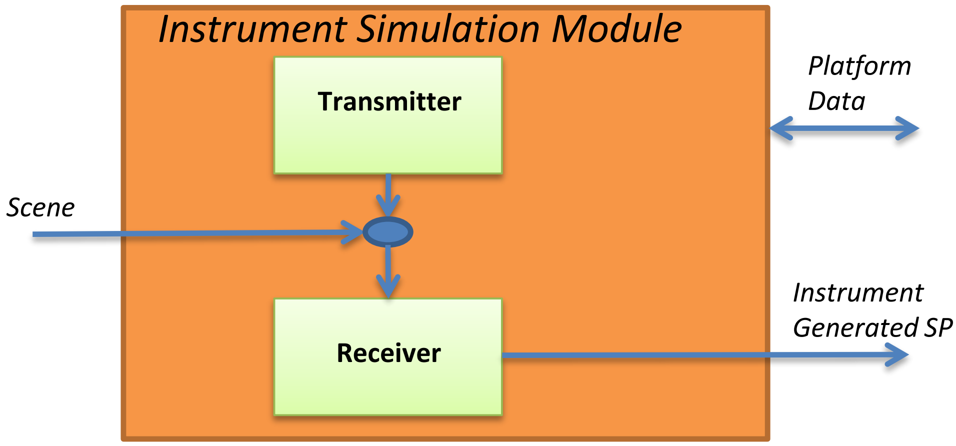

Fig. 3-7 ISM (example for an active instrument)

3.2.4 *The Platform and On-Board data Generation*

The Platform and On-Board data Generation (ODGM) Module implements the simulation of all platform functions needed directly by the instrument model as well as the ones needed to generate the data in format compliant with the space to ground interface (e.g., CCSDS Space Packet data) (i.e., RAW) e.g., by simulating GNSS, AOCS, temperatures, timing, and data formatting functions.

This function will also generate any HKTM source packet (platform ANC) that is produced on-board and that is needed on-ground to perform the processing when the parameters are not included directly in the instrument-generated telemetry. Standard functions that implement the generation of platform Ancillary Navigation CCSDS packets are available from ESA and can be used for this purpose (https://eop-cfi.esa.int/index.php/applications/tools/navigator).

3.3 The Ground Segment Simulation

For reference a view of the dataflow within a generic Earth Observation ground segment is shown here in Fig. 3-8.

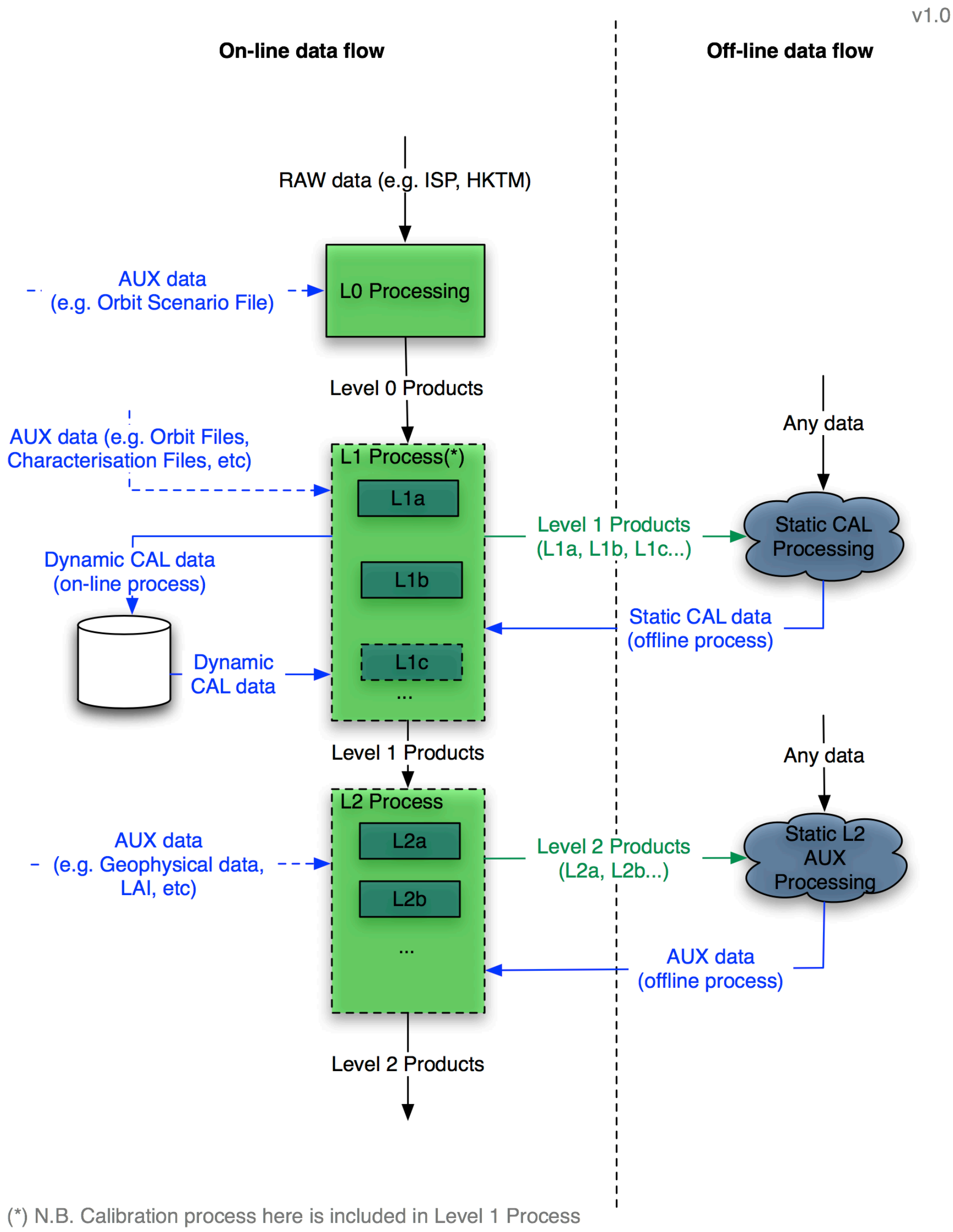

Fig. 3-8 Generic view of dataflow in an Earth Observation data processing Ground Segment

3.3.1 *The Level 1 Processor Prototype*

In the context of the E2ES chain the L1 Processing Module contains the L1 Prototype Processor (L1PP) which processes RAW data into Level 0 data into Level 1 and Calibration data (therefore also including the dynamic calibration function). In doing so it ingests RAW data, Level-0, Auxiliary, and Calibration data. It is worth noting that while in the operational GS a dedicated L0 Processor is developed, in the context of the E2ES the associated function is implemented in a very simplified way within the L1PP to allow RAW data to be used as interface between Space and Ground segment however the functionality to ingest also data formatted as Level-0 is maintained. The generic functionality of Level 1 processing is indicated in Fig 3-9.

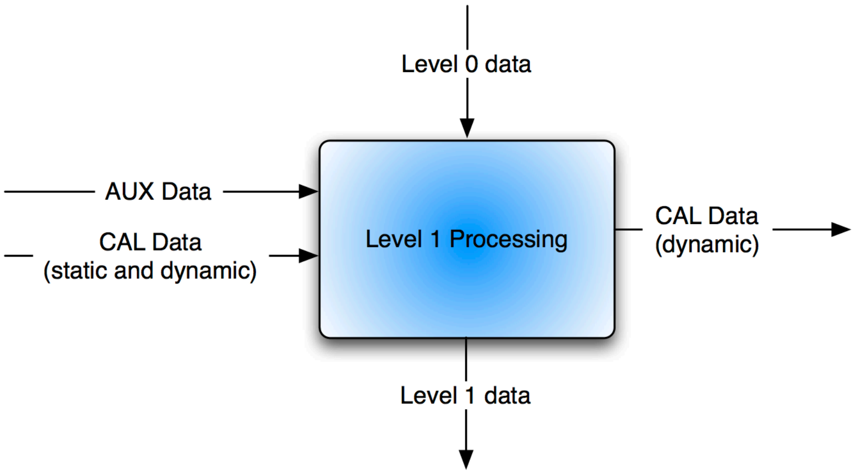

Fig. 3-9 Generic Level 1 Processing function

<u>Nomenclature Note:</u>

- ***dynamic CAL*** data (also called *on-line* calibration) are automatically produced by the L1PP during the processing of data from nominal or calibration modes of the instrument and used automatically in the processing chain (either internally or externally).

- ***static CAL*** data (also called *off-line* calibration) are data produced by the L1PP processor (also via dedicated CAL processors) and which are not automatically used in the processing chain. Static CAL parameters are typically derived either from dedicated calibration modes, or from nominal data spanning multiple files. Both these cases have in common that the parameter values cannot be derived and applied instantly to the same dataset from which they are derived (in contrast to online calibration). Use of static CAL data in the processing chain is implemented only following human review/intervention/authorisation (e.g., a permanent update of a table of sensor gain parameters following some ad-hoc observation of a calibration target). From the point of view of data flow the static CAL data are equivalent to AUX data or can be used as such.

NB: It is common practice that the instrument characterisation quantities either designed or calibrated on ground (e.g., gains, conversion factors, offsets, mis-pointing, biases, etc) are described in the so-called **CCDB** or **CKD** files. Within the context of the data processing Ground Segment and E2ES chain, these baseline files should be categorised as AUX files due to their "static" nature and treated as such with respect to the data flow and selection rules (e.g., versioning and validity), without any special handling.

It is also good practice to use the CAL category for any parameter computed in-flight via an on-line (dynamic) CAL process so that the baseline value (contained in AUX) is cleanly separated from the ones computed in flight while reserving updates to the AUX files only to the output of offline/static CAL process (e.g. permanent misalignment matrices, long term biases, pixel masks).

The L1PP is developed with the following objectives:

a. to serve as an element in the chain of the E2ES
b. to serve as the baseline and reference for the development of the L1 Operational processor (e.g., production of DPMs and TDSs) and to support commissioning as stand-alone processor.
c. optionally to be transformed into an L1 Operational Processor SW to be integrated into an Operational Ground Segment processor facility (as a separate SW product).

While the L1PP usually evolves as the algorithm baseline for the Ground Segment, there is also the need to support Ground Segment development by developing an early L1PP mock-up with I/F stubs compatible with the Ground Segment interfaces prior to inclusion of actual algorithms. Subsequently these components must converge, and the same software module is to be usable for both purposes.

Maintaining the same L1PP interfaces allows for the direct re-use of the L1PP module part of the E2ES and, later, to feed back any improvement in the L1PP stand-alone into the E2ES. This will ensure coherency in Performance Assessment between Space Segment and Ground Segment.

The L1PP detailed processing models will be used as the specification of the algorithms of the operational processors. Therefore, the L1PP will be developed strictly considering the operational data flow concepts and formats.

The L1PP will then be used for the following purposes:

a. L1 performance simulation with simulated data
b. instrument and algorithm performance assessment on data from AIV/AIT on-ground pre-launch measurements

c. algorithm evolution and improvement
d. instrument calibration using data from AIV/AIT on-ground pre-launch calibration
e. instrument and algorithm performance assessment using in-flight data (in particular, during commissioning phase)
f. monitoring (e.g., instrument performance aspects, and supporting the formulation of the Phase E2 monitoring baseline)
g. data analyses
h. generating L1 test data set for the L2 community

The L1PP together with the rest of the E2ES chain will be made available to various scientific actors supporting <MISSION-X> to allow them to perform independent processing with user defined configuration.

3.3.2 *The Level 2 Processor Prototype*

The *L2 Processing Module* consist of the *L2 Prototype Processor* (L2PP) which implements the processing algorithms to go from Level 1 data to Level 2 and which involves retrieval of geophysical parameters. As a component of the E2ES chain it allows the performance evaluation of the overall mission data which is in general defined at Level 2. In doing so it ingests Level-1 and Auxiliary data and outputs Level-2 products.

In early phases (A/B1 as well as B2) this module is generally referred to as the *L2 Retrieval Module* (L2RM) and contains the L2 retrieval algorithms resulting from science studies.

NB: *since the definition of the Level-2 products normally comes at a late stage in the mission lifetime, the L2PP included in the E2ES might initially provide geophysical data not formatted as a Level-2 Data product*.

The L2PP is developed with the following objectives:
a. to become an element in the chain of the E2ES with compatible interfaces
b. to serve as the baseline and reference for the development of the L2 Operational processor (e.g., production of DPMs) and to support commissioning standalone
c. optionally to be transformed into an L2 Operational Processor module to be integrated into the Operational Ground Segment
d. Test bed for algorithm development for the L2 scientific community
e. L2 performance assessment with realistically simulated L1 data

3.4 Performance Assessment Module (PAM)

The Performance Assessment Module (PAM) is a collection of SW and graphical functions to evaluates pre-launch the quality of the L1 and L2 products within the context of the E2ES mission performance activities: their adherence to performance requirements as well as the suitability of the relevant processing algorithms and sensitivity analysis by comparing the input to the simulation chain at geophysical scene with the Level 2 product and the input to the instrument simulation (ISM) as TOA stimuli with the Level 1 product. The performance assessment is made possible by producing statistics, graphs, and plots, by providing a visualisation aid, and allowing error characterisation and verification of error propagation. The PAM functions and algorithms can later be implemented or re-used in dedicated automated monitoring and offline calibration facilities within the operational ground segment.

The PAM functions can be grouped according to:

- Level-1 output products representing calibrated measurements are compared to the simulated impinging stimuli computed by the SGM (e.g., Earth radiance vs. TOA spectra in which case radiometric, spectral, and geometric quantities). The comparison can extend to derived quantities such as flags. The accuracy of the output is compared to performance requirements, error propagation is verified, and out-of-specification values are highlighted.
- Level-2 products output by the L2PP representing retrieved bio- and geophysical parameters are compared to the parameters used as input by the SGM; their accuracies are monitored, error propagation is verified, and out-of-specification or unexpected cases are highlighted and the dependency on the Instrument and L1 processing errors assessed.
- Calibration related output produced by the L1PP either as processed from nominal observation data or related to internal or external calibration, e.g., representing component sensitivities, environmental factors, variable intrinsic properties, geolocation/co-registration aspects, failure lists, gain, biases, etc.

The PAM shall provide the option to be run either in-chain with the previous modules or as a stand-alone execution. For stand-alone execution the PAM shall allow the option for the operator to provide additional metrics and graphs to be evaluated on already-existing simulations, including the combination of results from various simulations (e.g., to evaluate temporal evolution, or compare the performance of different operation modes).

3.5 The Test Data

A reference Test Data Set (TDS) is needed during the acceptance of the E2ES and L1PP modules as well as a reference for the L1OP, L2OP and the Ground Segment.

The deliverable TDS shall be generated using the E2ES and will include nominal and non-nominal cases (to be agreed with ESA) as well as examples of all possible modes/data types/calibrations of <MISSION-X>.[3]

The TDS shall comprise the following elements:

1. Infrastructure (openSF) related items, i.e., configuration files, scenarios, scripts
2. Output of the simulation chain: generated RAW data (Observation (OBS), Ancillary (ANC)), Auxiliary Data Files (AUX) used in simulation, Level 0.
3. Output of the processing chain within the L1PP: L1 Products, Calibration Products plus relevant AUX files, required by L1 processing, corresponding to the inputs supplied in (2), logs
4. Output of the processing chain within the L2PP: L2 Products and auxiliary files (e.g meteo) corresponding to the inputs supplied in (3), logs
5. Any auxiliary software tool needed to reproduce the TDS generation.
6. Checksums for all the files
7. Detailed README
8. Test data specification document, including version of all the processors and listing the complete set of data used in the simulation as well as in the processing to generate the TDS

[3] In case the processing performance of the E2ES, either at the level of SGM, ISM or L1 prevents the TDS generation within an acceptable time, an approach is to generate a smaller core TDS and then duplicate and recondition the data using separate software, to produce a longer data set.

4 REQUIREMENT STRUCTURE

The set of E2ES requirements are organized into categories of interest to the stakeholders, for example performance engineers or scientists. They are grouped according to the type of constraint imposed, e.g., on the operation of software, or the output format, or the set of tools provided. They are named also according to the E2ES component to which they apply. The naming format is XXX-YYY-NNN where XXX is one of the requirement types, YYY is one of the E2ES components, and NNN is a unique number.

4.1 Requirement Types (XXX)

- FUN Functional: capabilities of the simulator
- ARC Architectural: design specifications of simulator
- ENV Environmental: context in which the simulator operates, e.g., computer OS, compiler and library versions
- OPE Operational: general operational requirements
- INT Interface: interface requirements
- OUT Output: logging output requirements
- HMI Human-Machine Interface: user interaction requirements
- VVP Verification & Validation: verification, validation, system integration
- PER Performance: performance e.g., minimum hardware required
- TDS Test Data Set: reference test data set requirements
- OTH Other: miscellaneous

4.2 E2ES Components (YYY)

- GM (Geometry Module)
- SGM (Scene Generation Module)
- ISM (Instrument Simulator Module)
- ODGM (Onboard Data Generator Module)
- L1PP (Level-1 Prototype Processor)
- L2PP (Level-2 Prototype Processor)
- PAM (Performance Assessment Module)
- FCT (Support Functions)
- GEN (General or overall E2E)

5 GENERAL AND SOFTWARE REQUIREMENTS

The requirements listed below are grouped according to the types and applicable components as described above. Some requirements on specific components (e.g., L1PP) are specialised versions of general E2ES requirements, to ensure completeness of the component-level requirements.

5.1 Architecture and general functions

E2E-ARC-GEN-010
The E2ES shall simulate observations and the ground processor outputs, for single or multi-instrument missions, for user-defined scenarios, based on orbit state vector data, UTC observation time, target location, environmental perturbations, instrument(s) configuration and scene definition.

E2E-ARC-GEN-020
The E2ES shall be composed of the following modules in an integrated chain:

1) Geometry Module (GM)
2) Scene Generator Module (SGM)
3) Instrument Simulator Module (ISM)
4) Platform and On-Board Data Generation Module (ODGM)
5) Processing Modules (L1PP)
6) Processing Modules (L2PP)
7) Performance Assessment Module (PAM)

Note 1: The E2ES can be complemented with additional mission specific modules.
Note 2: In astronomy and space science missions an additional Level 0.5 Processing Module is defined.

E2E-ARC-GEN-030
The internal and external E2ES data flow shall be representative of the high-level data flow related to <MISSION-X> within the operational Ground Segment as per Fig 3-1.

E2E-ARC-GEN-040
The E2ES shall provide an end-to-end simulation capability (i.e. from physical scenarios to final data products) allowing the assessment of the fulfilment of the science objectives/mission requirements specified in the [MRD] and allowing the demonstration of compliance against the requirements in the [SSRD].

E2E-ARC-GEN-050
The E2ES shall also be able to support the assessment of the performance of <MISSION-X>, including as well any aspects related to synergy, either between different instruments of <MISSION-X> or with other missions.

E2E-ARC-GEN-060
The E2ES shall support the independent execution of any of the high-level modules specified by E2E-ARC-GEN-020, using as input results generated in earlier simulation runs.

E2E-ARC-GEN-070
The E2ES shall include a baseline set of AUX data containing the necessary instrument characterisation and calibration parameters used in the processing.

Note: This baseline is derived from the Satellite/Payload CCDB when available.

E2E-ARC-GEN-080
The E2ES architecture shall be optimised for performance considering parallelisation at model and system level.

E2E-ARC-GEN-090
The Geometry Module shall implement all functions related to geometry and provide them to all the other modules as needed, e.g., Orbit Propagation (PVT), Attitude Determination (Quaternions), Field of View and Coverage areas, etc., in a centralized manner, providing a centralised abstraction layer to the actual source of data with respect to the rest of the modules.

E2E-ARC-GEN-100
The Geometry Module shall support both stand-alone mode (e.g. internal orbit propagation) and the ingestion of externally generated Orbit and Attitude data (e.g. Predicted, Restituted or externally simulated).

E2E-ARC-GEN-110
The Scene Generator Module shall implement a simulation of the geophysical target based on a user-defined scenario and the corresponding forward model to generate Instrument TOA stimuli including the effect of the atmosphere as well as other stimuli related to external (e.g. lunar, solar, stellar, ground, etc).

Note: The objective of this requirement is to have a single unique interface for every stimulus to be processed towards the measuring system (the ISM). In the specific case of an internal calibration source, it is conceivable that the calibration data are generated within the ISM. In any case the calculation of the stimuli will also result in the relevant stimuli data to be an output so it can be later be compared or re-ingested by the ISM.

E2E-ARC-GEN-120
The Instrument Simulator Module shall model the production of instrument data by implementing a simulation of the measurement process, starting with input stimuli from the SGM, the geometrical conditions (with input from GM), the platform condition (with input from the ODGM) and an internal model of the instrument and its data generation, including instrument configuration and internal

calibrations with a level of detail sufficient to support the mission performance evaluation at each phase of the project.

Note: The ISM shall include functionality to inject as separate parameter errors in both measured stimuli as well as to simulated instrument characteristics. It shall implement the generation of data corresponding to every calibration mode of the instrument. The functionality of platform simulation as input for the modelling (e.g., thermal, electrical, AOCS, propulsion, environmental conditions) and the application of representative errors is allocated, within the reference architecture, to the ODGM Module.

E2E-ARC-GEN-130

The Platform and On-Board Data Generation Module(s) shall:

- simulate any platform parameter needed by the ISM,
- simulate and produce any platform-generated ANC Source Packet in the real satellite packet format that is needed for ground processing, e.g., PVT, AOCS, Temperatures,
- perform the formatting of the ISM generated data (OBS and instrument ANC).

Note: The ODGM shall interface with the ISM to provide any data necessary for ISM functionality. It shall also perform any eventual processing/modification that the platform might apply to the instrument-generated Source Packet (e.g., time stamping, etc.) and shall receive from the GM all data relevant to Orbit and attitude required for generation of the Platform ancillary telemetry.

E2E-ARC-GEN-140

The L1 Processing Module shall ingest satellite data (both RAW and Level 0 formats) and any necessary Auxiliary data (e.g., characterisation data, off-line calibration data, restituted orbit and attitude data, etc) as would be available in the actual ground processing and implement the Level 1 Processing, including any calibration function needed to produce both Calibration products and Level 1 products.

E2E-ARC-GEN-150

The L2 Processing Module shall ingest Level 1 data and any necessary Auxiliary data (e.g., Leaf Area Index, Meteo data, Land-Sea Masks, Ground characterisation maps, Orbit data) as would be available in the actual ground processing and implement the Level 2 Processing to generate Level 2 products.

E2E-ARC-GEN-160

The Performance Assessment module shall support the analysis necessary to evaluate overall system performance and support the verification of Space and Ground Segment requirements. It implements all the separate functions required to assess E2E performance at L1 and at L2 as follows:

- L1: The performance assessment module shall compare the calculated L1 outputs with the instrument stimuli generated by the Scene Generator module, it shall produce statistics, analysis, error characterisation and verification of error propagation and it shall support the verification of all Space Segment Requirements related to L1 quantities (defined as PAM1).

- L2: The performance assessment module shall compare the L2 outputs with the geophysical parameters input to the SGM and produce statistics and analysis necessary to evaluate the overall system performance and support the verification of Space and Ground Segment requirements (defined as PAM2).

Note: Performance assessment module related to L1PP is designated as PAM1. Performance assessment module related with L2PP is designated as PAM2 as they fall under different responsibilities in the E2ES development.

E2E-ARC-GEN-170
The implementation of the high-level modules specified by E2E-ARC-GEN-020 shall be modular in the form of interconnected building blocks, to facilitate the exchange of components by modified or more refined elements.

E2E-ARC-GEN-180
Each E2ES module shall be able to generate all the simulated mission products both error-free and with all/some errors applied, by configuration. It shall also be able to propagate both error-free and with all/some errors data through the processing stages.

E2E-ARC-GEN-190
Each E2ES module shall support calibration following the algorithms in the Algorithm Theoretical Baseline of each instrument. This can be both:

- calibration data to be produced (and subsequently ingested) from instrument outputs
- calibration data coming from external sources

E2E-ARC-GEN-200
The E2ES source code shall include explicit references to the ATBD sections being implemented, to allow easy traceability during the review process.

E2E-ARC-GEN-210
Each E2ES module shall be able to flag anomalous and corrupted data.

Note: The definition of anomalous data is mission specific, but it typically includes duplicate or missing input data, or data that deviates from the expected value.

E2E-ARC-GEN-220
Each E2ES module shall have the capability to read data from external files instead of computing the data internally, by means of configuration.

E2E-ARC-GEN-230
During the execution, the E2ES shall store the inputs and outputs of each of the modules (as defined in E2E-ARC-GEN-020).

5.2 Development/software Requirements

5.2.1 *Environment and common libraries*

E2E-ENV-GEN-010
The E2ES and all its modules, including the L1PP and the L2PP, shall be developed and shall execute on an x86 Linux Operating System environment.

E2E-ENV-GEN-020
The Linux environment shall be based on the latest Ubuntu LTS.

Note: At moment of writing the baseline version of the OS used by the mandated COTS is Ubuntu 22.04 LTS. In the future regular updates by COTS to stay aligned with Ubuntu LTS deliveries are expected.

E2E-ENV-GEN-030
All development shall use permissive-licensed (e.g., BSD, LGPL, MIT, Apache) open-source packages, compiler, and tools unless proved compatible with ESA Community licence [ESACLP].

E2E-ENV-GEN-040
The E2ES shall be developed using openSF as simulation framework, together with the related OSFI libraries version <latest> [OPENSF].

E2E-ENV-GEN-050
All E2ES modules shall be compliant with the latest version of the Generic E2E Simulator ICD [E2EGICD].

Note: At moment of writing the baseline version of the E2E ICD (aligned with openSF) is version 1.

E2E-ENV-GEN-060
All modules of the E2ES shall be written in a compiled language, C or C++ according to C++14 standard or newer. The compiler and version number used shall be TBD.

Note: During Phase 0 and Phase A/B modules might be written in other languages as supported by openSF (e.g., Python, Fortran, Matlab, IDL, Java) but is explicitly required that in Phase B2/C/D the SW is implemented in a performant compiled language. Autocode generated by MATLAB is not allowed.

E2E-ENV-GEN-070
All code shall be written to ensure portability (to bare-metal) across different 64-bit Linux distributions.

Note: The "bare-metal" refers to native compilation and execution on the host operating system without use of containers or a VM. Containers can be used for delivery purpose only but not to as mechanism to implement portability across different/newer Linux distributions.

E2E-ENV-GEN-080
The E2ES and its modules shall make use of the latest stable version of the EO Mission software [EOCFI] libraries for all orbit, attitude, and geometrical calculations.

E2E-ENV-GEN-090
The L1PP and L2PP shall support multithreading and parallel processing by using the OpenMP library X.Y or newer.

E2E-ENV-GEN-100
All non-binary files, with the exclusion of the [E2EGICD] Configuration Files, shall be compliant to the [EOFFSDT] standard.

E2E-ENV-GEN-110
Any GPU support shall be implemented using a common API for all modules/tools (e.g, openCL, CUDA, Vulkan).

E2E-ENV-GEN-120
Build scripts shall be based on Cmake version 3 or newer.

5.2.2 *Other Software requirements*

E2E-OTH-GEN-010
The developed SW and the development environment shall include provision and procedures for plug-in replacement of the version of COTS used (openSF, OSFI, EOCFI).

Note: This is a technical maintainability requirement to allow keeping the COTS (openSF, OSFI and EOCFI) regularly updated for bug fix and performance improvements with small or negligible effort.

E2E-OTH-GEN-020

The E2ES code shall not include any:

- Uninitialized and unused variables
- Memory leaks

Note: In general, when activating the compiler warning flag (-Wall -Wextra), no compiler warnings should be issued the build.

E2E-OTH-GEN-030

It shall be possible to execute every E2ES module as standalone, outside the E2ES environment, i.e. outside the E2ES processing chain and outside the E2ES HMI.

E2E-OTH-GEN-040

All individual E2ES SW modules shall be able to be executed standalone from arbitrary user-defined location in the filesystem independently from the location of any other data ingested or produced.

Note: For this purpose, it is recommended to use of the system environmental variables defined in the [E2EGICD] to identify input/output location.

E2E-OTH-GEN-050

The integrated E2ES comprised of orchestrating framework, SW modules, data and configuration files and any other private data defining the simulations (e.g., database, parameter iteration definition etc) shall be able to be installed and executed in arbitrary user-defined locations in the filesystem as well be able to be transferred to another computer.

Note: This requirement intends to ensure that there is no pre-defined location on where the E2ES SW or its data must be installed to work using the environmental variables as defined in [E2EICD] as necessary.

E2E-OTH-GEN-060

All the data and configuration files shall be uniquely identified through their file name and not only via their location in the filesystem.

E2E-OTH-GEN-070

Every software delivery shall include an installation kit to fully automatize the installation including the option to perform a clean removal of the previous version as well as to set up any required environmental variable.

E2E-OTH-GEN-080

Installation and uninstallation functionality shall allow the user to select if only the SW or also the data have to be removed.

E2E-OTH-GEN-090
It shall be possible to install the E2ES without being a UNIX super-user.

E2E-OTH-GEN-100
The installer shall allow the user to select any location within the filesystem for the E2ES.

E2E-OTH-GEN-110
All software deliveries shall be full i.e., no patches and include integrity check mechanism (e.g., md5).

E2E-OTH-GEN-120
The E2ES SW delivery shall be of 2 types:

- User Delivery (containing the binary run-time E2ES as well as all the data and configuration)
- Developer Delivery (containing all the User Delivery + source code, build, test, and packaging scripts)

E2E-OTH-GEN-130
The SW and data deliveries shall be packaged so that they are delivered as a single file and not include any unused file.

E2E-OTH-GEN-140
The SW deliverables shall include all configuration and data items to perform the verification as per the SSVP.

E2E-OTH-GEN-150
All deliverables shall be covered by a 12-month warranty for corrective maintenance or bug fix.

E2E-OTH-GEN-160
The E2ES delivery shall include a report of performance metrics created by the functionality described in E2E-OTH-GEN-160 for a run on a reference E2E Simulator configuration, also to be included in the delivery, including the relative utilization efficiency of CPU cores, memory, and storage I/O.

Note: The report shall include a detailed description of the computer hardware on which the simulation was performed as well as the SPECfp2017 benchmark score of that same computer system.

E2E-OTH-GEN-170
The E2ES software development shall employ best-practices such as continuous integration, static code analysis, and conformance to coding standards.

E2E-OTH-GEN-180
All software shall be maintained under strict configuration control using a CM tool (e.g., GIT).

E2E-OTH-GEN-190
The E2ES development shall be tracked through an on-line SPR tracing/ticketing system (e.g., JIRA, Redmine).

E2E-OTH-GEN-200
The E2ES build script shall include an option for the user to build executables suitable for profiling (using tools such as gprof, valgrind) or for performance (the default build).

E2E-OTH-GEN-210
The SW deliverables shall be built, tested, and delivered in both Debug mode (no optimisation, full debug information) and Release mode (enable compiler optimisation corresponding to O-3 and no debug info included).

E2E-OTH-GEN-220
The E2ES delivery shall include a script or equivalent functionality that displays software performance metrics for a run of the software compiled for profiling, as described in E2E-OTH-GEN-200).

E2E-OTH-GEN-230
The Configuration managed code and data repository (E2E-OTH-GEN-190) shall be remotely accessible (read-only) from the end customer (ESA).

E2E-OTH-GEN-240
The software (including adopted components such as mathematical libraries) shall be selected and developed to be compatible with a release of source and binary according to the ESA Community License Type 3.

E2E-OTH-GEN-250
The E2ES software shall provide means to validate offline all non-binary input files: e.g., XML inputs shall make use of schema as validation tool [HXML].

E2E-OTH-GEN-260
The E2ES shall allow to validate all to-be-used input configuration files before running the simulation, checking that the syntax for all configuration parameters of the modules to be used is correct.

Note 1: Configuration parameter's semantical check is excluded. The [OSFI] library automatically write/read syntax corrected configuration files and return errors otherwise.
Note 2: The Parameter Editor application part of the openSF suite automatically edit/format/check configuration file compliant to [E2EGICD].

---

6 FUNCTIONAL REQUIREMENTS

The requirements described in this section apply to the overall E2ES tor and need to be implemented by the framework or by any of its modules/tools (e.g., compare functionality is not implemented by the openSF framework but rather by a dedicated tool/module).

6.1 General Functional Requirements

E2E-FUN-GEN-010
The E2ES shall include the functionality to process:

- Input geophysical parameters
- Input scenes
- Observation data (RAW)
- Ancillary (ANC) data
- Real or simulated level 0 data
- Real or simulated level 1 data
- Auxiliary (AUX) data

Into

- Level 0 data
- Level 1 data
- Level 2 data
- L1 and L2 Performance assessment and compliance reports
- Calibration data (CAL)
- Breakpoints (BRK)

E2E-FUN-GEN-020
The E2ES shall include a functionality to allow controlling and monitoring the E2ES operation and managing its related data (input, output, databases).

E2E-FUN-GEN-030
The E2ES shall include functionality to allow:

- Computing the geometric, and other <MISSION-X> specific performance indicators
- Optimising the instrument settings as function of the level 1 processing results
- Assessing the data quality
- Assessing the overall <MISSION-X> performance
- Statistical analysis of the instrument behaviour and algorithm performance over configurable range of data and simulation time

E2E-FUN-GEN-040

It shall be allowed the use of the E2ES in different user-selectable configurations allowing the users to select, independently of the configuration, the source of the inputs and the destination of the outputs.

E2E-FUN-GEN-050

The E2ES functions shall be able to simulate any of the operational scenarios encountered by the instrument throughout its life.

Note: Examples of the scenarios or conditions are different season, In eclipse or during daytime, At Beginning Of Life (BOL) or at End Of Life (EOL), Calibration targets, etc.

E2E-FUN-GEN-060

The E2ES shall include a functionality for the assessment of the impact of data correction and processing algorithms on data quality.

E2E-FUN-GEN-070

The E2ES shall include functionality to execute and mix-and-match in the same environment both the Phase 0/A Modules (e.g., older version of OSS) as well as the newly developed modules with minimal adaptation of the architecture and interfaces.

Note: This functionality supports L1 integration e.g., allowing direct and easy comparison in the same environment as well as in the PAM of the results generated with the Phase 0/A OSS version and the one generated within Phases B1/B2/C/D/E1.

E2E-FUN-GEN-080

The E2ES shall be capable of computing on-line calibration data.

E2E-FUN-GEN-090

The E2ES shall be able to generate and ingest off-line calibration data.

Note: The generation of off-line calibration data can take place in the PAM.

E2E-FUN-GEN-100

The E2ES modules shall all have the capability to be run stand alone, with user-specified input and output files as per [E2EGICD].

E2E-FUN-GEN-110

The E2ES modules shall be able to exchange all data i.e., products, auxiliary and calibration data, in a format compatible with the product and file formats used in the data processing ground segment.

Note: To be implemented when data processing ground segment specification is available.

E2E-FUN-GEN-120
The E2ES shall store and load all the configuration data needed for the operation of its internal modules.

E2E-FUN-GEN-130
The E2ES, once configured and started, shall be able to run unattended, i.e., without any user intervention.

E2E-FUN-GEN-140
The E2ES and all its modules shall be able to gracefully handle any software error and among others that:

- generation of core dumps or crashes do not occur
- resources are freed
- no child process is left running
- error and warning messages are self-explanatory processes terminated gracefully with an appropriate return status
- trap mathematical errors (e.g., division by zero, underflow, overflow, etc.) and handle these errors without stopping the execution

E2E-FUN-GEN-150
The E2ES and all its modules shall be able to gracefully handle any non-nominal processing or degraded mode (e.g., missing inputs) and ensuring that:

- error and warning messages are self-explanatory
- flags are set on impacted data
- processes terminated gracefully with an appropriate return status

Note: The way in which errors are handled (abort, use of a model, continue, etc) is left to the detailed design phase and to the [ATBD] specification.

E2E-FUN-GEN-160
All the E2ES modules shall be able to correctly handle geometric boundary conditions.

Note: Geometric boundary conditions are, for example, international date line, etc.

E2E-FUN-GEN-170
The E2ES and all his modules shall be able to correctly handle date and time boundary conditions.

Note: Date and time boundary conditions are, for example, leap years and leap seconds.

E2E-FUN-GEN-180
The way to operate and use the E2ES and any of its modules shall be independent of the type of data used when multiple types are possible.

Note: E.g., the L1PP Module will be able to perform all its functions whether the input is Level 0 or RAW.

E2E-FUN-GEN-190
The E2ES shall support the cross-validation against the instrument teams breadboards to ensure the E2ES performance (e.g., algorithm implementation) with respect to the performance models defined by the instrument providers.

E2E-FUN-GEN-200
The E2ES shall be able to work in 2 modes:

- compatible with invocation as per openSF/generic E2E ICD [E2EGICD] when used within the E2E Performance simulator infrastructure
- standalone use (user invocation by command line)

E2E-FUN-GEN-210
The E2ES and all its modules shall be able to process/handle all <MISSION-X> instrument(s) configurations and settings including diagnostic, calibration and test modes.

Note: This requirement addresses the generally common case where each instrument can be considered in isolation. Missions including multiple synergetic instruments either on-board or on ground should define requirements to ensure that E2ES architecture supports the synergetic data flows/functionality.

E2E-FUN-GEN-220
The E2ES and its modules shall be able to support time-based scenario orchestration (see [E2EGICD] sections 2.3 and 3.3), allowing automated batch test data generation, processing, and performance assessment reporting of a simulated operational timeline.

Note: In the reference architecture the ground segment modules i.e., the L1PP, L2PP and PAMs, do not execute in a time-base but always in data-driven mode.

E2E-FUN-GEN-230
It shall be possible to specify and execute a time-based scenario by specifying start and end times.

E2E-FUN-GEN-240
The E2ES scenario orchestration function of E2E-FUN-GEN-220 shall allow defining a list of instrument operating modes (limited to the simulation modules alone) on a time basis, e.g., 10 minutes in mode A, 5 minutes in mode B, etc. as defined in [E2EGICD] sections 2.3 and 3.3.

Note: In the reference architecture the ground segment modules i.e., the L1PP, L2PP and PAMs, do not execute in a time-base but always in data-driven mode.

E2E-FUN-GEN-250
It shall be possible in the E2ES to define, save, and load the time-based scenarios as per format defined in the [E2EGICD].

E2E-FUN-GEN-260
It shall be possible to use the E2ES to generate RAW and Level-0 Test Data Sets to be used in the Ground Segment validation.

E2E-FUN-GEN-270
The E2ES shall allow the user to inject errors/perturbations to the space segment simulation, including among others: platform behaviour, instrument behaviour and data stream errors.

E2E-FUN-GEN-280
The parameters of each module of the E2ES that are identified as perturbable or that will need to be iterated for sensitivity analysis, montecarlo etc, shall be exposed via the standard configuration file parameter [E2EGICD].

Note: By doing this automated user defined iteration and perturbation are directly provided by [OPENSF] and no dedicated functionality is required in the module.

E2E-FUN-GEN-290
It shall be possible, in a single operation, to save and load the overall simulation context to reproduce on a separate instance of the E2ES exactly with the same configuration.

E2E-FUN-GEN-300
Every module of the E2ES shall verify the correct formatting of any input data and be robust to any format error.

E2E-FUN-GEN-310
In case a format error is detected, a meaningful error message shall be issued which shall include at least: filename, nature and position/location of the error.

E2E-FUN-GEN-320
It shall be possible to install, multiple versions of the E2ES on the same computer.

E2E-FUN-GEN-330
It shall be possible to execute the E2ES from two different user accounts and keep all configuration and all data separated.

E2E-FUN-GEN-340
The output data shall include a log of the E2ES runs specifying:

- An E2ES run identifier
- The simulation support data
- Any support data if used
- The conditions under which the E2ES is operating, including a reference to all the input data used in the run and to all the output data generated; a summary of the main features of the run shall also be included (e.g., E2ES configuration, E2ES version/release number, mode of operation, functions inhibited, settings, etc.)
- The status of all the E2ES functions during the E2ES operation
- The content of the E2ES internal variables, as selected by the user
- All non-nominal events
- All real-time events
- Duration (wall-clock and, as a debugging option, CPU time) of the E2ES overall and of each module
- Unique identifier

E2E-FUN-GEN-350
Every module shall allow in the log for different severity category of messages as: DEBUG, INFO, WARNING, ERROR (as per [E2EGICD]).

E2E-FUN-GEN-360
Every module shall report progress and completion status in compliance to [E2EGICD].

E2E-FUN-GEN-370
It shall be possible to enable or disable the generation of each category of messages.

E2E-FUN-GEN-380
The E2ES shall support Monte-Carlo analyses. In particular, it will be possible to assess error propagation statistically by Monte Carlo analysis: several scenarios are executed and the output files include the particular realization of both systematic and random errors, which are modelled by means of random number generator following statistical distributions for each type of noise and accounts for differences between systematic and random errors as well as for the correlation between errors.

Note: OSFEG (openSF Error Generation Library) provides methods for mathematical modelling of a perturbation within statistical analysis scenarios. OSFEG is provided in the form of source code.

E2E-FUN-GEN-390
All algorithms depending on random number generators, in particular error generation, will allow the setting of the random number generator seed, to allow for reproducible results.

E2E-FUN-GEN-400
The E2ES shall support functional error propagation. In addition to the noise-free output files, additional error files will be generated. These files contain the variance for each value of the signal. A covariance matrix describing also the correlations in the resulting errors can be later propagated through the rest of the modules by using classical statistical techniques.

E2E-FUN-GEN-410
General purpose functions used throughout multiple processing modules shall be implemented into libraries to avoid code duplication.

E2E-FUN-GEN-420
The user shall be able to control the simulation either via command line or via a user interface (openSF or dedicated HMI).

E2E-FUN-GEN-430
The user shall be able to delete an entire scenario.

E2E-FUN-GEN-440
The user shall be able to copy a scenario, modify its inputs and settings and launch the new scenario.

E2E-FUN-GEN-450
The user shall be able to enable/disable by configuration a debug execution mode that generates breakpoint files for analysis.

E2E-FUN-GEN-460
The user shall be able to store all intermediate results of each of the modules.

E2E-FUN-GEN-470
The E2ES shall be able to distinguish and manage permissions for access/operations to the simulation and its parameters separately for the following user categories:

- Operator
- Expert operator
- Developer

Note: This requirement is intended to avoid misconfigurations of the E2ES by edition of files such as the calibration database which requires expert knowledge.

E2E-FUN-GEN-480

The E2ES shall make no assumptions on the machine on which it is running, the directory it is running in, the directories where input and output files are expected, the memory addresses that his code or data are loaded into.

6.2 Geometry Module (GM) Functional Requirements

E2E-FUN-GM-010

The GM shall implement all common functions related to geometry and provide them to the other modules. These functions shall include, when applicable, at least:

- Orbit Propagation (PVT)
- Platform Attitude Determination (Quaternions)
- Field Of View
- Line of Sight
- Coverage and Coverage areas
- Time functions
- Geolocation functions
- Celestial target calculation (sun, moon, etc)
- Support for calibration manoeuvres (e.g., special attitude modes TBD) Platform/Instrument geometry coupling
- Scene interaction geometry

E2E-FUN-GM-020

Any necessary geometrical calculations, e.g., Sun/star position, satellite position, attitude, required as input to the instrument model or to generate ancillary (ANC) HKTM, shall be computed using the data computed by the common GM.

E2E-FUN-GM-030

The GM shall be able to perform its function both in stand-alone mode (e.g., by mean of internal orbit propagation) or in externally driven mode (e.g., by ingestion of externally generated Orbit and Attitude data).

E2E-FUN-GM-040

The EO Mission Software CFI libraries version 4.X or higher [EOCFI] shall be used for all orbit, attitude, and geometrical calculations.

E2E-FUN-GM-050
The GM file input formats (e.g., Orbit, Attitude, etc.) shall be compliant with [CFIFS].

E2E-FUN-GM-060
Within the E2ES the GM shall consist of a standalone module with persistent input and output data.

E2E-FUN-GM-070
The GM shall allow the introduction of errors to the calculated orbital, attitude and geometrical related parameters as bias, random and harmonic behaviour.

E2E-FUN-GM-080
The GM shall implement timing functions and handle any discontinuity (e.g., leap seconds) in a centralised way.

E2E-FUN-GM-090
The numerical errors introduced by the GM module shall contribute in a negligible way to the instrument error budget.

E2E-FUN-GM-100
The GM shall compute and generate the satellite PVT and attitude for both Nominal and Real orbit as required by the E2ES modules.

Note: In Phase B1 and later the L1PP and PAMs always compute all their known geometry solely based on the content of the generated RAW/L0 data and on static AUX files. No output from the GM is made available to the ground processing modules.

6.3 Scene Generation Module (SGM) Functional Requirements

E2E-FUN-SGM-010
The SGM shall implement and compute the geophysical target/values (the truth) based on a user-defined scenario and input data.

E2E-FUN-SGM-020
The SGM shall be able to ingest all static or time-varying data needed to provide input to other modules for that epoch of simulated acquisition.

E2E-FUN-SGM-030
The SGM shall consider any external factor affecting the forward model or the geophysical target in calculating the stimuli to the ISM (e.g., sun illumination, AUX data, etc.).

E2E-FUN-SGM-040
The SGM shall implement the forward model to generate geophysical parameters and observable stimuli to the instrument (e.g., radiance spectra, reflected signal, etc.) based on the geophysical target or calibration source, observation geometry, and the effect of environment or another signal-propagation medium.

E2E-FUN-SGM-050
The SGM shall generate the stimuli to be used as input to the ISM.

Note 1: The ideal flexible architecture (illustrated in Fig. 3-3) entails that the stimuli scene being computed to the maximum extent on a grid independent from the position and attitude of the observing system and leaving to the ISM the task to sample it at the appropriate resolution for the relevant line of sight, thus facilitating the run of the simulation in different geometry conditions without the need of re-running every time computationally expensive SGM simulations.
Note 2: Implementing only a flexible architecture as per Note 1 might not be possible or desirable for processing performance or for accuracy reasons. In that case it might be required to compute stimuli directly on the target ISM (instrument) grid. It is recommended in that case to implement this stimuli generation as a special option within the flexible architecture controlled by E2ES configuration flags applicable to SGM, GM and ISM.

E2E-FUN-SGM-060
The SGM shall allow generation of scenes for at least TBD time (variability of the scene) including any applicable stimuli from external calibration sources.

Note: To simulate entire orbits of data, the SGM must not functionally be limited to work only on individual scene and must be able to use the orbital and attitude geometry (as defined by the GM) to generate corresponding sequence of stimuli covering range of multiple orbits.

E2E-FUN-SGM-070
A simplified scene capability shall be implemented by the SGM to support long duration scenario generation, for example multiple orbits according to mission timeline and dynamic GM computations.

Note: The implementation can be using look-up tables or an emulator in place of a full model (e.g., an RTM), or in a simplified Forward Model, provided that the L2 processing can be performed, and the retrieval uncertainties quantified.

E2E-FUN-SGM-080
The SGM shall support the E2ES time-orchestration for the generation of long scenarios.

Note: The purpose of having the SGM supporting the long-term scenarios is to create test data sets for Industry to test the calibration approach, to generate test data for the Ground Segment.

E2E-FUN-SGM-090
The SGM shall be able to load pre-configured scenes to ease execution by the user. The default input, without parameter specification beyond scene size, acquisition epoch, and location, shall generate scenes with a realistic distribution of geophysical parameters.

E2E-FUN-SGM-100
The SGM shall allow the user to configure if the atmospheric model is used (it shall be possible to switch it on and off).

E2E-FUN-SGM-110
The SGM shall allow the definition and automatic injection in the simulation of user-defined errors (biases, drifts, statistical errors, noise, time, or space dependent) on any of its input data (including the atmospheric model) to simulate the effect that these errors would have on the generated stimuli and external-calibration-source stimuli.

E2E-FUN-SGM-120
The SGM shall allow generation of stimuli at a user specified resolution to allow oversampling. Oversampling shall be possible on spatial as well as on other relevant signal characteristics e.g., spectral resolution.

Note: Oversampling of stimuli allows minimisation of interpolation or gridding errors by the ISM. See Fig. 3-3.

E2E-FUN-SGM-130
The SGM shall support the generation of calibration data in the ISM by generating the stimuli corresponding to external astronomical or ground based calibration target (e.g., sun, stars, moon, transponders).

E2E-FUN-SGM-140
The numerical errors introduced by the SGM shall be negligible compared to the instrument error budget.

E2E-FUN-SGM-150
The SGM shall be a scalable w.r.t. the use of multiprocessors/multicore computer by making use of a multi-threading architecture.

E2E-FUN-SGM-160
The SGM shall be capable of running multiple (independent) instances simultaneously on the same machine.

E2E-FUN-SGM-170
The SGM shall make available to the Performance Assessment Module all the inputs needed to perform comparison with the retrieved L2 data, including at least: geophysical target data, injected errors (if any), BOA, TOA, and external-calibration-source stimuli.

E2E-FUN-SGM-180
The forward model implemented in the SGM shall be consistent with the L2PP retrieval algorithms.

6.4 Instrument Simulator Module (ISM) Functional Requirements

E2E-FUN-ISM-010
The ISM shall perform the simulation of the <MISSION-X> instruments based on the input stimuli from the SGM, the geometrical conditions (with input from GM) and an internal implementation of the instrument model. It will include all thermo-mechanical, attitude control, and disturbance effects on the input stimuli to the Instrument Module. It shall contain at least the following functionality:

- Detailed simulation of the Instrument and its components
- Simulation of all Instrument Modes (observation and calibration)
- Environmental effects (e.g., thermal, electrical)
- Error simulation
- Timeline simulation (e.g., Measurement and Calibration mode switching)
- CCSDS Instrument Source Packet generation (f this formatting functionality is not delegated to the OBDGM)

E2E-FUN-ISM-020
The ISM(s) shall handle the simulation of the sensor(s) behaviour, having different outputs depending on the type of instrument.

E2E-FUN-ISM-030
The ISM shall provide the possibility to have distinct configurations per instrument.

E2E-FUN-ISM-040
The ISM shall implement any additional platform simulation needed as input for the modelling (e.g. thermal, electrical, AOCS, propulsion, environmental conditions).

E2E-FUN-ISM-050
The ISM shall be able to generate fully realistic measurement parameters as would be generated by the instrument in all its modes of operation including in particular every measurement and calibration mode.

E2E-FUN-ISM-060
The ISM shall ingest in form of AUX files formatted compliant to [EOFFSDT] the necessary data and parameters characterising the instrument either as design or as ground calibrated default values, and not hard-code these data in software.

E2E-FUN-ISM-070
The output of the ISM shall be formatted as RAW CCSDS Source Packets telemetry and with all fields and parameters fully representative of those generated on-board including both scientific and ancillary parameters (e.g. timing, temperature, position, velocity etc).

Note 1: In case the CCSDS formatting is delegated to the ODGM the output of the ISM can be produced in other formats (e.g., netCDF).
Note 2: This requirement does not apply in Phase 0/A/B1.

E2E-FUN-ISM-080
The ISM (together with the ODGM) shall provide to the (simulated) ground segment all the dynamically generated platform data, required by processing, in the same form, as they would be available in the actual ground segment. This shall apply in the case that the data is included as parameters within the Instrument Source Packets or is formatted as separate ancillary/HKTM source packets generated by the platform (e.g., GNSS and Star Tracker source packets).

E2E-FUN-ISM-090
The ISM shall allow to define and automatically inject into the simulation user-defined errors (e.g., bias, drifts, statistical errors, noise, linear, harmonic, time, or space dependent) applied to any of its input data or internal modelling elements to simulate the effect that these errors would have on the mission products at Level-1 and Level-2.

E2E-FUN-ISM-100
The parameters controlling the errors in the ISM shall be configurable and separated from the quantities they apply to.

E2E-FUN-ISM-110
The ISM shall allow configuring the operation of its internal model as it would be done in the real instrument via TC whenever this affects the results (e.g., change of measurement integration time, delay windows, modification of calibration sequences executed on board as OBCP, etc.). This configuration shall be performed via a static auxiliary configuration file and shall not require any recompilation.

E2E-FUN-ISM-120
The ISM shall use data generated by the Geometry Model to access geometry related quantity, e.g., simulated position, timing, attitude, line of sight, visibility, occultation, Sun illumination, manoeuvres, etc.

E2E-FUN-ISM-130
The ISM shall generate the stimuli corresponding to the internal calibrations (e.g. LED, black bodies, noise diodes, diffusers).

E2E-FUN-ISM-140
The ISM shall support the control of the operational modes (e.g. observation mode, calibration modes, etc) via configuration files using the module-execution modes mechanism defined in section 2.3 of [E2EGICD].

Note: This requirement enables automated batch execution of user-defined operational scenarios with openSF.

E2E-FUN-ISM-150
The ISM shall be a scalable w.r.t. the use of multiprocessors/multicore computer by making use of a multi-threading architecture.

E2E-FUN-ISM-160
The ISM shall be capable of running multiple (independent) instances simultaneously on the same machine.

6.5 Platform and On-Board Data Generation Module (ODGM) Functional Requirements

E2E-FUN-ODGM-010
The ODGM shall implement all functions related to the generation of on-board RAW telemetry for both ISM and Platform in its final and complete form as it would appear on the space to ground link [S2GICD].

E2E-FUN-ODGM-020
The ODGM shall interface with the ISM generating the platform parameters (e.g. temperature, voltages, navigatiion data, on-board time (OBT), PPS pulse, etc.) needed by the ISM model to simulate both observations data (OBS) and instrument ancillary data (ANC).

E2E-FUN-ODGM-030
The ODGM shall support the generation of CCSDS idle packets both at level of instrument and at the level of the platform.

E2E-FUN-ODGM-040
The ODGM shall implement simulation as well as the generation of platform ancillary data (ANC) required by the L1 ground segment processing, including at least navigation packets and HKTM packets, in the correct CCSDS format as per [S2GICD].

E2E-FUN-ODGM-050
The ODGM shall ensure that the simulated on-board data coming from different sources (instrument, AOCS, etc) is consistent with the timing (e.g. timestamps) and to the environmental and geometrical conditions.

E2E-FUN-ODGM-060
The ODGM shall use data generated by the GM to access geometry related quantity, e.g. simulated position, timing, attitude, line of sight, visibility, occultation, Sun illumination, manoeuvres, etc.

E2E-FUN-ODGM-070
Any CCSDS space packet generated by the ODGM shall comply to the format as per [S2GICD].

6.6 Level 1 Prototype Processor (L1PP) Functional Requirements

NB: Requirements in the L1PP section of this document are numbered identically to the ones in the L2PP, with adaptations as required. In general requirements with the same number address the same aspect. This approach should be respected when tailoring.

E2E-FUN-L1PP-010
The L1PP shall ingest satellite data (both RAW and Level 0 formats) and implement the level 1 Processing including any calibration function needed and output both of Level 1 products and dynamic (on-line) Calibration products (see Fig. 3-8) depending on the input.

E2E-FUN-L1PP-020
The processing of L1PP shall be implemented according to the [L1ATBD]. The implementation shall include both processing observation as well as processing of on-line calibration data (on-line CAL).

E2E-FUN-L1PP-030
When fed with RAW data (instrument source packets), the L1PP shall implement a function to generate and output the corresponding Level-0 product prior to performing L1 processing.

E2E-FUN-L1PP-040
The L1PP processor shall produce Level 1 products and on-line Calibration products in format according to [L1IODS].

Note: Although less efficient for flexibility reason it highly recommended that on-line calibration products are explicitly generated as CAL file to be automatically ingested by the L1PP rather than shortcutting within the L1PP as this allow independent control of CAL and OBS processing and facilitates anomaly investigation.

E2E-FUN-L1PP-050
The L1PP shall be compliant with the file naming convention of the Level 1 format definition as used in the operational Ground Segment.

Note: All related to the operational ground segment is currently undefined.

E2E-FUN-L1PP-060
The L1PP shall allow to define its input as belonging to two categories: either mandatory or optional, in line with the specification in the [L1ATBD]. A missing mandatory input shall result in "No processing" while a missing optional input shall result in "Degraded processing" and marked in the L1 product by means of a quality flag specified in the [L1ATBD].

Note: This requirement defines the functional aspects of the L1PP orchestration. The definition of optional input is specified in the [L1ATBD].

E2E-FUN-L1PP-070
The L1PP shall be able to directly ingest any necessary AUX data (e.g. Characterisation data, static calibration data, orbit and attitude data, etc.) in the same format in which these would be available in actual ground segment.

Note: This requirement is a special case of E2E-INT-GEN-010.

E2E-FUN-L1PP-080
The L1PP module shall implement a dedicated mode to enable processing of data generated on-ground prior to launch where ancillary/auxiliary data might be missing or incorrect (e.g. navigation data, timing data, counters, etc) or the Calibration data are provided as external AUX or CAL files and not within the instrument data flow.

E2E-FUN-L1PP-090
The L1PP shall allow the writing of intermediate breakpoint data (BRK) e.g. in between processing steps/internal algorithms, before or after calibration, etc.

E2E-FUN-L1PP-100
A user-enabled flag shall control generation of BRK within L1PP.

E2E-FUN-L1PP-110
The L1PP shall be a scalable w.r.t. the use of multiprocessors/multicore computer by making use of a multi-threading architecture.

E2E-FUN-L1PP-120
The L1PP shall be capable of running multiple (independent) instances simultaneously on the same machine.

E2E-FUN-L1PP-130
The L1PP shall make no assumptions on the machine on which it is running, the directory it is running in, the directories where input and output files are expected, the memory addresses that his code or data are loaded into.

Note: As consequence it shall be possible to move the L1PP SW in any location within the filesystem while maintaining all its functionalities and its ability to execute.

E2E-FUN-L1PP-140
The L1PP shall be capable of interfacing with the simulation framework openSF [OPENSF].

E2E-FUN-L1PP-150
The L1PP shall be able to work in 2 modes:

- compatible with invocation as per openSF/generic E2E ICD [E2EGICD] when used within the E2ES infrastructure
- standalone use (user invocation by command line)

E2E-FUN-L1PP-160
In the standalone mode, the L1PP shall be self-orchestrating including the selection of the observation and applicable calibration and auxiliary data to allow stand-alone use (i.e. not requiring any external generated job order) .

E2E-FUN-L1PP-170
In the standalone mode, while executing, the L1PP shall be data-driven i.e. shall iteratively perform the RAW/L0 to L1 processing based on automatic detection of process-able data in an input directory (i.e. no repeated manual invocation from command line needed) until all data have been processed.

E2E-FUN-L1PP-180
In standalone mode, the L1PP shall be able to resume the processing taking as input the intermediate L1 data products (e.g. starting from L1a or from L1b).

E2E-FUN-L1PP-190
The L1PP shall not require any user interaction therefore allowing automated unattended batch processing.

E2E-FUN-L1PP-200
The L1PP shall be able handle changes in the <MISSION-X> operational baseline without requiring recompilation and it shall not assume a hard-coded sequence of operating modes. Any such sequence shall be implemented if needed as dynamic configuration items.

Note: This only applies in the case that there is no update to the algorithm.

E2E-FUN-L1PP-210
The L1PP shall have the capability to limit the data processing to a time range that can be specified using run-time configuration.

Note: This functionality allows to select a subset of data present at its input to support faster processing of a specific time segment.

E2E-FUN-L1PP-220

The L1PP shall have the capability to limit the data processing to a set of one or more measurement modes that can be specified using run-time configuration (e.g. only process data packets of a specific APID).

E2E-FUN-L1PP-230

The L1PP shall be able perform aggregate/accumulate calculations that span multiple measurements/time intervals/orbits e.g. averaging of several measurements on a sliding window or determining a median across multiple measurements as required by the [L1ATBD].

Note: This requirement supports the SW architectural aspects implementing any needed persistence across data sets that is required for aggregation and accumulation.

E2E-FUN-L1PP-240

The L1PP shall be designed to allow easy replacement, addition, or removal of algorithms with no or minimal changes in the architecture or structure of the L1PP software, e.g. by use of dynamic libraries.

E2E-FUN-L1PP-250

The internal and external L1PP data flow shall be representative of the high level data flow related to the <MISSION-X> within the operational Ground Segment as per Fig. 3-8.

E2E-FUN-L1PP-260

The L1PP shall calculate geo-location information for all target-related parameters as specified in [L1ATBD] and [L1IODS].

E2E-FUN-L1PP-270

The L1PP shall be able to calculate/estimate the uncertainty (random + systematic) of the generated Level 1 product.

E2E-FUN-L1PP-280

The L1PP shall compute and output all parameters needed on-line or off-line (e.g. by the PAM or by an offline Monitoring Facility) to evaluate the instrument performance trend monitoring as required by [SSRD].

E2E-FUN-L1PP-290
The L1PP shall be able to compute and include in the output products, the data quality flags that account for both the L1 processing algorithm conditions (e.g. numerical convergence) as well as flags in the instrument measurement data, in the instrument ancillary data, and in the satellite ancillary data used as input.

Note: The logic and the algorithm to interpret, compute and generate data quality flags is defined in [L1ATBD].

E2E-FUN-L1PP-300
The L1PP shall be able to identify, process and flag degradation (as per [L1ATBD]) in the L1PP input data including:

- Duplicated packets
- Packets out of order
- Non-nominal packets
- Corrupted and missing packets
- Data gaps inside the packets
- Instrument parameters out of nominal range
- Nominal measurements out of nominal range

E2E-FUN-L1PP-310
The L1PP shall be able to generate the L1 and Calibration products even in case of degraded input data (when this is possible as per [L1ATBD]), and in such a case it shall flag output as being of degraded quality.

E2E-FUN-L1PP-320
The L1PP shall flag and report any degraded output product, in the metadata part of the same products.

E2E-FUN-L1PP-330
The presence of invalid data (i.e. data that cannot be processed) in the L1PP input files shall not prevent the processing of the remaining valid data within the same input files.

E2E-FUN-L1PP-340
The L1PP software shall rely only on internal algorithms and on data available within its input data files for all geometrical, orbital or attitude calculations; specifically, it shall not interface with the SGM, ISM or GM used in the simulation part of the E2ES.

Note: This requirement ensures that real values and errors included in the simulation are not known to the L1PP which receives its inputs only via S2G (Space to Ground) data link and on-ground AUX files (e.g., instrument characterisation data).

E2E-FUN-L1PP-350
The L1PP processing scheme shall avoid, to the maximum extent possible, dependencies between days or orbits to allow out-of-sequence processing; if unavoidable, the maximum effort should be made to avoid the need of sequential processing.

Note: This requirement supports orbit level parallelisation of the processing.

E2E-FUN-L1PP-360
The L1PP numerical implementation of the algorithm describe in the [L1ATBD] shall be documented in a "Level-1 Detailed Processing Model" DPM document which shall specify the detailed processing steps required to generate Level-1 products.

E2E-FUN-L1PP-370
It shall also provide the necessary inputs to the PAM to perform the end-to-end comparison of retrieved physical quantities with the simulated quantities input to the ISM.

6.7 Level 2 Prototype Processor (L2PP) Functional Requirements

NB: Requirements in the L2PP section of this document are numbered identically to the ones in the L1PP, with adaptations as required. In general requirements with the same number address the same aspect. This approach should be respected when tailoring.

E2E-FUN-L2PP-010
The L2PP shall ingest Level 1 data, perform the level 2 Processing, and produces the Level 2 products.

Note: This requirement is a specialization of E2E-INT-GEN-010.

E2E-FUN-L2PP-020
The L2PP shall be implemented according to [L2ATBD].

E2E-FUN-L2PP-040
The L2PP shall produce Level 2 products in format according to [L2IODS].

E2E-FUN-L2PP-050
The L2PP shall be compliant with the file naming convention of the Level 2 format definition as used in the operational Ground Segment.

E2E-FUN-L2PP-060
The L2PP shall allow to define its input as belonging to two categories: either mandatory or optional, in line with the specification in the [L2ATBD]. A missing mandatory input shall result in "No

processing" while a missing optional input shall result in "Degraded processing" and marked in the L2 product by means of a quality flag specified in the [L2ATBD].

Note: This requirement defines the functional aspects of the L2PP orchestration. The definition of optional input is specified in the [L2ATBD].

E2E-FUN-L2PP-070
The L2PP shall be able to directly ingest any necessary AUX data (e.g., geophysical data, Meteo data, orbit and attitude data, etc) in the same format these would be available in the actual ground segment.

Note: As consequence it shall be possible to move the L2PP SW in any location within the filesystem while maintaining all its functionalities and its ability to execute.

E2E-FUN-L2PP-090
The L2PP shall allow the writing of intermediate breakpoint data (BRK), e.g., between processing steps/internal algorithms, etc, as defined during the design and development phase.

E2E-FUN-L2PP-100
A user-enabled flag shall control generation of BRK within L2PP.

E2E-FUN-L2PP-110
The L2PP shall be a scalable w.r.t. use of multiprocessors/multicore computer by making use of a multi-threading architecture.

E2E-FUN-L2PP-120
The L2PP shall be capable of running multiple (independent) instances simultaneously on the same machine.

E2E-FUN-L2PP-130
The L2PP shall make no assumptions on the machine on which it is running, the directory it is running in, the directories where input and output files are expected, the memory addresses that his code or data are loaded into.

E2E-FUN-L2PP-140
The L2PP shall be capable of interfacing with the simulation framework openSF [OPENSF].

E2E-FUN-L2PP-150
The L2PP shall be able to work in 2 modes:

- compatible with invocation as per openSF/generic E2E ICD [E2EGICD] when used within the E2ES infrastructure
- standalone use (user invocation by command line)

E2E-FUN-L2PP-160
In the standalone mode, the L2PP shall be self-orchestrating including the selection and ingestion of the Level 1 and applicable auxiliary data (i.e., not requiring any external generated job order).

Note: This functionality allows to select a (time) subset of the data present at its input to support faster processing of time segment of specific interest.

E2E-FUN-L2PP-170
In the standalone mode, while executing, the L2PP shall be data-driven i.e., shall iteratively perform the L1 to L2 processing based on automatically detection of process-able data in an input directory (i.e., no repeated manual invocation from command line needed).

Note: This requirement supports the SW architectural aspects to support any needed persistence across data sets that is required for aggregation and accumulation.

E2E-FUN-L2PP-180
In standalone mode, the L2PP shall be able to resume the processing taking as input the intermediate L2 data products (e.g., starting from L2a or from L2b).

E2E-FUN-L2PP-190
The L2PP shall not require any user interaction therefore allowing automated unattended batch processing.

E2E-FUN-L2PP-200
The L2PP shall be able handle changes in the <MISSION-X> operational baseline without requiring recompilation, i.e. it shall not assume a hard-coded sequence of operating modes. Any such sequence shall be implemented if needed as dynamic configuration items.

Note: This only applies to instrument modes and not to the processing baseline.

E2E-FUN-L2PP-210
The L2PP shall have the capability to limit the data processing to a time range that can be specified using run-time configuration.

Note: This functionality allows to select a subset of data present at its input to support faster processing of a specific time segment.

E2E-FUN-L2PP-230
The L2PP shall be able perform aggregate/accumulate calculations that span multiple measurements/time intervals/geographical location e.g. averaging of several measurements on a

sliding window or determining a median across multiple measurements or areas as required by the [L2ATBD].

Note: The logic and the algorithm to interpret, compute and generate data quality flags is defined in [L2ATBD].

E2E-FUN-L2PP-240
The L2PP shall be designed to allow easy replacement, additions or removal of algorithms without or with minimal changing of the architecture or structure of the L2PP software, e.g. by use of dynamic libraries.

E2E-FUN-L2PP-250
The internal and external L2PP data flow and interfaces shall be representative of the high level data flow related to <MISSION-X> of the data processing within the operational Ground Segment as per Fig. 3-8.

E2E-FUN-L2PP-270
The L2PP shall be able to calculate/estimate the uncertainty (random + systematic) of the generated Level 2 product.

E2E-FUN-L2PP-280
The L2PP shall compute and output all parameters needed (e.g. by the PAM or by an offline Monitoring Facility) to evaluate the geophysical mission performance as required by the [MRD].

E2E-FUN-L2PP-290
The L2PP shall compute and include in the output products, data quality flags that account for both the L2 processing algorithm conditions (e.g. numerical convergence) as well as flags contained within the L1 Product.

Note: This requirement ensures that real values and errors included in the simulation are not known to the L2PP which receives its inputs only via Level 1 products and on-ground AUX files.

E2E-FUN-L2PP-300
The L2PP shall be able to identify, process (where possible as per [L2ATBD]) and flag degradation or incompleteness in the input data including:

- Time and spatial gaps (when applicable)
- L1 values out of nominal range or unrealistic from geophysical point of view

Note: This requirement supports the parallelisation of the processing.

E2E-FUN-L2PP-310
The L2PP shall be able to generate the L2 products even in case of degraded inputs (when this is possible as per [L2ATBD]), and in such a case it shall flag output as degraded quality.

E2E-FUN-L2PP-320
The L2PP shall flag and report any degraded output product in the metadata part of the same products.

E2E-FUN-L2PP-330
The presence of invalid data (i.e. data that cannot be processed) in the L2PP input files shall not prevent the processing of the remaining valid data within the input files.

E2E-FUN-L2PP-340
The L2PP software shall rely for its processing only on internal algorithm and only on data available within its input data files i.e. shall not interface with the SGM, ISM and GM used in the simulation part of the E2ES.

E2E-FUN-L2PP-350
The L2PP processing scheme shall avoid, to the maximum extent possible, dependencies between days, orbits or geographical locations to allow out-of-sequence processing; if unavoidable, the maximum effort should be made to avoid the need of sequential processing.

E2E-FUN-L2PP-360
The L2PP numerical implementation of the algorithm describe in the [L2ATBD] shall be documented in a "Level-2 Detailed Processing Model" DPM document which shall specify the detailed processing steps required to generate Level-2 products.

E2E-FUN-L2PP-370
The L2PP shall also provide the necessary inputs to the PAM to perform the end-to-end comparison of retrieved physical quantities with the simulated quantities input to the SGM.

6.8 Performance Assessment Module (PAM) Functional Requirements

E2E-FUN-PAM-010
The PAM shall compute the necessary performance quantifiers to demonstrate compliance to the <MISSION-X> [SSRD] and the <MISSION-X> mission objectives specified in the [MRD].

E2E-FUN-PAM-020
The PAM shall assess the instrument and L1PP performance by comparing the calculated L1PP outputs with the instrument stimuli generated by the Scene Generator Module.

E2E-FUN-PAM-030
The PAM shall assess the end-to-end mission performance by comparing the calculated L2PP outputs with the geophysical model used as input to the Scene Generator Module.

E2E-FUN-PAM-040
The PAM shall support the visualisation of intermediate breakpoints.

E2E-FUN-PAM-050
The PAM shall assess the L1PP-derived calibration by comparing the online calibration data (CAL) with the instrument parameters (AUX) used during the simulation by the ISM.

E2E-FUN-PAM-060
The PAM shall allow the comparison of E2E datasets with reference datasets acquired through external means (e.g., on-ground measurements and/or data from other operational satellite instruments).

E2E-FUN-PAM-070
The PAM shall allow the characterization of the end-to-end chain behavior and assessment of the <MISSION-X> sensitivity (e.g., different AUX, design, calibration, and characterization data files) to different physical input scenarios.

E2E-FUN-PAM-080
The PAM shall allow the characterization of the end-to-end chain behavior and assessment of <MISSION-X> sensitivity to different instrument configurations and all operating modes.

E2E-FUN-PAM-090
The PAM shall allow the characterization of the end-to-end chain behavior with respect to errors and performance with respect to errors (e.g., sensitivity of retrieved L1 or L2 to instrument errors or to error in the needed AUX data).

E2E-FUN-PAM-100
The PAM shall produce statistics and accumulated results in support of requirement verification.

E2E-FUN-PAM-110
The PAM shall perform consistency checks on all its inputs, both internal and across results.

E2E-FUN-PAM-120
The PAM modules generating static/off-line CAL and AUX product shall be able to operate in standalone mode, i.e., independently from the other modules making up the Performance Assessment Module and triggered both automatically and manually.

E2E-FUN-PAM-130
Seasonal and long-term dependencies shall be reflected in the calibration data sets (CAL/AUX), which are generated by the PAM as input for the L1PP.

E2E-FUN-PAM-140
The PAM shall be able to support both pre-launch (e.g., ingestion of data produced during AIV and support of its modes) and in-flight activities.

E2E-FUN-PAM-150
The PAM shall include tools to support:

- Modification of instrument parameters (e.g., AUX files)
- Modification of calibration algorithm parameters

E2E-FUN-PAM-160
The PAM shall be able to compute TBD performance and calibration metrics, e.g. geolocation error, relative and absolute error maps at L1 and L2, Instrument Response parameters, etc.

E2E-FUN-PAM-170
The PAM shall be able to produce comparison output in graphical form (line, histogram, etc.) in PNG and PDF vector format as well as in tabulated one (e.g. CSV).

E2E-FUN-PAM-180
The PAM shall be able to produce summary reports in a user-friendly format (e.g. PDF, web page, XML + Stylesheet).

E2E-FUN-PAM-190
It shall be possible for the user to superimpose on the plots from E2E-HMI-GEN-070 (display of runtime errors and out-of-limit conditions) plots representing the expected performance, in order to allow a direct comparison between them.

E2E-FUN-PAM-200
It shall be possible for the user to zoom in and out on the plots.

E2E-FUN-PAM-210
It shall be possible for the user to superimpose the measured performance parameters of two different evaluation periods (neither necessarily consecutive nor equal length), to allow a direct comparison between them.

E2E-FUN-PAM-220
The PAM shall be able to invoke user-developed compiled scripts (e.g., C/C++, Python, Matlab) that generate as output a plot or report (e.g., statistical analysis) in a standard format (e.g. PDF and CSV) and optionally display it.

E2E-FUN-PAM-230
The PAM shall be able to support the assessment of the performance related with potential synergy at instrument level and/or L1PP.

E2E-FUN-PAM-240
The PAM shall enable the prototype of generation of any static/off-line CAL product (used as AUX in input to the L1PP) according to the [L1ATBD] and [L1IODS].

E2E-FUN-PAM-250
The PAM modules generating static/off-line CAL and AUX product shall be able to operate in standalone mode, i.e. independently from the other modules making up the Performance Assessment Module and triggered both automatically and manually.

E2E-FUN-PAM-260
Seasonal and long-term dependencies shall be reflected in the calibration data sets (CAL/AUX), which are generated by the PAM as input for the L1PP.

7 OPERATIONAL REQUIREMENTS

This section specifies the operational requirements of the E2ES.

7.1 General operational requirements

E2E-OPE-GEN-010
The E2ES shall be capable of simulating a scenario of at least X days.

Note: Generation of long-term scenario are required for TDS generation to be used for Ground Segment validation.

E2E-OPE-GEN-020
The E2ES shall be capable of executing unattended for at least 7 days.

E2E-OPE-GEN-030
The user shall be able to stop or pause and restart the execution of a simulation.

Note: This functionality is supported directly by openSF [OPENSF] as according to [E2EGICD].

E2E-OPE-GEN-040
The user shall be able to use real instrument (generated on-ground and in-orbit) as well as simulated input data to the E2ES.

E2E-OPE-GEN-050
The E2ES shall consider for each module the provision of a bypass, static, or low-accuracy execution mode for TBD E2ES functions during operation, so that rapid (albeit less accurate or partially incomplete) results can be obtained.

E2E-OPE-GEN-060
Each simulation initialisation, as defined by the combination of all the user inputs, shall be stored in configuration files which can be loaded to easily initialise an E2ES run.

E2E-OPE-GEN-070
The parameters specifying user control over the E2ES functions and operation shall provide complete control over the selection of the E2ES functions and over the initialisation, execution, and termination of the E2ES operation. They shall include as a minimum the following control parameters and data:

- Flags that determine the system to be analysed: spacecraft identifier, instrument version
- User inputs that determine which functional configuration of the E2ES, which E2ES version/release number, which external interfaces, the user wants to use during E2ES operation

- User inputs that determine which elementary functions of E2ES constituent modules the user wants to include or exclude during the E2ES operation
- User inputs that determine the type, location in scenario/module, and severity of errors or non-nominal conditions the user wants to include or exclude during the E2ES operation
- User inputs that determine which elementary functions of the support functions user wants to include or exclude during the E2ES operation
- Parameters defining the conditions under which the E2ES shall terminate its operation (e.g., time, number of samples, user interrupt, etc.)

E2E-OPE-GEN-080
For use with the Instrument Data Simulation, flags that determine the nature and extent of the E2ES run shall be defined:

- The number of data
- Orbit: full or subset
- Coverage and location
- The source observation data used
- Unique identifier per measurement
- Oher TBD flags as required
- Parameters that specify the computation of the Level 1 data quality (e.g., number of measurements for absolute and relative error computations)
- Flags which control the extent and content of the support output data i.e., which E2ES variables, function status, parameters, etc., are to be output by the E2ES
- Parameters defining the conditions under which the E2ES shall terminate its operation (e.g., time, number of samples, user interrupt, etc.)

E2E-OPE-GEN-090
Defaults values for all user inputs shall be provided by the E2ES.

E2E-OPE-GEN-100
The user shall have the possibility to define and set tags for the physical/environmental conditions of the simulated scenario and corresponding data. Each source data and scenario shall therefore be associated, within the E2ES, with these identifying tags allowing the User an easy selection of the relevant set of scenario/source observation data.

E2E-OPE-GEN-110
The E2ES shall operate, without loss of computational performance, in case of missing samples/data in the datasets (e.g. due to simulated loss of the data link).

E2E-OPE-GEN-120

The E2ES shall be able to distinguish and manage permissions for access/operations to the simulation and its parameters separately for the following user categories:

- Operator
- Expert operator
- Developer

Note: This requirement is intended to avoid misconfigurations of the E2ES by edition of files such as the calibration database which requires expert knowledge.

8 PERFORMANCE REQUIREMENTS

E2E-PER-GEN-010
The E2ES and its modules shall be able to satisfy their performance requirements on a x86 computer with at least 8 cores, 32 GB RAM, and 1TB SSD storage (TBC at proposal time).

Note: This will drive the need for code optimisation, parallelization, machine dimensioning and possible redefinition of scene/ADF resolution to achieve a processable baseline with the required timeliness.

E2E-PER-GEN-020
The E2ES shall be able to generate space segment simulated data (RAW) corresponding to 1 hour of observation in less than 1 hour i.e., in real time.

Note: This requirement applies to full-accuracy simulation, not the low-accuracy mode contemplated in E2E-OPE-GEN-020 and is targeted to allow execution of sensitivity analysis in a reasonable time (e.g., iterative generation of the same simulated orbit with 10 different values of a parameter in less than 2 days).

E2E-PER-GEN-030
The L1PP shall be able to convert 1 hour of simulated data (RAW) into Level-0 data in less than 5 minutes.

E2E-PER-GEN-040
The L1PP shall be able to completely process 1 hour of simulated data (RAW), of any instrument operating mode, to Level-1 in less than 15 minutes i.e., four times faster than real time.

Note: This requirement is sized to the generation of long data set from the Ground Segment testing as well as to support commissioning where data is daily manually processed and must be analysed within the same day.

E2E-PER-GEN-050
The L1PP shall be able to process Level-0 data of any instrument operating mode up to Level-1 four times faster than real time.

Note: The processing of RAW to L0 within the L1PP is considered negligible.

E2E-PER-GEN-060
The L2PP shall be able to process Level 1 data (corresponding to 1 hour of observation) in less than 30 minutes i.e., 2 times faster than real time.

E2E-PER-GEN-070
The PAM shall be able to perform all its processing and produce the performance assessment reports and plots for 1 hour of simulated observation in less 10 minutes.

E2E-PER-GEN-080
The E2ES shall be able to simulate scenes on which the upper limit size shall not be limited by the software design but rather by the available computational resources.

9 INTERFACE REQUIREMENTS

The following requirements address elements of the interface which is relevant to the processing in the Ground Segment.

9.1 General Interfaces Requirements

E2E-INT-GEN-010
The E2ES and any of its module shall be able to ingest the relevant payload, platform, auxiliary and calibration data files in the Ground Segment format and as defined in the [L1IODS] and [L2IODS] (e.g., L0 science files, L0 ancillary/HKTM files, Instrument characterisation AUX files, Orbit Files, Level 1 Files, CAL files, etc).

E2E-INT-GEN-020
The E2ES architecture shall provide the possibility to have distinct configurations per:
- instrument: possibility to have different configurations for each instrument
- product type: possibility to have different configurations for each product type
- observation timeline: the timeline or sequence of modes and measurements intended for the mission simulation

E2E-INT-GEN-030
The definition of content and format of AUX and CAL files containing parameters derived by: Space Segment design, ground characterisation, ground and in-flight calibration, both on-line and offline (e.g. from CKD, CCDB) shall be modular with parameters segregated in different files to reflect the relevant domain, algorithm using it as well as the expected update periodicity and category of parameter e.g. alignment parameters separated from sensor parameters, rapidly varying parameters separated from slowly varying (off-line) parameters.

Note: It is not desirable to have all configuration parameters contained in a single monolithic block as it prevents user visibility at file level of different simulation conditions as well as an efficient orchestration.

E2E-INT-GEN-040
The file-naming of AUX and CAL files derived from CCDB and CKD in different conditions (different pre-flight AIV characterisation campaign, different in-flight orbit, etc) shall be unique and according

to [EOFFSTD] in particular with respect to validity and time elements, such as to allow the selection of the relevant one for the specific simulation or L1 processing only based on the filename.

E2E-INT-GEN-050
The E2ES and any of its module shall generate any output data file in the Ground Segment format and as defined in the [L1IODS] and [L2IODS], e.g. L0 files, CAL files, L1 and L2 Files.

E2E-INT-GEN-060
The design of the E2ES shall incorporate flexibility to adapt to changes in the input data space packet definition within any RAW or Level 0 Data files (e.g. avoid hard-coding packet/parameters structure).

E2E-INT-GEN-070
The E2ES modules' input and output data shall be in a common format which is open and cross-platform, readable with freely available tools, or be convertible to such a format without loss of information.

10 VERIFICATION, VALIDATION AND SYSTEM INTEGRATION

E2E-VVP-GEN-010
The E2ES development shall follow the verification and validation processes as defined by the ECSS standard ECSS-E-ST-10-02C.

E2E-VVP-GEN-020
The E2ES shall be verified using a set of reference test cases.

E2E-VVP-GEN-030
Each module and the overall software shall be (scientifically) validated to check that the requirements for the specific intended use have been fulfilled.

Note: This requirement pertains to the independent scientific validation of the scientific algorithms in the E2ES with no reference to the ECSS SW validation. It applies primarily to SGM and L2PP.

E2E-VVP-GEN-040
The E2ES shall be delivered for integration together with a confidence test that includes configuration data, input data, reference data and ancillary data (if required). The aim is to run the confidence tests on the integration machine (before proceeding with the integration itself) and compare the obtained output data with the provided reference data.

E2E-VVP-GEN-050
The E2ES test data for each software version shall be stored and kept available for customer inspection, including test drivers, configuration, input data, output data and any other additional tests results (plots, statistics).

E2E-VVP-GEN-060
The E2ES verification and validation documentation shall cover the full specifications by tracing each test to one or more requirements.

E2E-VVP-GEN-070
Automatic testing capability shall be provided, in order to automatize testing for subsequent versions of the SW and for regression testing.

10.1 Level 1 Prototype Processor (L1PP) Verification and Validation

E2E-VVP-L1PP-010
The L1PP verification shall be performed against the requirements that are specified in the present document as well as the requirements flown down from the [SSRD].

E2E-VVP-L1PP-020
The L1PP validation shall be performed against the [L1ATBD] or dedicated L1 validation document.

Note: The L1 Validation of ATBD can be performed e.g., using existing tools that perform similar processing.

E2E-VVP-L1PP-030
The verification/validation of the L1PP shall be performed using the following types of data:

- instrument simulator data
- data obtained during ground AIT/AIV activities and the calibration campaign of the space segment

10.2 Level 2 Prototype Processor (L2PP) Verification and Validation

E2E-VVP-L2PP-010
The L2PP verification shall be performed against the requirements that are specified in the present document as well as of the relevant one in the [MRD].

E2E-VVP-L2PP-020
The L2PP validation shall be performed against the [L2ATBD] or dedicated L2 validation document.

Note 1: The L1 Validation of ATBD can be performed e.g., using existing tools that perform similar processing.
Note 2: This requirement addresses algorithmic validation and not the geophysical one of the products.

E2E-VVP-L2PP-030
The validation of the L2PP shall be performed using the following types of data:
- Simulated Level 1 data as generated by the L1PP
- Data obtained from the Level 1 Operational Processor

11 HUMAN-MACHINE INTERFACE (HMI) REQUIREMENTS

NB. A number of requirements in this section are satisfied automatically when using the openSF framework [OPENSF].

E2E-HMI-GEN-010
It shall be possible to perform all operation of the E2ES by mean of an HMI.

E2E-HMI-GEN-020
The E2ES shall be capable of interfacing with the simulation framework openSF [OPENSF].

E2E-HMI-GEN-030
The E2ES HMI for each Module/Instrument on <MISSION-X> shall use the same concept, widgets and graphical framework and look-and-feel.

E2E-HMI-GEN-040
The E2ES HMI shall:
- Manage (i.e., create, modify, view, delete, store, retrieve) all E2ES input/output data
- Display on the screen relevant information on E2ES processes
- Allow the user to monitor in real-time the operation of the E2ES
- Allow the user to control in real-time the operation of the E2ES

In accordance with the detailed requirements specified below

E2E-HMI-GEN-050
The E2ES HMI shall allow the user to manage and display all the E2ES output data and the Performance Assessment Module outputs.

E2E-HMI-GEN-060
The E2ES HMI shall provide the user with the real-time display of warning and error messages. The message shall identify the nature of the error and shall be color coded (e.g. warning orange, error red, informative blue).

E2E-HMI-GEN-070
The E2ES HMI shall automatically display all run-time errors or out-of-limit conditions.

E2E-HMI-GEN-080
The HMI shall be readable at a distance of 50 cm.

E2E-HMI-GEN-090
It shall be possible to use and display the HMI locally or remotely.

E2E-HMI-GEN-100
It shall be possible to place separate HMI windows on separate video displays (screens) and to move them from one to the other with the mouse (e.g., log window, scenario window, error window, execution window, etc.).

E2E-HMI-GEN-110
The HMI shall include a progress/status bar indicating the completion status of the simulation/processing as well as the remaining estimated time.

12 REFERENCE TEST DATA SET (TDS) REQUIREMENTS

E2E-TDS-GEN-010
A reference test data set generated with the E2ES shall be delivered (TDS), to be used for validation of the L1PP/L2PP and well as a reference for the L2OP and ground segment.

E2E-TDS-GEN-020
The TDS shall include examples of all modes/data types/calibrations of <MISSION-X> including nominal and non-nominal cases.

E2E-TDS-GEN-030
The TDS shall comprise:

- Infrastructure (openSF) related Items, i.e., configuration files, scenarios, scripts, and database
- Inputs/Outputs of all E2ES modules including Calibration Products
- Any auxiliary software tool needed to reproduce the TDS

E2E-TDS-GEN-040
Each TDS delivery shall be accompanied by a release note specifying at least:
- detailed data set content
- timeline
- scenarios covered

E2E-TDS-GEN-050
The TDS shall
- be coherent (e.g., outputs of the preceding stage are the inputs of the current stage)
- have a timeline coverage of at least two days of simulated mission
- be consistent with the reference instrument timeline generated by Mission Planning for overall validation purpose
- ensure full validation of the interfaces, calibration, and retrieval algorithms

13 INPUT TO THE SOW

13.1 Deliverables

13.1.1 *Software and data*

The following software deliverables shall be provided:

D-SW-01 E2ES executable form (including L1PP and L2PP)
D-SW-02 E2ES Source Code
D-SW-03 L1PP (standalone) in executable and source code
D-SW-04 L2PP (standalone) in executable and source code

D-SW-05 Test Data Set defined according to a 48-h mission scenario to be agreed with ESA.

NB. Each delivery of the E2ES (D-SW-01/02) shall also include a stand-alone installable L1PP and L2PP Module (D-SW-03/04)

Any source code shall include the necessary script and build tools (makefiles, etc) to generate the running executables.

These deliverables shall be provided to the Agency without any restrictions and IPR for use, re-use, modification, adaptation, or redistribution.

13.1.2 *Documentation*

The following documentation shall be provided by the contractor, without any restrictions on (re)distribution. It consists of a tailored set of the ECSS documentation plus documentation which is specific to Processors development.

- Software Development Plan (SDP) (including Configuration and Product Assurance Plan aspects)
- Software Design Document (SDD) (at level of architectural design and overall processing logic and dataflow.)
- Software User Manual (SUM)
- Software Verification Plan (SVerP)
- Software Verification Report (SVR) (as needed following Test Execution)
- L1 Algorithm Theoretical Basis Document (ATBD)
- L2 Algorithm Theoretical Basis Document (ATBD)

- L1 Detailed Processing Model (DPM) (at level of detailed design of the numerical implementation of algorithm and orchestration including e.g., pseudo-code and details of all functions I/O, data software types, internal branching, error handling, etc)
- L2 Detailed Processing Model (DPM) (at level of detailed design of the numerical implementation of algorithm and orchestration including e.g., pseudo-code and details of all functions I/O, data software types, internal branching, error handling, etc)
- Input / Output Data Specification (IODS L1) in accordance with the L1PP level 1 format definition description and the calibration and characterisation database
- Input / Output Data Specification (IODS L2) in accordance with the L2PP level 2 format definition description and the AUX data definition.
- Test Data Set Description Document (TDSD)

Any call in the E2ES source code to COTS and third-party libraries, including the EO mission CFI library, shall be documented in the ATBD and DPM with associated inputs and configuration parameters.

Formal deliveries shall include documentation, software, installation packages and all associated datasets.

13.1.3 *Delivery Plan*

13.1.3.1 E2ES

As a minimum five deliveries of the E2ES shall be provided:

- A preliminary version for the E2ES PDR including full infrastructure, representative data flow and orchestration and stub for its components and preliminary formats
- A version for Instrument Engineering Qualification Model EM/EQM performance verification and calibration, to be delivered at the start of the first performance measurement campaign; including preliminary PAM
- A version for Instrument Protoflight Model (PFM) / FM2 performance verification and calibration, to be delivered at the start of the PFM / FM2 performance measurement campaign which is fully representative format of all I/O data: RAW, L0, L1 Product, L2 product and
- A version to support ground segment test data generation (fully representative simulation and formats)
- A version to support ground segment evolution of L1PP (fully representative processing at L1)
- A version to support ground segment evolution of L2PP (fully representative processing at L2)

During any of the delivery it shall be possible for ESA to request an update to version of COTS (EOCFI, openSF, etc), OS and Compiler of any component at no cost.

13.1.3.2 L1PP

As a minimum, and in addition to the three deliveries as part of the E2ES above, an additional three formal (separate) deliveries of the L1 Processor Prototype shall be provided:

- A pre-launch version, to be delivered at launch – 12 months
- A pre-launch version, to be delivered at launch – 1 months (TBC)
- A post-launch version, to be delivered before the end of the in-orbit commissioning phase

The Contractor shall also provide the cost of additional (optional) L1PP deliveries should that be requested by ESA.

13.1.3.3 L2PP

As a minimum, and in addition to the three deliveries as part of the E2ES above, an additional three formal (separate) deliveries of the L2 Processor Prototype shall be provided:

- A pre-launch version, to be delivered at launch – 6 months
- A pre-launch version, to be delivered at launch – 1 months (TBC)
- A post-launch version, to be delivered before the end of the in-orbit commissioning phase + 1 month

The Contractor shall also provide the cost of additional (optional) L2PP deliveries should that be requested by ESA.

13.1.3.4 TDS

TDS delivery number shall coincide with L1PP deliveries (3 + 3); however, it shall also be possible to have (in agreement with ESA) TDS deliveries at separate times (e.g., to cope with delay in L1PP development).

13.2 Maintenance and Support

13.2.1 *Support task for commissioning*

The Contractor shall provide TBD manpower support to ESA during the execution of the commissioning. This support includes use of the E2ES and of the L1PP and L2PP on simulated and real data and contributions to analysis of discrepancies.

13.2.2 *Support task for cross validation*

The Contractor shall provide TBD manpower support to ESA for the execution of a cross validation campaign between L1PP/L2PP and the corresponding Operational processor. This support includes definition of appropriate test scenarios, generation of reference test data, and contributions to analysis of discrepancies.

13.3 Other

13.3.1 *Licensing*

Input to GCC and Clause 42 will include licensing of source code and binaries according to ESA Community Licence - Permissive for any agency use (i.e. not limited to <MISSION-X>) [ESACLP]

End of Document