White Paper on End-to-End mission performance simulation chains

PE-RP-ESA-GS-0754

Noordwijk, 30-11-2025

V 3

Summary

The present document captures lessons learned in the procurement and technical development of End-to-End mission performance simulation chains for Earth Observation. These tool chains include simulation of both the target scene and the space segment, as well as the Level 1 and Level 2 processing. It provides recommendations, templates and references to tools implementing common practices addressing the lessons learned.

Noordwijk, 30-11-2025

1 Purpose of the Document

The assessment of an ESA Earth Observation mission commences Phase 0/A/B1 when the mission concept is being studied as part of the Future Missions definition process. Crucial elements supporting decision making are the execution of End-to-End performance simulations up to Level 2 to estimate how well mission objectives in terms of intensive and extensive properties can be met by the proposed payloads and algorithms.

Performance engineering includes the assessment of the entire system (space and ground) based on subsystem and component level information. In this process, the End-to-End performance simulations are initially executed to define system requirements to meet mission objectives and later in phase B2/C/D to verify the expected system performance during the mission development and in preparation for in-flight verification.

The present document gives an overview of procurement and particular technical aspects of these mission performance software simulation chains, based on the common practice gained in numerous present and past Earth Observation projects (EOP-F and EOP-P). In doing so, lessons learned, recommended practices and templates especially applicable for the mission phase B2/C/D/E1 are presented.

1.1 Authors

The procurement of End-to-End mission performance simulation chains in ESA is currently implemented in diverse fashion within the different ESA projects. Under the coordination of EOP-PEP, performance and processor engineers from EOP-PEP and other divisions have been exchanging experience from EOP projects, e.g., ERS, ENVISAT, CRYOSAT, SMOS, SWARM, AEOLUS, FLEX, Biomass, Forum, SKIM, Harmony, S2, S3, S4, S5P, S5, Premier, and CarbonSat and more recently on Copernicus expansion missions on a regular basis for the benefit of each project. The meetings have been the platform for generating the present document. The members of this group (past and present) are:

R. Koopman (1), P. Jurado (formerly 2) , T. Kanitz(1), R. Mecozzi (formerly 1), B. Andela (formerly 1), M. Zundo (1), O. Le Rille (1), I. Avruch (formerly 1), S. Mattia (1), G. Paolini (3), M. Fornari (4), C.Zelli(1), L. De Cinti (1), Y. Aoun (formerly 1), A.Garcia (1), M.Langhe (1), D. Lajas (1), F.Marnas(formerly 1), G. Obrecht(1)

(1)= EOP-PEP (2) = EOP-FM, (3)=EOP-PRS, (4)=EOP-PYP

1.2 Reference documents

Ref	Title	Version
[EOFFST]	Earth Observation File Format Standard, PE-TN-ESA-GS-0001. https://eop-cfi.esa.int/index.php/docs-and-mission-data/system-support-docs	3.0.2
[MSFFS]	Earth Observation Mission Software File Format Specification, PE-ID-ESA-GS-584, https://eop-cfi.esa.int/index.php/docs-and-mission-data/system-support-docs	1.9
[GRAWL0]	EO generic RAW and L0 specification, PE-TN-ESA-GS-586, https://eop-cfi.esa.int/index.php/docs-and-mission-data/system-support-docs	1.2
[RARCH]	EO E2ES Reference Architecture, ARCHEO-TN-002. https://eop-cfi.esa.int/index.php/docs-and-mission-data/e2e-perf-simulator-docs	3.2
[GE2ERD]	Generic End-to-End Simulator and L1/L2 Processor Req. Document, PE-TN-ESA-GS-402, https://eop-cfi.esa.int/index.php/docs-and-mission-data/e2e-perf-simulator-docs	2.0
[CCDBG]	Guidelines for (S)CCDB mapping to GS files https://eop-cfi.esa.int/index.php/docs-and-mission-data/e2e-perf-simulator-docs	1.2
[GENICD]	Generic E2E Simulator Interface Control Document, PE-ID-ESA-GS-0464, https://eop-cfi.esa.int/index.php/docs-and-mission-data/e2e-perf-simulator-docs	1.4.2
[BIBLOS]	BIBLOS Technical Specification and SW libraries (*) https://e2e-performance-simulators.gmv.com/	3
[EOCFI]	Earth Observation Mission Software CFI, http://eop-cfi.esa.int	4.29
[OPENSF]	OpenSF software and Documentation, http://eop-cfi.esa.int/index.php/opensf	4.6
[DFDL4S]	DFDL4S binary R/W library https://eop-cfi.esa.int/index.php/applications/dfdl4s	2.1.1
[ECL]	ESA Community licence, https://essr.esa.int/license/list	2.4

*Freely available on request by EOP-PEP

2 Contents

3 Introduction to End-to-End simulator (E2ES)

End-to-End mission performance simulators (here after referred as **E2ES**) are chains of software tools developed to simulate the generation of the geophysical observables target scene, the whole observation process from space including orbit, pointing and guidance, the instruments, the on-board data generation and the processing of the outputs of these instruments to retrieve the estimate of the original observables.
Several activities have been run in the past by EOP and TEC to define reference architecture/nomenclature for these software systems ([RARCH],[GE2ERD]) and this document will make use of the convention and approach defined there.

The basic architecture of a complete E2ES is given in Figure 1. The figure shows among others the data flow between Geometry Module (GM), a Scene Generator (SGM), the Instrument Simulator (ISM), a Level 1 processor (L1 PM), a Level 2 processor (L2 PM), and a Performance Assessment (or Evaluation) Module (PAM). Details can be found in [GE2ERD]

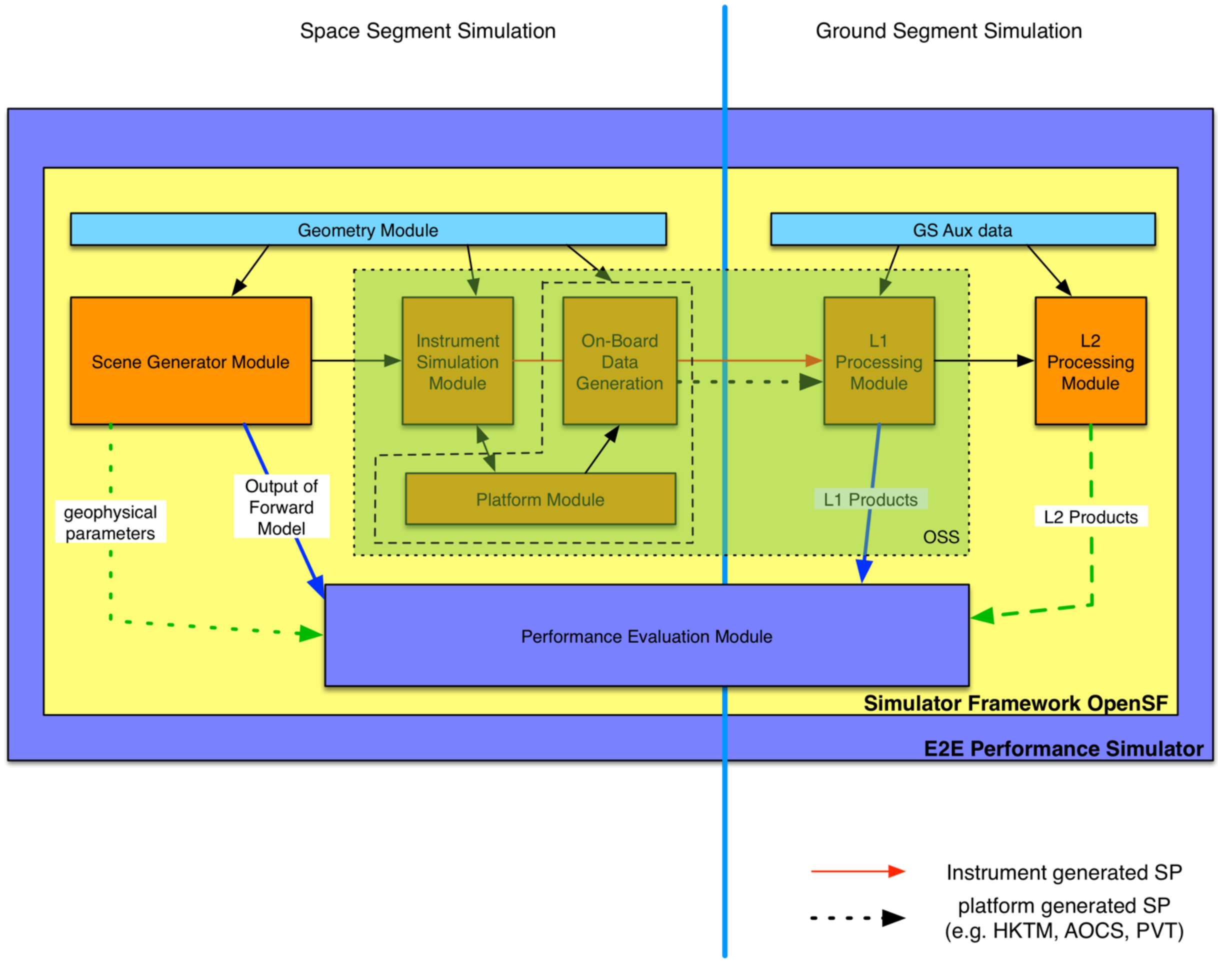

Figure 1 Generic layout of an End-to-End simulator with Scene Generator (SG), Instrument Simulator (IS), Level 1 processor (L1), Level 2 processor (L2), and Performance Assessment Model (PAM)

3.1 Purpose of the E2ES in the Project life cycle

3.1.1 Phase 0/A/B1:

End-to-End mission performance simulation chains (E2ES) are built to assess mission requirements and concepts, and support generation of the system requirements. E2ES are developed under the responsibility of the Phase 0/A/B1 Study Manager and with support from the EOP-PS Mission Scientist. As the emphasis in this phase is on the feasibility of the L2 retrieval the E2ES is customarily procured with the Observing System Simulator (OSS) functionalities from Fig. 1 as a single component that simply adds errors on the output of the Scene Generation. The OSS is later (Phase B1 or B2) expanded into detailed and separate modelling of ISM and of L1PP often with 2 different consortia. As mentioned in the introduction, for this phase best practices are well developed and systematically applied in EOP-F.

3.1.2 Phase B2/C/D/E1

In this phase, the purpose of the simulation chain is to support the development and verification of detailed L1 and L2 algorithms, the Ground Segment and its processing components, as well as the satellite (instrument and payload) itself at Prime and ESA side. It must adapt flexibly to frequent algorithm evolution ensuring ready data generation and quick implementation turnaround. The simulation chain can initially be based on reuse of the End-to-End simulator of phase 0/A(/B1).

The major deliverables needed by the Ground Segment in this phase are:

a) The actual E2ES software toolchain;
b) the algorithm and product specification (ATBD, DPM, IODS);
c) the set of auxiliary data files (e.g. instrument characterisation parameters, DEMs);
d) the Test Data Set for verification and validation.

The L1 activities are performed under the responsibility of the EOP-P Project while the L2 are under the responsibility of EOP-S ideally in cooperation with EOP-P for the SW engineering aspects.

In this Phase it is important to maintain aligned and compatible both the L1 and L2 elements since they are defined by different entities (see section 4.2.12 on the organisational aspects) and must eventually be integrated.

3.1.3 Phase E2

At the end of the Commissioning Phase (E1), the components and the responsibility for their maintenance are handed over to the Mission Manager. While in this phase the operational processors and all Ground Segment components are fully available the End-to-End tool chain remains available to supports algorithm and processor maintenance and evolution. Considering that real measurements become the main source of input to the processors. The simulation components in the End-to-End toolchain are nevertheless available to support specific anomaly investigations (e.g. simulation of new modes of operation when nominal mode can no longer be used due to instrument degradation).

3.2 Top-level Generic Architecture of the E2E Simulator

As shown in Figure 1, the architecture is split between the *Simulation domain* (target and space segment) on the left and the *Ground Segment domain* (data processing part) on the right.

In the Simulation domain, the Scene Generator (SG) generates the TOA stimuli (e.g. radiance, scattering coefficients, etc) from a given geophysical target considering the instrument sampling and geometry derived by the Geometry Module using a simulated orbit and attitude.
The TOA stimuli from the target are then fed into the Instrument Simulator (IS) which simulates the measurement process including all necessary sampling, errors, degradations, etc and generates, together with the On-Board Data Generation and Platform modules, the complete set of Ancillary, Platform and Instrument Source Packets (ISPs) in RAW data format [GRAWL0].

In the Ground Segment domain, the Level 1 Prototype Processor Module (L1PP) ingests the RAW data ISPs, formats them internally as Level 0 files[1] and generates ancillary, measurements and calibration Level-1 data applying the necessary processing.

The Performance Assessment Module (PAM) confronts then input TOA stimuli with the outputs of the L1PP to perform sensitivity analysis and L1 performance verification (so called "*inner*" or Level 1 loop). Finally, the Level 2 Prototype Processor process the Level 1 files and executes the geophysical retrievals producing the Level 2 data files; these are compared with the original geophysical observable target scene by the Performance Assessment Module, allowing performance assessment and sensitivity analysis at Level 2 (also known as "*outer*" or Level 2 loop).

A detailed reference architecture and corresponding generic requirements are presented in detail in [RD1]

3.2.1 The E2ES for Mission and System Requirement Verification

As described above, there are two major loops in the E2ES that support verification at two different levels.

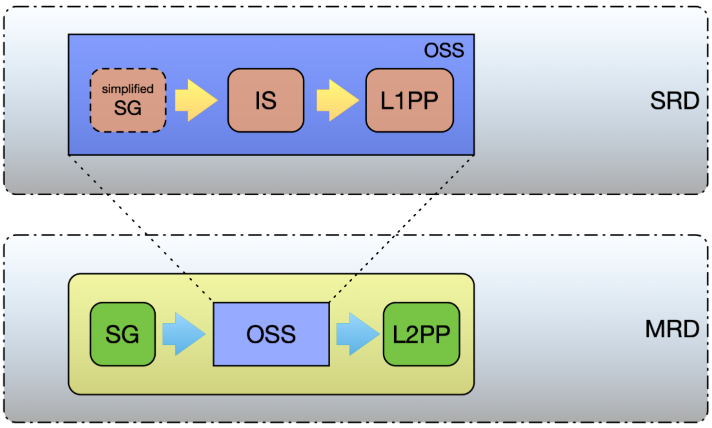

Figure 2 E2EScomponent and verification level

[1] A full functional L0 module is not required/implemented for the purpose the E2ES Mission performance simulator, therefore the interfaces are kept at RAW/ISP level. Internal L0 conversion is a simplified "repackaging" of RAW data as per [GRAWL0]

The outer loop represents the complete E2ES up to Level 2 and is necessary for the verification of the MRD, while the E2ES while the inner loop (when the OSS is specialised in ISM and L1PP) is used for verification at Level 1 of the Satellite requirement (SRD)(see also discussion on 4.2.17)

3.3 Responsibilities in the E2ES chain

For most current EO missions the Space Segment Prime Contractor, and its subcontractors are responsible for the Instrument Simulator and the Level 1 processor and related elements like the Geometry Module and any Offline Calibration Processor, and the ‘inner loop’ of the PAM (comparing input TOA stimuli the Instrument Simulator with the output radiances of the Level 1b processors).

Generally, ESA directly (i.e. independent from the Space Segment Contract) contracts scientific institutions/algorithm developers for the prototyping of the Level 2 Products, for the Scene Generator as well as for the outer loop of the PAM

To achieve an E2ESsystem that can technically and programmatically integrate the L1 and the L2 elements, two approaches are suggested in section 4.2.12

3.3.1 Space Segment prime contractor

The Space Segment prime contractor is often responsible for procuring and developing the instrument simulator and the Level-1 processor, which it uses to demonstrate compliance with Level-1 system requirements. Because Level-1 processing plays a central role in the Ground Segment, the Level-1 algorithms also incorporate the calibrations and parameters needed by the Level-2 processors. These elements are not tied to verification of Space-Segment Level-1 requirements but instead support MRD-level scientific/Level-2 requirements.

Although for some missions ESA directly procured the E2E chain software from dedicated industry (e.g. S5P, SMOS), the usual practice is different: the Level-1 prototype and the instrument simulator are developed under the responsibility of the prime contractor, and these typically become either the operational Level-1 processor or the reference processor. In this approach, the detailed processing model and the test data produced by the prototype define the boundary conditions for the independent development of the operational processor. The intention is that, with this approach, the required instrument knowledge and data are directly available for E2E modelling and processing, while also ensuring coherence between the algorithms used for space-segment requirement verification and those implemented in the ground-segment processor while reducing the manpower needed on the ESA side to supervise Processors and Simulators development.

Considering the critical role of the chain within the Ground Segment, for many recent missions (including those with external partners such as EUMETSAT) this approach has produced undesirable outcomes. An agile software development cycle is fundamentally different from a hardware-procurement model, and this mismatch contributed to several issues: weak commitment, limited flexibility, insufficient understanding of Ground Segment needs, restricted numbers of software and data releases, slow bug-fixing, long turnaround times, and delays driven by dependence on satellite milestones. In addition, priority given to space-segment development and testing reduced the availability of competent manpower, and some deliverables were accepted despite being inadequate, simply to close major satellite milestones.

Furthermore, the inability of this approach to meet Ground Segment needs in a timely manner has consistently created friction with the operational ground segment (whether EOP-G or EUMETSAT) and with the Level-2 scientific algorithm developers and institutions. This has ultimately resulted in higher costs and increased ESA management effort.

R-1.
Procure the E2ES toolchain directly as a dedicated ESA activity under the umbrella of a consortium led by a strong experienced SW house in the domain and including dedicated WPs with the space segment industry to provide required inputs: input to instrument modelling, L1 and calibration approach, ATBDs, characterisation and test data, participation to reviews, pre-allocated support time as well as sufficient number of iterations of those.

3.3.2 ESA Mission scientist

The ESA Mission Scientist works with the Project to ensure that mission objectives (typically defined at Level-2) are met. The Mission Scientist convenes the Mission Advisory Group (MAG) and is the reference point for scientific and geophysical matters. The role also includes providing expertise and support to the Level-2 processor and Scientific Scene Generator developments.
For most missions, the Mission Scientist is also the Technical Officer for the Level-2 processor-prototype procurement. However, this role typically lacks dedicated software expertise, which must therefore be supplied by the Project team.

To maintain coherence between Level-1 and Level-2, ensure proper product interfacing, and guarantee adequate processing performance and software quality for both the Level-2 processor prototype and the Scene Generator, the Mission Scientist often—though not always—manages the contract with support from the Ground Segment engineer or a dedicated Processor Engineer.

R-2.
Ensure that the technical management of the Level 2 software element is always performed jointly by the Mission Scientists (for scientific aspects) and by a dedicated Processor Engineer (for SW aspects) familiar with the overall E2ES toolchain development (including Level 1).

3.3.3 Scientific Algorithms' developers

In the nominal case, the highest-level ESA product of a mission—and the one with greatest visibility to the user community—is the Level-2 product. Typically, ESA contracts scientific algorithm developers or institutions to define and implement the Level-2 prototype algorithms and to support the review of the Level-1 algorithms. These teams are often assisted by software developers to construct the prototype processor.

To verify the full performance chain up to Level-2, the algorithm developers must also generate geophysically realistic scenarios, rather than purely synthetic ones, by developing a dedicated Scene Generator (SGM).

3.3.4 Instrument Performance engineer

The ESA Performance Engineer within the Project team is responsible for instrument performance. The primary role is to oversee the hardware procurement and ensure that the instrument-level requirements verified by the Space Segment prime are properly addressed. The Performance Engineer is involved in E2E development up to Level-1 to guarantee adequate instrument modelling in the Instrument Simulator software and to assess the suitability of the Level-1 algorithms. This role works closely with the Simulator and Processor Engineer, who is responsible for the software engineering and coordination of the procured E2E chain software elements.

3.3.5 Simulator and Processor Engineer

The ESA Simulator and Processor Engineer within the Project team has a strong software background and serves as the focal point for defining, developing, accepting, testing, and using the End-to-End Mission Performance chain and its components. The engineer coordinates the various needs into a coherent set of E2E-chain software requirements that satisfy all stakeholders. Working closely with the Instrument Performance Engineer (Level-1), the Mission Scientist (Level-2), and the Ground Segment Engineer, the role provides guidance on software-development best practices.

The Simulator and Processor Engineer is responsible for overall integration of the E2E chain at Level-1 and Level-2, either internally or through industrial contracts. The role includes deploying the E2E toolchain for internal ESA use and enabling the Performance Engineer and others to operate it. The engineer works with the Ground Segment engineer to ensure that all required deliverables to the operational Ground Segment are available on time and with adequate quality.

He manages anomaly tracking with respect to the E2E-chain developers and oversees delivery of the chain to internal and external stakeholders. The role may cover several E2E chains, for example when multiple instruments are involved, and ensures that consistent, proven technical baselines and practices are applied across developments.

In several Project teams this critical role is not formally assigned, and the duties are instead delegated to the Ground Segment Engineer, the Instrument Performance Engineer, or even outsourced to industry. This has led to inconsistent or insufficiently acceptable deliverable quality.

R-3.
Ensure that the necessary SW expertise is present, and that the Processor Engineer function is formally present in the Project team either as dedicated support or as shared resource.

3.3.6 Ground Segment System engineer

The ESA Ground Segment System Engineer defines the Ground Segment requirements within the Project, establishes the overall GS architecture, and manages the interfaces with the FOS and PDGS (or other external entities) to agree on data and operational interfaces. The engineer defines and implements the overall Ground Segment testing and provides general ground-software expertise within the Project.

When no dedicated Processor Engineer is present, the Ground Segment System Engineer may also cover aspects of the E2E Mission Performance chain. However, this arrangement is not optimal, as the role may lack the hands-on, low-level software expertise required to perform the specialised tasks of a Processor Engineer.

4 Lessons Learned

Every mission needs an End-to-End simulator, and hence significant experience is available within ESA from past developments. For mission phases 0/A/B1 significant harmonisation is already achieved within EOP-FM. Some multi-instrument missions in phase B2/C/D/E1 have demonstrated also the benefit of common practices in specific areas in this mission phase.

4.1 Harmonisation of E2ES in ESA EOP

EOP-PEP has consolidated inputs, experience, and lessons learned from missions already flying or under development, as well as from TEC activities and studies, participation in TEBs and ITTs, and technical-progress meetings. Based on this, it has compiled a set of general procurement practices and technical and architectural requirements recommended for cross-cutting application to all upcoming missions.

To support this effort, EOP-PEP has also taken over the operationalisation of the openSF software-simulation framework, originally developed in EOP-F. This ensures that internal and industrial support is available to all missions, provides seamless interfacing with the Mission Software CFI libraries for orbit and attitude calculations, and maintains end-to-end architectural coherence.

The relevant aspects and corresponding recommendations are listed below.

Mission-specific solutions, algorithms, and requirements must still be defined separately for each mission. Although coordinated efforts between EOP and TEC are ongoing to develop generic processing components (e.g. BIBLOS, GRL, see [BIBLOS]), these elements lie outside the scope of this document.

4.2 Technical and Procurement aspects

4.2.1 Requirements template for E2ES Mission Performance chain

A generic User Requirements Document (equivalent to an ECSS Technical Specification) is available [GE2ERD]. It systematically covers all software components of the reference architecture shown in Fig. 1, as well as all software-system aspects—interfaces, processing performance, logging, dynamic behaviour, terminology, error handling, programming languages, installation, and more—and includes placeholders for mission-specific algorithms and ATBDs.

[GE2ERD] is designed for tailoring and should serve as the baseline for a standardised set of E2E Mission Performance toolchain requirements to be included in the ITT. It has already been applied in several Copernicus and Earth Explorer missions, simplifying preparation work and producing structured, higher-quality proposals that were easier to evaluate.

R-4.
Use a tailored copy of the Generic User Requirement [GE2ERD] when preparing the ITT.

4.2.2 Reference Architecture for Earth-Observation End-to-End simulators

A study (ARCHEO [RARCH]) was carried out to define reference architectures for End-to-End simulators, identifying commonalities across missions and, more broadly, across mission families grouped by sensing type (active/passive), frequency range, and similar criteria, while retaining the flexibility required for mission-specific needs. The generic User Requirements Document [GE2ERD] incorporates the ARCHEO reference architecture, which has served as the foundation for ongoing development of open-source software components for implementing E2E toolchains (BIBLOS [BIBLOS]).

R-5.
Use ARCHEO-E2E reference architecture and associated standardised terminology as basis for the architectural specification of any the End-to-End simulator [RARCH].

4.2.3 Interfaces

Use of bespoke interfaces between software modules of an E2ES toolchain demonstrated several disadvantages: it prevented direct integration into a common orchestration framework, increased development and testing time and cost, created vendor lock-in because components cannot be replaced without adaptation, and made comparison across similar missions difficult.

To address this, a simple, language-agnostic, generic, and lightweight file-based software interface—the "ESA generic E2E ICD" [GENICD]—has been defined. It provides configurability while supporting standard invocation, logging, and data definition. This approach also simplifies:

- re-use of an orchestration framework,
- re-use of subsystems and components from existing E2ES toolchains for future missions,
- combination of different missions within a single E2ES environment.

Using [GENICD] to specify module interfaces ensures out-of-the-box compatibility with the ESA openSF orchestration framework. Open-source, multi-language software libraries (OSFI, part of [OPENSF]) are freely available to implement it.

R-6.
Use the generic interface definition for the interfaces of SW module as specified in [GENICD].

4.2.4 E2ES Orchestration framework

To facilitate the use of standard/harmonised interfaces in the procurement of E2ES Mission simulation toolchain, to reduce the costs and development times avoiding re-implementation for every project, as well as to offer advanced automation and control/orchestration functionalities, ESA makes available the openSF [OPENSF] software framework to all internal and external projects.

openSF is an open cross-platform (Linux, OSX and Windows) orchestration framework available as binary and source code (under the ESA Community Permissive source licence) and there are many industrial players that can offer expertise in using it for development and integration. Since it implements the interface according to [GENICD] it allows the E2ES module developers maximal freedom while still ensuring drop-in integration process.

The openSF orchestration layer allows:

- scheduling and automating the orchestration of simulated scenarios for sensitivity studies with parametric iteration of user defined parameter (e.g. errors);
- modular execution of scenarios on subsystem and component level for independent performance analysis;
- automatic invocation of plotting tools for input/output comparison;
- full simulation and results traceability by archiving of results together with the configuration of each run, allowing to reproduce runs or support anomaly investigation;
- logging and filtering facilities.

R-7.
Require the use of the openSF [OPENSF] orchestration framework.

As openSF is maintained and developed under a long-term contract, it is continuously updated in response to feedback from performance engineers and module or processor developers. It is used across all Earth Explorer missions currently in Phase 0/A/B1 and is already employed in several Phase B2/C/D/E1 missions, including both Copernicus Sentinels and Earth Explorer missions.

It should be noted that the use of the openSF orchestration framework does not constrain development: if needed, any software module can still be executed and orchestrated directly from the command line (for example, via scripts).

4.2.5 File formats

The standardisation of header and semantically meaningful file naming is an important element of ensuring a sound and coherent file handling, data exchange and circulation within the GS and across missions but also in relation to the E2ES toolchain (e.g. for naming the Level 0, Level 1 or Auxiliary data files)

A generic tailorable and flexible standard for EO file format definition has been developed [EOFFST] and is used in all Explorer and Copernicus missions when ESA is fully responsible for the system. The EO File Format standard has been designed to allow user defined data format (e.g. NetCDF, XML, bespoke binary, JPG, TIFF, etc) but maintaining a standard XML header, filename structure and metadata set.

R-8.
Application of a tailored [EOFFST] is recommended for all missions where the Payload Data Ground Segment (called EOF-EOS) is designed to ESA specification. For those PDGSs where ESA is not the party providing the specification (e.g. Eumetsat), it is recommended that ESA seeks and confirms the existence of an equivalent underlying standardisation following the same principles covered by [EOFFST].

4.2.6 Earth Observation Mission software CFI

The coherency across all processors, systems, instruments, mission analysis, simulator of orbital, timing, pointing and attitude calculation is critical to ensure that comparison and interfacing can be performed with known errors, furthermore standardisation in relevant data formats (e.g. orbit and attitude files) allows seamless exchange and coupling of tools and SW systems and reproducible results.

While there are different set of libraries and algorithms available most of the calculation performed by EO systems are the same and therefore there is no reason to implement them multiple time with the associated higher cost and risks. The implementation of these algorithm is a long and specialised work, therefore ESA has implemented and made available to all users the EOCFI software libraries. These are a collection of multiplatform precompiled C, C++ and Java libraries for timing, coordinate conversions, orbit propagation, satellite pointing calculations, and target visibility calculations [EOCFI]. These libraries are already the de facto standard for nearly all ESA prototypes, operational processors, and orbit propagator tools, and recently EUMETSAT has begun to use the EO CFI for its own developments in Metop-SG.

R-9.
It is recommended to require using the Mission CFI [EOCFI] and corresponding formats [MSFFS] for all Simulation and Processors development.

The Mission CFI is subject of continuous maintenance and development, and it is continuously adapted in response to feedback from performance engineers and developers of processors and mission planning tools. Its application for in E2E simulation alleviates the burden of verification of the complex algorithms involved in mission and orbit analysis.

4.2.7 Calibration and characterisation algorithms

Calibration algorithms developed by the Instrument or Satellite Prime have, in several cases, proven to be inadequate and have shown limited commitment. Issues observed include lack of flexibility, insufficient understanding of the actual operational scenarios, and limited awareness of how the Ground Segment functions. These shortcomings have been particularly evident—though not exclusively—in the context of external and off-line calibration activities (e.g. transponder-based calibration or long term performance monitoring).

R-10.
Organise a formal review by independent experts of the calibration algorithms and characterisation methods developed by the mission prime contractor.

4.2.8 Scene generation

The functionality of the Scene Generation to be developed according to the reference architecture and that is used (see Fig 2) to verify the scientific requirement at MRD (Level 2) is focused on the simulation of geophysical realistic observable targets and scene. This is often not sufficient to provide inputs/TOA stimuli needed for the verification at Level 1 of the space segment requirements that might require high contrast, synthetic scene with non-geophysical values.

Furthermore, the procurement of Level 2 components (overall E2ES up to level 2) is assigned to (scientific) parties which are different from the one responsible of the development of the Instrument Simulator (IS) and of the Level 1 Processor Prototypes (L1PP) required to verify SRD (space segment).

It is desirable to explicitly define these two separate functions as well as to avoid contractual coupling where one party is dependent on the other to complete its task. (See approach defined in section 4.2.12)

R-11.

- *Define a single technical data interface (file format) used by all contracts (L1 and L2) between Scene Generation and Instrument simulator (or OSS) that can satisfy both needs of SRD and MRD verification.*
- *Procure as part of the Instrument Simulator a Simplified Scene Generator module compatible with these interfaces able to generate and inject the non-geophysical stimuli needed for SRD verification in the simulation chain independently from the full Scene Generator.*

4.2.9 Algorithms' maturity

Especially for innovative missions, where algorithms are less mature and do not have established heritage, it is important to ensure a review of algorithms at Level 1 and Level 2 compatible with the overall schedule. This was in past projects not always the case and discrepancies were found too late to be implemented.

R-12.

Organise a public peer review of the level 1 and level 2 ATBDs at a point in time (e.g. SAT CDR) that any feedback can still be implemented in the pre-launch delivery of the level 1 and level 2 prototype processors.

4.2.10 Terminology

Several EOP projects have begun their E2ES Mission Performance Chain development with inconsistent nomenclature across missions—for both E2ES components and data products. Identical terms have been used for different functionalities, different terms for the same functionality, and in some cases entirely new bespoke names have been introduced. Examples include OSS vs OPSI vs ISM vs FrontEnd, GPP vs L1PP/L2PP, GM vs GMd, RAW vs L0, and others. (see also glossary in AnnexA 7)

This inconsistency creates clear problems for software and documentation reuse, both within ESA and—more critically—for industry, which must rename elements in proposals, software, and documentation. The result is avoidable errors and misunderstandings.

R-13.

To avoid ambiguity, and facilitate the procurement process it is recommended to strictly adopt a single standard terminology in all E2ES relevant procurements. The terminology is described in Annex A and [GE2ERD] as well in [GRAWL0] for RAW and L0 data.

4.2.11 Flow of Calibration and Characterisation data to the ground segment

The data related to calibration and characterisation parameters coming from instrument design or measured on ground prior to launch are generally generated and provided by the Satellite/Instrument prime contractor. This data is used in the E2ES mission performance chain within the Simulator, the Level 1 Processor Prototype and eventually in the operational level 1 Processor.

It is generally the case that this data is provided in a way which is not adequate for direct use in a software system in term of format, parameter grouping, file-naming and for tracking temporal evolution and validity of the various parameters. Common case is that this data named CCDB or CKD (misleadingly since there is no Data Base involved) is provided as huge Excel spreadsheet or as collection of sheets or files, or as a directory structure populated with identically named files for the various ground characterisation campaign and in general just reflecting the working practices of that particular space segment industry team. Furthermore, some of the parameters are automatically updated while in orbit by the Level 1 Calibration algorithms while others are not, so there is also the need to cleanly update and evolve this set of data also to ensure separation between as-designed and as-built values.

The cases where the provided formats for the so called CCDB or CKD has been used "as-is" in the E2ES mission performance chains and Level 1 Processor resulted in technical orchestration and software difficulties, in an inefficient or limited technical solutions, in a not compliance of the provided files with the ground segment standards and in direct coupling of space segment activities not only from a content point of view but also at format level. It is strongly recommended to avoid using directly the physical files comprising the CCDB and CKD from space segment in any prototypes and operational processors.

This issue has been well addressed in some Project by decoupling the space and ground segment representation of these parameters and assigning to the Level 1 developer the responsibility to:

define the grouping, file naming, and reformatting of the relevant parameters from the CCDB or CKD data into ground segment AUX files according to GS format standards and keeping into account the Level 1 calibration and measurement processing and orchestration.
to (automatically) convert the format of the data received by the instrument/space segment industry (the CCDB/CKD) into these AUX files which are the its corresponding GS manifestation.

The guidelines have been collected in a dedicated Guideline TN [CCDBG].

R-14.
Follow [CCDBG] and decouple space and ground segment responsibility and tasks in particular by assigning to the Level 1 Prototype Processor developer the task to:

- *define content and format of the AUX files containing the instrument parameters based on the CCDB/CKD provided by space segment industry;*
- *ensure separation at semantic and at file level between values from design (as-designed or as-built) and in-orbit calibrated ones even when they relate to the same parameters;*
- *convert the CCDB/CKD into that AUX format to be used in the GS.*

4.2.12 Procurement organisation of L1 vs L2 E2E

The transition from Phase A/B1 to B2/C/D introduces challenges not only because the scope and detail of simulation and processing increase, but also because responsibility for Level-2 and Level-1 developments is separated across different contracts. This section outlines two organisational approaches designed to ensure seamless integration and coordination and to avoid the emergence of two incompatible software products—the L2 E2ES and the L1 E2E—which would require additional integration effort and risk late discovery of inconsistencies between Level-1 and space-segment activities and Level-2 scientific retrieval.

The proposals below also take into account the relative schedules of the various developments.

Note: The scenario in which the full Level-1 E2ES is procured directly within the Satellite Prime contract is not discussed here (as it is not recommended). If required for programmatic reasons, it can replace step (b) below.

4.2.12.1 Integrated Approach

This is the recommended approach and should be selected in Phase B2/C/D as the first option.

a) **Level 2 E2ES:** Using the heritage of the E2ES developed in Phase A/B1, set up a contract directly from ESA with a Consortium including a SW engineering house and Scientific entities to maintain and extend the (scientific) SGM and L2PP (as evolution of the L2 Retrieval Module developed in Phase A/B1).

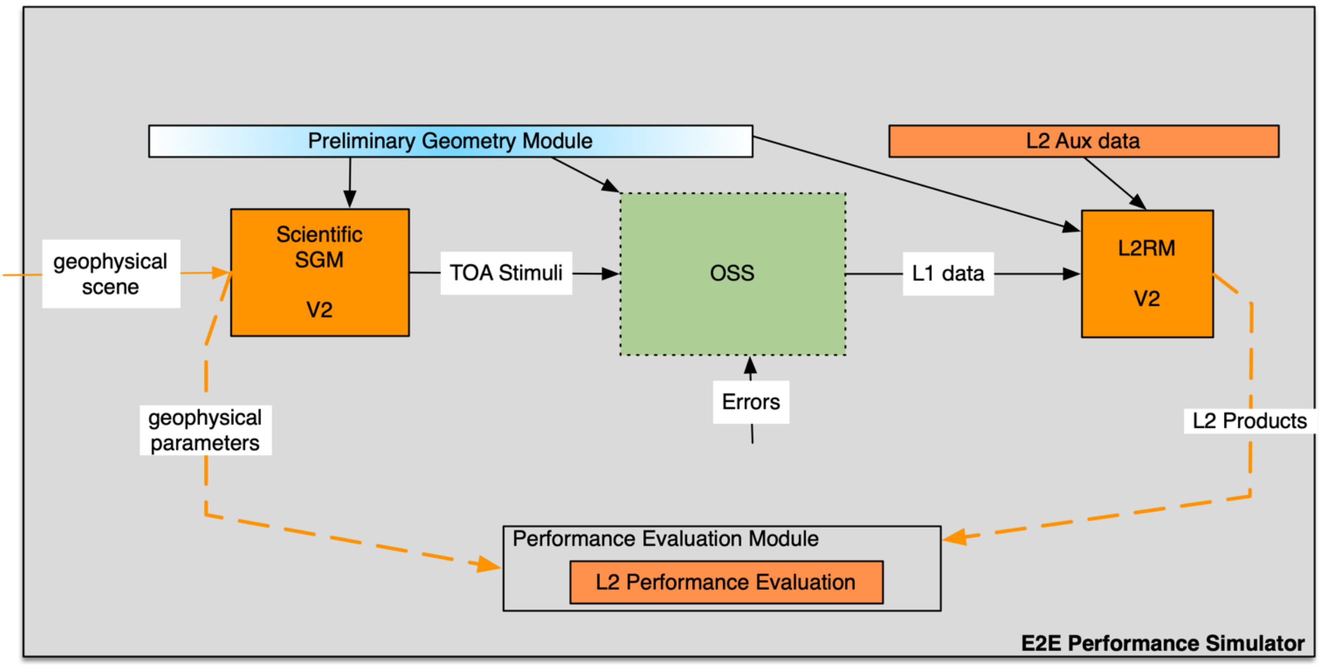

Figure 3 Evolution of L2 E2ES from Phase A/B1

b) **Level 1 E2ES:** Set up, directly from ESA, a contract with a Consortium led by a SW house with experience in E2ES, with ESA reference architecture and tools and with dedicated support WP from Instrument/satellite Prime as subcontractors.

The output of this development is an E2ES limited at L1 (ISM+L1PP) and including also a Simplified Scene Generator (to remove dependencies from (a)) built on the same infrastructure/environment as the L2 E2ES above. The provision/support for algorithms, data and of independent validation is performed by the Instrument/Sat Prime within this contract.

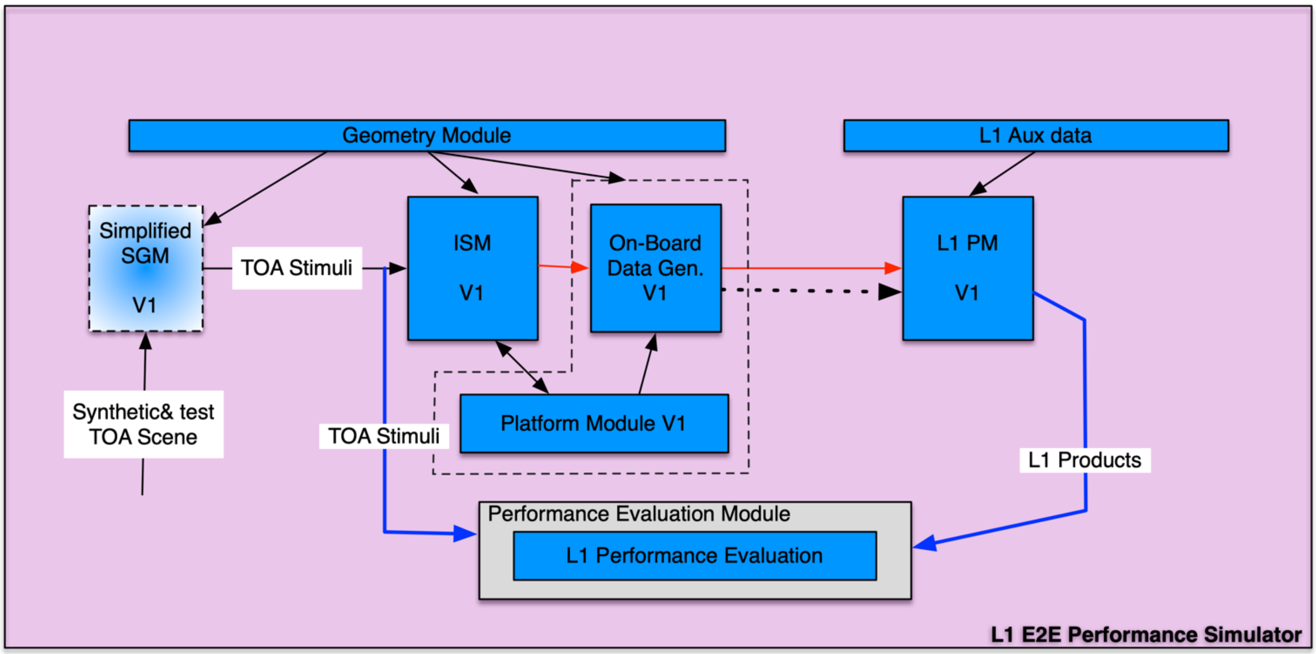

Figure 4 L1 E2ES SW Product

c) **Merge** the two developments to achieve a single E2ES Mission Performance Chain by defining a task as part of "a)", or separately with a dedicated SW house, to integrate the L1 E2ES developed in b) with the upper level L2 E2ES developed in a) and make the resulting available to both L1 and L2 teams to continue their work. The task for integration will include:
- verification and adaptation of I/F as needed;
- substitution of OSS in the L2 E2ES with the ISM/OBDGM/L1PP developed in L1 E2E;
- integration of the L1 PAM function with the L2 PAM function;
- substitution of GM developed in the L2 E2ES with the one developed in L1 E2ES and integration of any missing function (e.g. specific to SGM);
- Harmonisation of any common AUX files required by both L1 PM and L2 PM.

d) Afterwards **maintain** a single unified E2ES integrating new modules coming from L1 or L2 as they are unit tested and qualified by the responsible team.

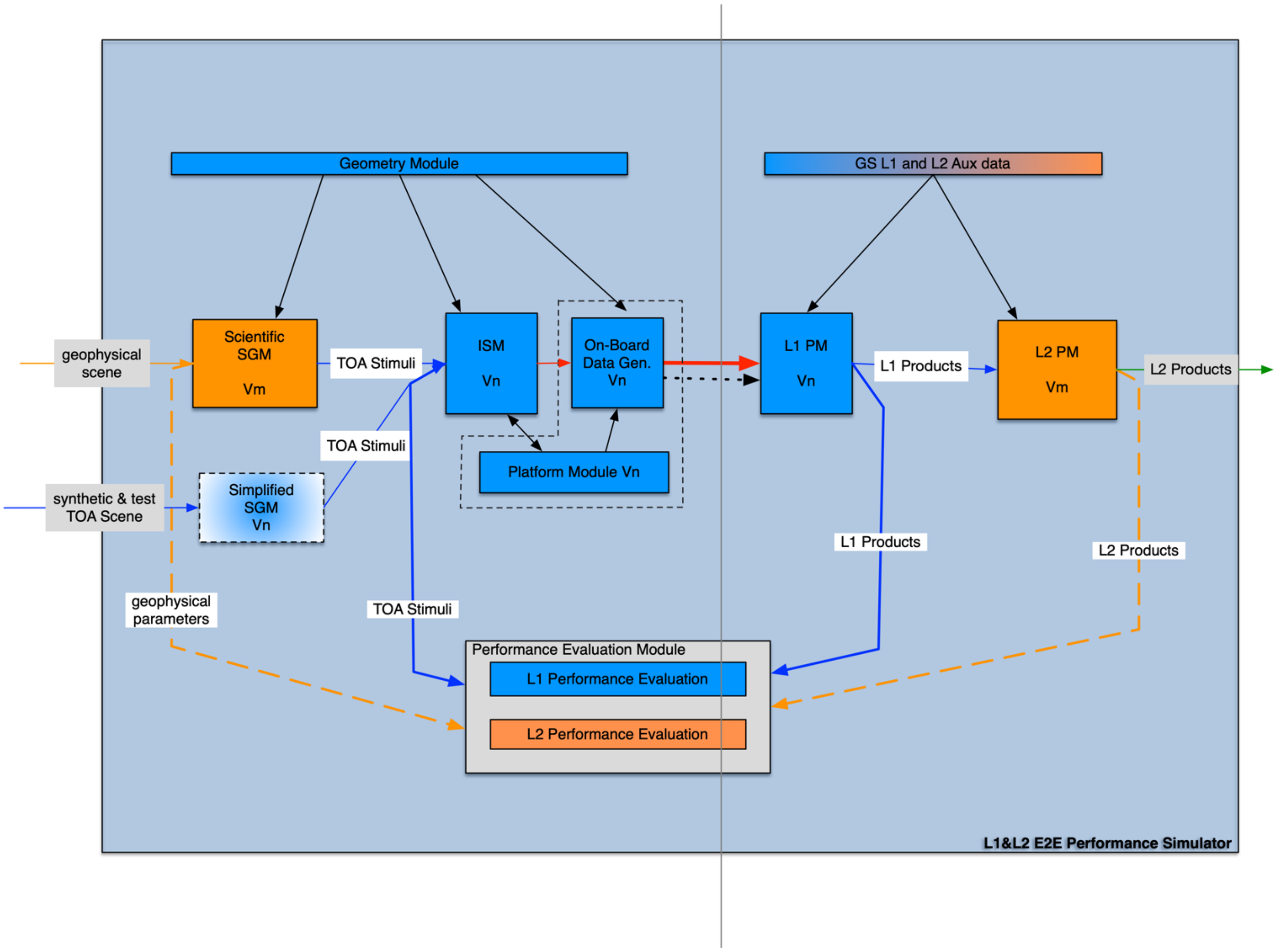

Figure 4 Integrated L1 and L2 E2ES (Blue L1, Orange Science/L2)

Once the Integrated E2ES be maintained in a shared repository; the key advantage is that it allows ESA, L1 and L2 teams:

- to access/use at <u>any moment</u> in time a functioning E2ES chain that is best representative of the behaviour of the system and on which to base the respective L1 or L2 work;
- to directly inject evolutions and corrections and validate them with no additional separate steps;
- to independently generate any needed scenario and test data.

4.2.12.2 Parallel Approach

This approach (illustrated in Figure 5, Parallel L1 and L2 development approach) retains two separate contracts, software products, and development streams for the L1 E2ES and L2 E2ES. Coordination is performed offline, not through a unified L1–L2 E2ES, but by ad-hoc data exchange between the teams (e.g. providing L1 products or other test datasets from L1 to L2) and by comparing and evaluating L1 and L2 results independently.

The advantage is that the two developments require little or no coordination. Each can proceed on different software and data baselines, with different requirements, operating systems, logging conventions, and so on.

The disadvantages are significant. Only limited, manually exchanged L1 datasets can be used for L2 testing; full end-to-end verification and validation is not available out-of-the-box; performance assessment becomes labour-intensive; and any change (e.g. interface updates) introduces delay, dependency, and additional cost because fixes can only be implemented and validated by the other team.

This approach is feasible but <u>not recommended</u>.

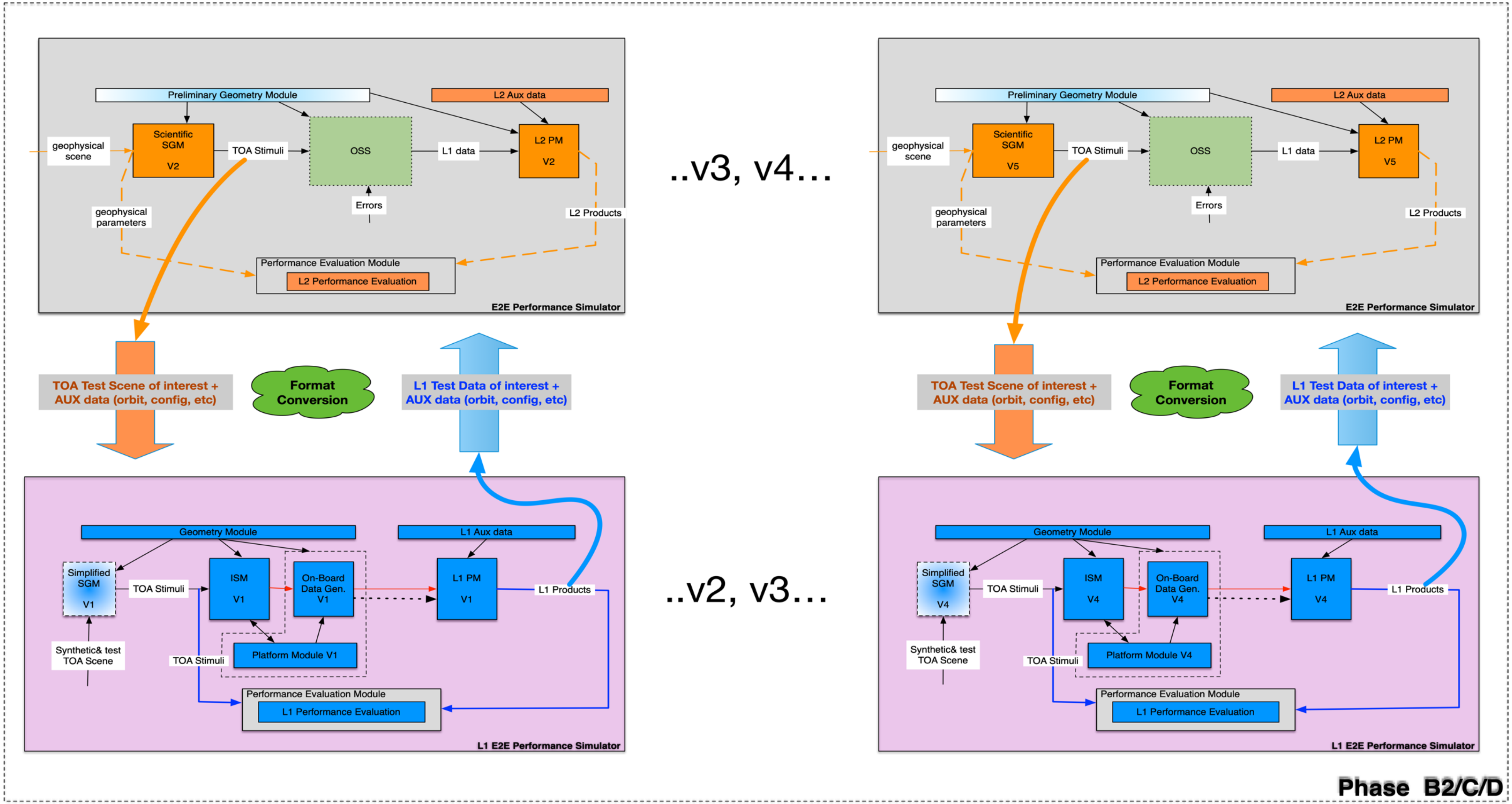

Figure 5 Parallel L1 and L2 development approach

4.2.13 Deliverable Items and Services

Different stakeholders depend on the timely availability of the E2E mission-performance toolchain, its components, and the data it generates. Their schedules and priorities differ: what is critical for space-segment designers may be less critical for ground-segment integrators, mission-performance engineers, or scientists. It is therefore essential that all elements developed as part of the E2ES toolchain be fully deliverable to ESA and to any external party without restrictions.

There have been cases where this was not ensured because the DISL (the contractually binding document) was not consistent with the task definitions (e.g. those in [GE2ERD]). As a result, key E2ES elements were developed but not contractually deliverable. This problem typically arises when [GE2ERD] is not followed and the E2ES is procured through the satellite prime.

R-15.
Strictly use a standard formulation for the E2ES deliverables in the Deliverable Items and Services List (DISL) for the procurement at the start of Phase B(2). The proposed formulation is described in Annex B.

4.2.14 E2ES Deliveries and Development Process

The initial definition of E2ES software deliveries and delivery dates is inherently speculative, and experience shows that these deliveries often need to be revised or delayed due to evolving inputs and requirements from multiple actors (Space Segment, scientists, software developers, the operational Ground Segment, etc.). This has negative impacts on functionality, cost, and schedule. A mechanism is therefore needed from the outset that allows deliveries to be provided flexibly and incrementally in response to external constraints.

In addition, while the exact delivery schedule and number of drops are mission-dependent, the users of the E2ES have different timelines and priorities. It is therefore beneficial to interleave formal deliveries with periodic intermediate ones, avoiding a model in which the first delivery is tied for example to the Satellite/Instrument Critical Design Review. Instead, intermediate deliveries should be planned well in advance so that the Space Segment, Ground Segment, and Level-2/Science teams can all receive timely inputs without dependencies or interlocks.

Finally, it is essential that deliveries remain available to all parties even when some inputs are missing or delayed.

It is suggested this is achieved in 2 ways:

1) **Technical:** Request that all the SW and document development is maintained in a common repository (e.g. Git) integrated with a complete CI process (e.g. Jenkins, Bamboo, …) that ensures automatic one-click check-out, build, testing and packaging and deliver in a target electronic out-tray together will all the relevant test data, configuration and documentation. In this way the latest E2ES Product is buildable and deliverable instantly at any time with little or no effort. This technique has been implemented in deliveries of a complex multi platforms

100K plus line of code for a SW product in EOP-PE and reduced industry's effort for build, delivery and test from 2+ weeks to 1 day.

2) **Planning:** Shift from defining deliveries by fixed, pre-specified content to establishing a timeline of regularly spaced "agile" deliverables (typically every 6–9 months), each of which consolidates all inputs agreed and available at that point in time.

The starting point is to associate some E2ES deliverable dates with Space Segment milestones and others with Ground Segment or Science milestones (see Figure 6). If a specific input is not available at a given time—for example, an updated CCDB or a revised product format—it will simply be deferred to the next scheduled delivery, while all other components developed and tested up to that point will still be released. Additional informal deliveries can be introduced at any time with minimal effort when needed, for example to address critical software bugs.

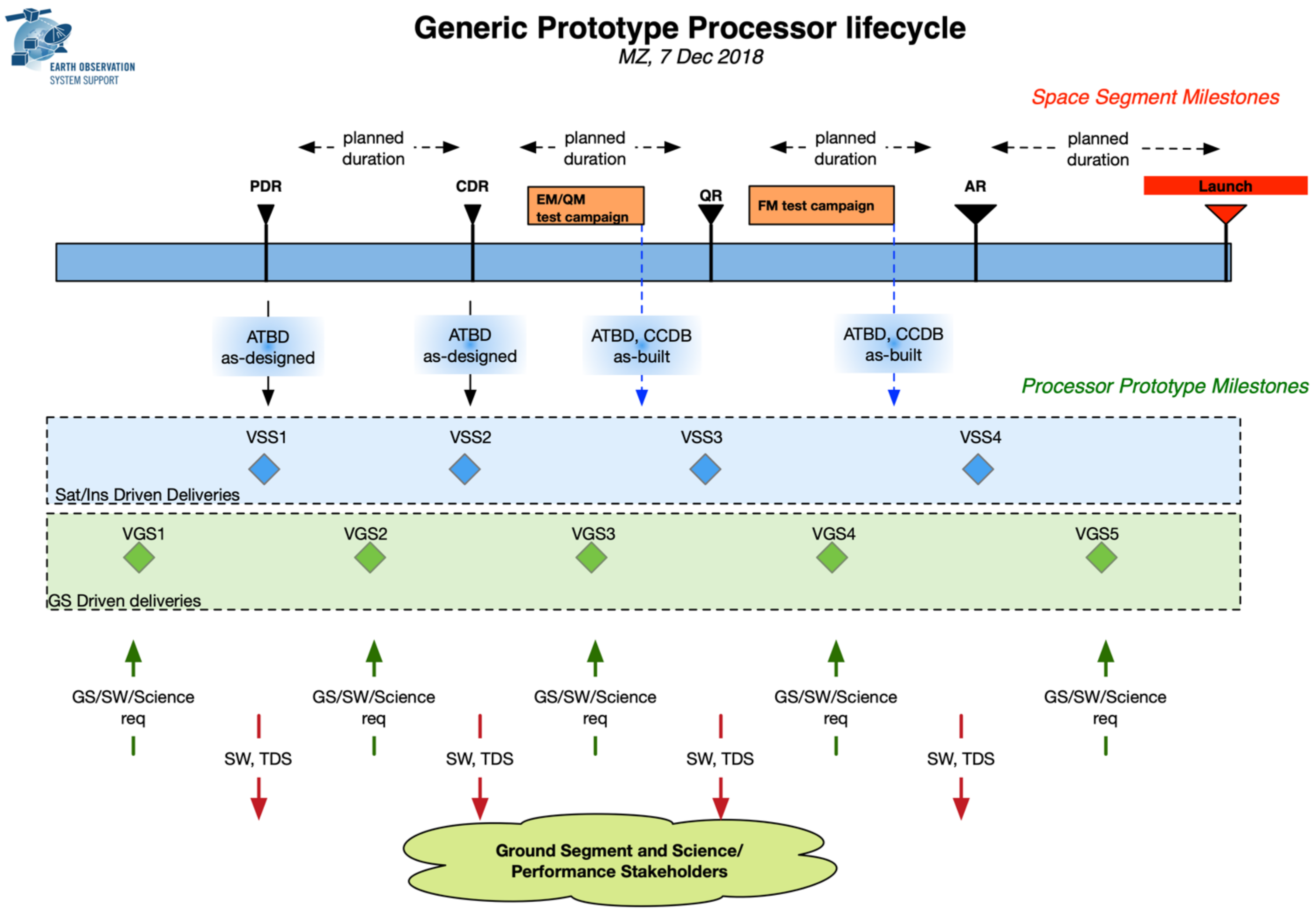

Figure 6 Processor deliveries' schema

4.2.15 SW environment preventive maintenance

In past EOP projects it has been observed that development has been performed starting from a specific frozen baseline of the operating system and COTS definition as available at the time of the ITT, with no plan nor provision for evolution/maintenance of the environment and no upgrade of the OS, of the COTS and of other libraries and framework (e.g. openSF).

In previous years E2ES developments, this has resulted in major obsolescence, delays, functional limitation as the platform, COTS or library initially used was deprecated, found buggy or not maintained any more.

R-16.
It is recommended to include in the initial SoW a preventive maintenance mandatory task that, once a year, covers the execution, upon request from the Agency, of update of the OS, COTS, libraries/Frameworks.

4.2.16 Software Licensing

Standard ESA contractual clauses for software are tailored to on-board flight-system development, where protecting industrial IPR is a primary concern. Historically, clauses have restricted distribution to the Project, to the Programme (e.g. Copernicus or Meteosat), or to specific parties.

This approach is unsuitable for the E2ES mission-performance simulator toolchain, which must be distributed widely to a variety of internal stakeholders and benefits—especially for processing components—from contributions and troubleshooting by the scientific and engineering community. A modern open-development model directs budget toward added-value functionality and refinement rather than repeated re-implementation of proprietary code.

Making the software freely available enables internal and external entities to exchange, modify, and compare algorithms and data across missions. It also provides a starting point for new missions of similar type, enabling reuse, reducing cost, and lowering risk as experience and solutions transfer from one EO project to the next.
BIBLOS libraries [BIBLOS] are also being developed as open source and will be released by ESA. However, this approach needs to be extended to all new developments. Several upcoming Earth Explorer and some Copernicus missions already include modified clauses supporting this model. Under this approach, IPR (or BIPR) may remain with the developer, provided that a permissive binary-and-source licence is granted for broad redistribution (for any Agency purpose, including beyond the specific programme) and sublicensing.

To safeguard Member State interests while allowing broad distribution, ESA has created a permissive licence in an open-source spirit but limited to Member States: the ESA Community Licence Permissive [ECL].

R-17.
Require that both source code and binary are delivered to ESA according to the standard ESA licence ECL Permissive with full right to sublicense for any ESA activity and add to the contract a tailored clause 42 to this effect as per text here below.

CLAUSE 42: SOFTWARE

Sub-Clause 42.4:
The Contractor shall deliver directly to the Agency the software, developed under this Contract, also in source code form and distribute it under an ESA Software Community Licence – Permissive – V2.4, attached hereto as Appendix #.

4.2.17 Mission Level Verification Approach and tools

ESA missions involve several levels of verification and validation, including:

a) verification of Level-1 requirements within the Space Segment procurement, initially to support design and later to confirm compliance with the Space System SRD;
b) verification and validation of Level-1 end-user requirements using simulations and real data, based on operationally produced products (E2E);
c) verification and validation of Level-2 requirements using Level-1 performance budgets;
d) verification and validation of Level-2 end-user requirements using simulations and real data, based on operationally produced Level-1 and Level-2 products (E2E).

While points b), c), and d) make direct use of the E2ES (at L1 or L2) as defined in this document, the approach for verifying point a) — compliance with the Space Segment SRD — has been less consistent. This is partly because the requirements to be verified differ in nature: some concern internal instrument quantities, while others correspond to parameters that appear in operational end-user data products.

For example, certain low-level instrument requirements may rely on dedicated tools (e.g. optical parameters computed with Zemax) and bespoke measurement campaigns or processing steps that are not part of the nominal L1 processing and are not represented in the L1 products. Other requirements, however, map directly to parameters present in the operational L1 products.

The difficulty arises when verification of Space Segment requirements relies on processed—not directly measured—quantities. In such cases, the numerical processing becomes part of the system under verification. To ensure correctness, several missions mandated the Space Segment Prime to verify the SRD requirements using the L1 Processor Prototype (L1PP). Since this would not be accepted unless the L1PP were placed under the same Space Segment contract, it led to procuring the L1PP through the Space Segment Prime. This, in turn, created the broader issues described elsewhere in this document, because the L1PP (and the E2ES) became primarily instruments for verifying SRD requirements rather than for performing end-to-end performance assessment.

A suitable approach that preserves the involvement of the Space Segment prime while allowing the E2ES to be procured independently—as recommended in this white paper in section 4.3—is to keep the Space Segment prime engaged at the beginning and at the end of the performance-assurance lifecycle, without requiring it to use tools developed under an external contract.

This can be achieved by agreeing on an SRD-requirement verification approach making use of different verification levels (internal vs external):

1. **Internal requirements** (within Space Segment contract) : verified by the Space Segment prime using ad-hoc tools, COTS, or analysis methods approved by ESA.
2. **External requirements** (within Space Segment contract) : verified using breadboards or methods developed or selected by the Space Segment prime (and approved by ESA). These breadboards are not subject to the software-engineering, architectural, delivery, or interface constraints imposed on the L1PP.
3. **External requirements** (outside the Space Segment contract): verification task using the E2ES, including the L1PP.

Under this scheme, the Space Segment prime retains responsibility at the start of the performance chain by defining and delivering the ISM and L1PP algorithms (Technical Specifications / ATBDs), the data definitions (IODS), and the test data. At the end of the chain, in addition to SRD requirement verification, it performs a full verification of the external performance requirements (independent from the one performed within the Space Segment contract) using the implemented L1PP.

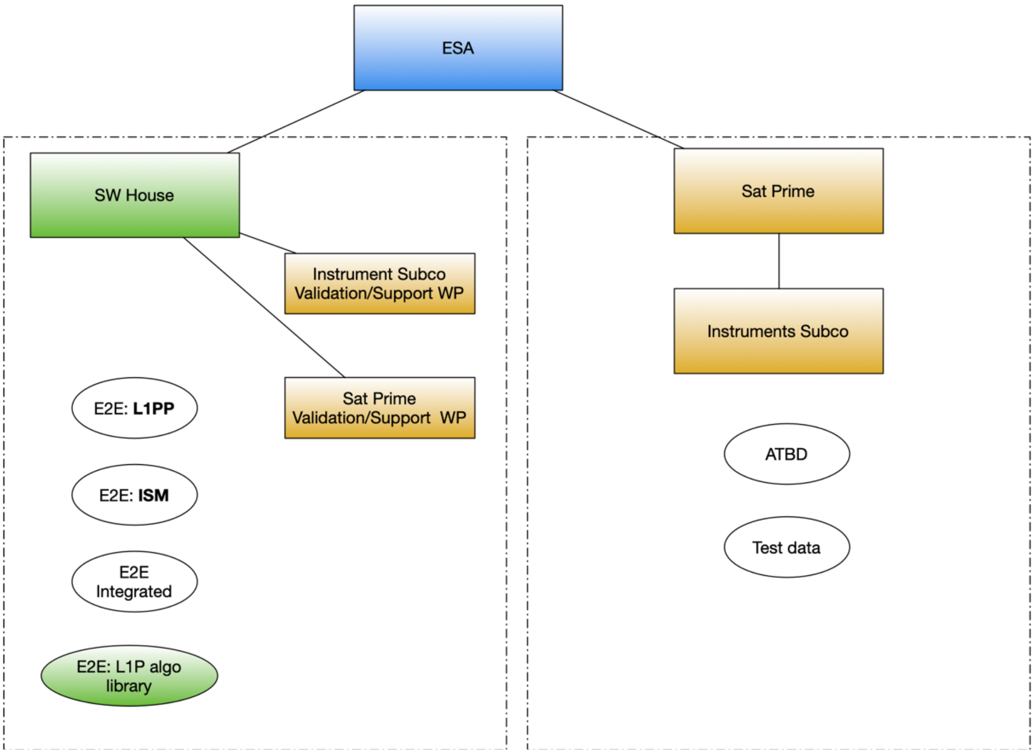

Figure 7 Satellite Prime Involvement

R-18.
If the E2ES procurement is implemented separately from the Space Segment contract, it is recommended to include in that dedicated contract a mandatory task, performed by the Space Segment industry, to independently verify the agreed external SRD requirements using the E2ES and the L1PP.

4.3 Organisation

In addition to the contractual, procurement and technical aspects described in sections above there are also lesson learned regarding the organisation between ESA, L2/Science, Industry and Operational actors.

A commonly used allocation of responsibilities and products is shown in Figure 8 below.

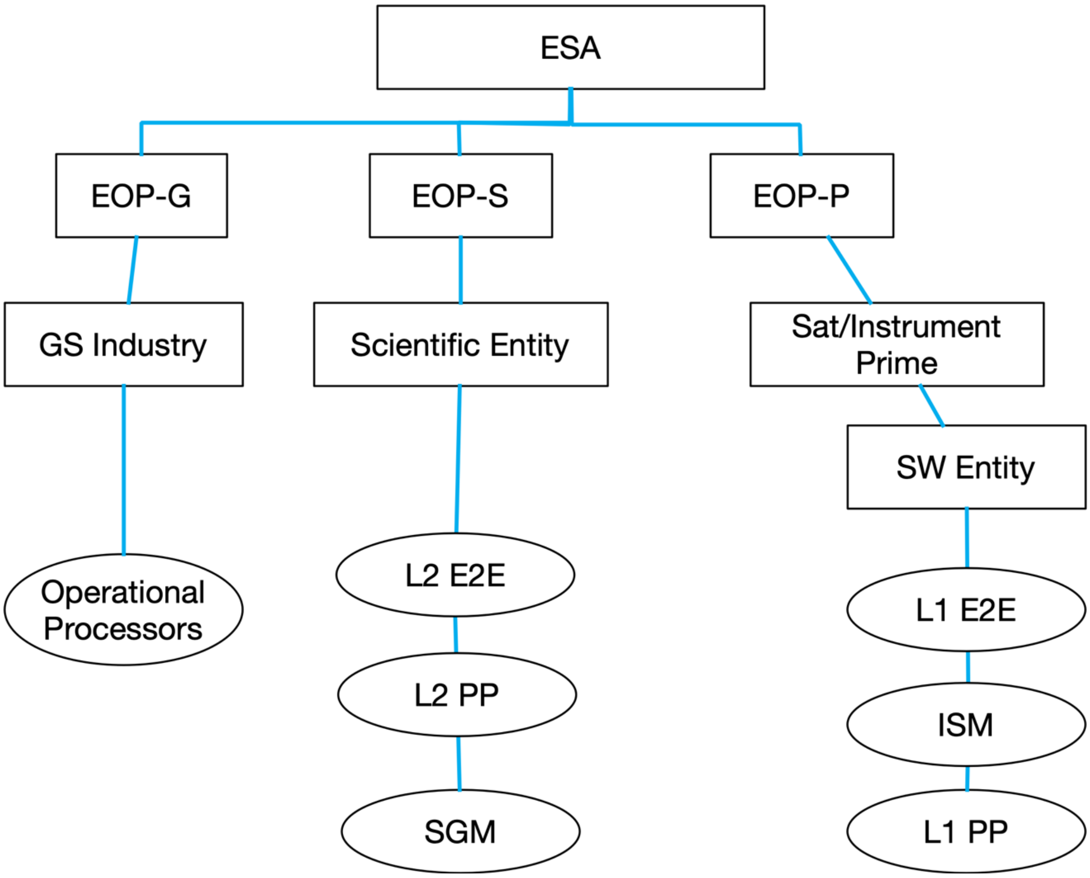

Figure 8 traditional organisation

Lessons learned from current EOP projects show that:

1. Embedding E2ES procurement within the main Space Segment contract consistently leads to problems. Coordination with other actors—including ESA, the Ground Segment, and the Science Consortium—is limited or impossible; priorities are driven largely by Space Segment needs; flexibility is minimal; and the required system-level ownership and E2ES expertise are weak or absent. The result is delays, additional coordination and re-implementation costs, and technical solutions that are only marginally compliant.

2. The development and quality of Level-2 deliverables are often inadequate and late. In most cases, software-engineering practices and requirements are weak or absent, and coordination with Level-1 is minimal. This repeatedly leads to late discovery of missing software elements,

inconsistent products, and algorithmic incoherence between L1 and L2 in the E2E chain, as well as issues within the operational Ground Segment (e.g. in the L2PP).

A revised organisation ensuring an effective coordination with L2/Science and avoiding these pitfalls is proposed and being used in new projects (e.g. Harmony and Forum).

R-19.
Procure the E2ES under a direct ESA contract with a competent SW house including dedicated work packages for support and independent validation using the E2ES by the Instrument/Sat prime as per Figure 9.

R-20.
Ensure that, starting from Phase B2/C/D, the L2 scientific elements are developed with support from a competent SW entity and with SW and IF requirements coherent with the L1 developments including provision of early integration tasks between L1 and L2 E2E. Figure 9.

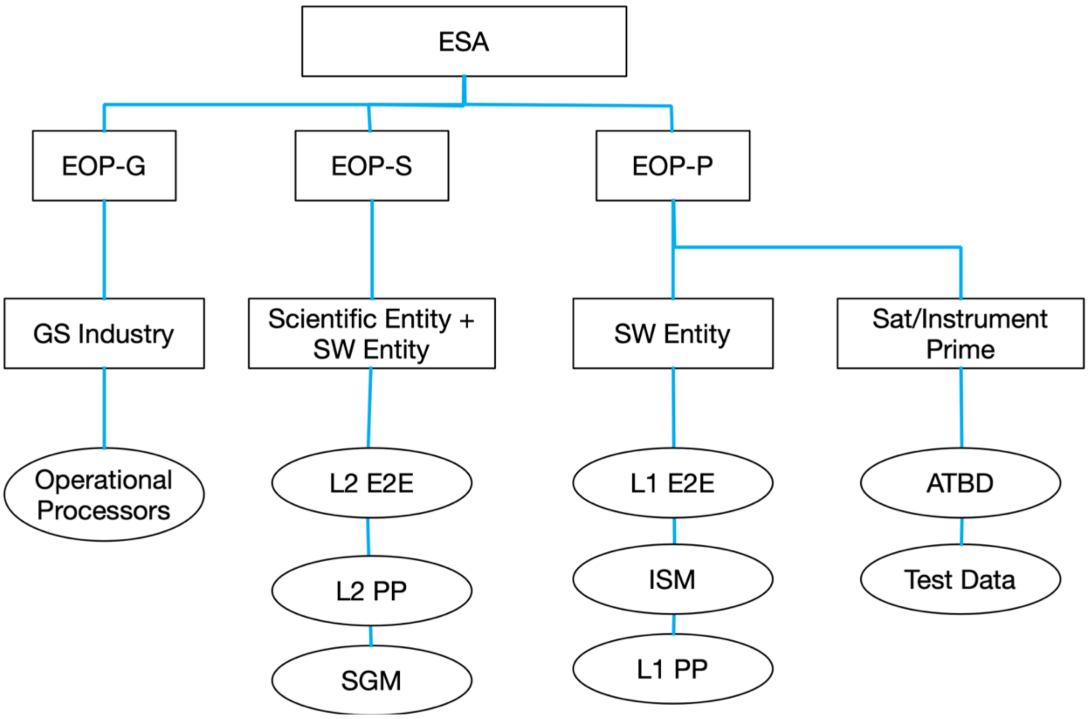

Figure 9 recommended E2E organisation

4.4 2.2 E2ES components re-use decisions

As described in section 3.1 the E2ES are developed and evolve in various phases. To reduce costs, development time and risks the re-use of existing standard tools, software module algorithm should be considered at every stage and especially at the beginning (start of Phase 0/A) and at the start of Phase B2/C/D when the OSS is substituted by a detailed modelling and the Level-1 Processor Prototype defined. A number of activities at ESA (EOP and TEC) have defined and produced re-usable software components and libraries, e.g. BIBLOS [BIBLOS], EOCFI [EOCFI], DFDL4S [DFDL4S], openSF [OPENSF], etc.

Information about the availability of existing components, libraries and framework and the suitability for re-use is not generally known to the EOP-P Project teams therefore this should be systematically addressed especially in Phase B2. Such evaluation should consider:

- Reuse of models and algorithms from Phase 0 to Phase E2.
- Reuse of models and algorithms within ESA projects.
- Need for automated operation.
- Need for automated input/output comparison, i.e., scenarios and expected L1/L2 output.
- Provision of in-house maintenance and support.

R-21.
Evaluate as part of the procurement decisions upon entering Phase B2 and the ITT preparation, the suitability of existing elements for re-use.

4.4.1 Component re-use lifecycle

In addition to re-use of SW libraries and frameworks there is also re-use of SW components across project phases.

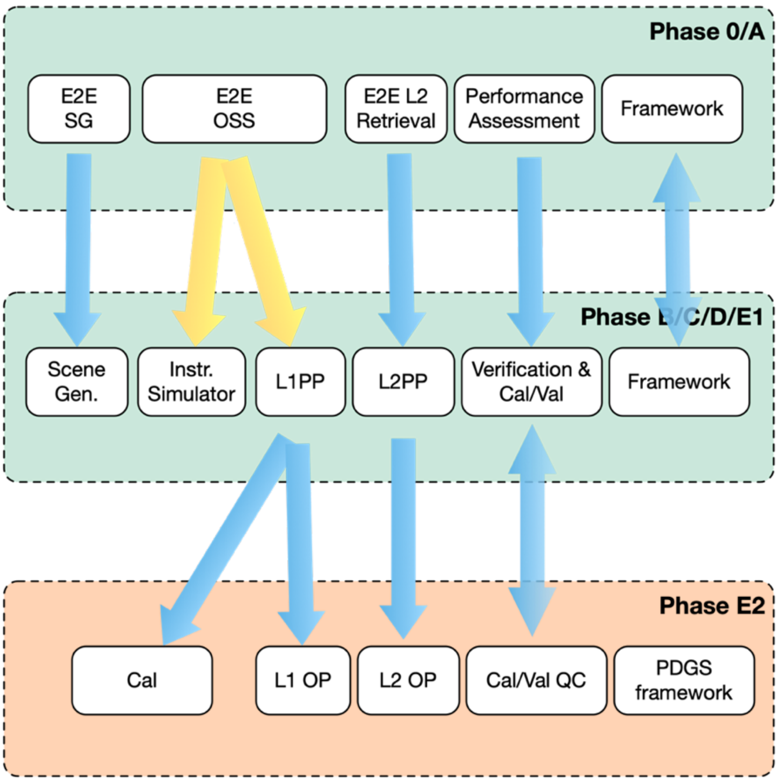

Fig. 3 Component re-use and relationship lifecycle

The models that are expected to cover most needs are:

1) Transition from Phase 0/A/B1 to B2/C/D/E1:

a. No re-use.
b. Re-use of the framework.
c. Re-use of the framework and the entire E2ES (as input at the beginning of the phase B2/C/D/E1 development to be further evolved).

2) Transition from Phase B2/C/D/E1 to Phase E2 (limited to prototype to operational developments):

a. Independent coding of processors and tools (based on common algorithms and test data).
b. Re-use of L1 prototype SW (or underlying libraries) for the operational processor and calibration.
c. Re-use of L1 and L2 prototype SW for the operational processors and calibration.

The decision on re-use of elements developed in phase B2/C/D/E1 in phase E2 can be taken independently of the choice of high-level development model (see chapter 5). Ongoing developments in standardisation of modules for such tools may change this in the future. The re-use of elements of the generic E2ES framework into the PDGS is generally considered not necessary/applicable.

It is important to note that the decision on the development model for the operational processors affects the deliverables for the prototype processors and hence the procurement of the E2ES for the Phase B2/C/D/E1.

5 Development of the Data Processors

5.1 Development models for Prototype and Operational Processors

In the procurement of the operational processors (OP) one of the following three approaches is generally used:

OPTION 1: "Cascade": PP (prototype) separated from OP. OP implemented from scratch the Processing Baseline defined by the L1PP, which is the combination of the Detailed Processing Model (DPM), the Input-Output Data Specification (IODS), and the Test Data Set (TDS) (corresponding to option 2a in 4.4.1 above).

OPTION 2: "Evolution": PP that itself becomes the OP (corresponding to options 2b and 2c in 4.4.1 above).

OPTION 3: "Hybrid": PP to OP via the Processing Baseline and in addition the exchange libraries of algorithms from the PP.

5.1.1 Discussion

The choice between development models should ideally be driven by the level of consolidation of the algorithms in the domain that the instrument/mission addresses, constraints related to the selected industrial consortium, internal or external split of responsibility (EOP-P, EOP-G, Eumetsat, etc) and costs.

5.1.1.1 Evolution vs Cascade trade-off

In general Data products, algorithms and payloads that have a high degree of innovation are expected to require frequent algorithm updates before and after launch and hence the "**Evolution**" model is considered more suitable. The advantages are:

- Lower cost as it avoids two separate developments and the long and expensive update cycle from Prototype to Operational repeated at every change.
- Less management required of the interface between PP and OP teams.
- Transfer of data/algo/etc from PP, E2E simulator/AIV to Operational and vice versa is direct.

The disadvantages of the "**Evolution"** model wrt "**Cascade**"are:

- it requires an industrial contractor of high SW expertise in addition to algorithmic one to produce an operationally ready and performance *drop-in* Operational Processor;
- it foregoes the "independent" implementation and cross checking normally performed in the "cascade" model (although this can be performed in other ways).
- makes the Operational processor implementation dependent on software design and choices made during the development of the Prototype processor.

Provided that the Prototype Processor includes adequate requirements to support later evolution into the Operational Processor (e.g. code quality, scalability, performance, interfaces), the **Evolution** model is the most efficient and should be adopted as the default approach unless specific reasons justify an alternative.
One example is when the Operational processors requires execution within a specialised software infrastructure (e.g. message-based interfaces, direct linking of processor code into infrastructure components) or specialised hardware environments (e.g. GPU clusters). In these cases the **Cascade** model (or the Hybrid see below) is a more suitable choice because the operational implementation is expected to be fundamentally different at software level, interfaces and executing environment from the one required by the Prototype.

Finally, the selection between the "**Cascade**" and "**Evolution**" models is also driven by programmatic considerations—for example, when responsibility for the operational Ground Segment lies outside ESA (such as with EUMETSAT)—regardless of the maturity of the algorithms in the mission's domain. An assessment must therefore consider not only cost aspects but also the technical risks associated with two developments (Cascade) versus one (Evolution), as well as the risks related to integrating an Operational processor CFI into an externally provided infrastructure, whether standard or non-standard. For innovative missions where the operational Ground Segment is provided by an entity other than ESA and the "**Cascade**" model is selected, it is essential that ESA secures, from the outset, agreements with that party and with industry to enable frequent updates of the processing baseline.

5.1.1.2 The Hybrid model

The "**Hybrid**" model is an intermediate approach that can also be used to reduce implementation time and maintaining algorithm alignment when Prototype algorithm API/libraries are re-used in the Operational processor. An example of this approach is shown below (Figure 10, Figure 11)

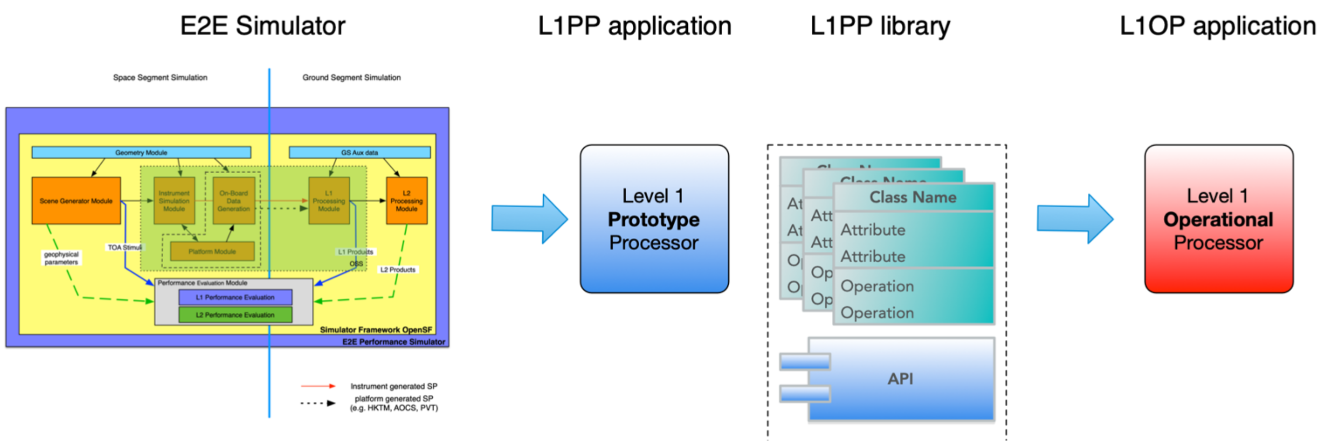

Figure 10 Hybrid model with re-use

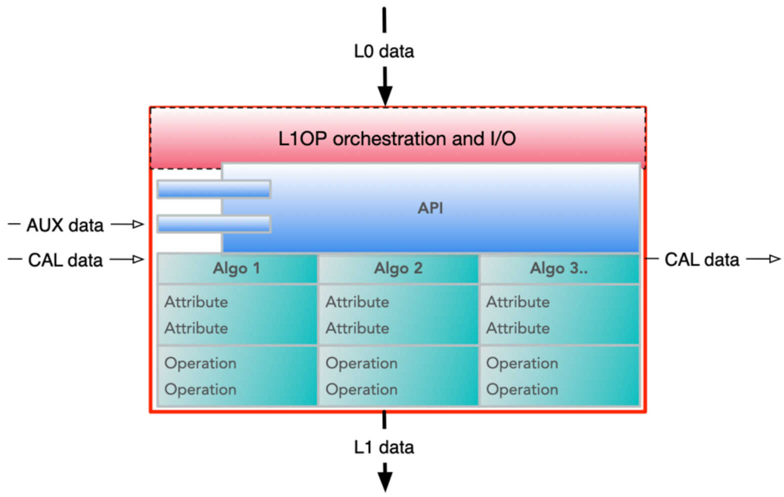

Figure 11 L1 Operational processor with API re-use

5.1.1.3 Experience from past

Finally, the diversity of organisational approaches in the past has not only been driven by technical or scientific considerations but also by the need to compensate mismatches between project needs and contractor experience. A further source of diversity is the lack of best practices resulting in similar problems being addressed different methodologies and with different success.

Operationally robust E2ES systems that can evolve throughout the entire project lifecycle—rather than being one-off developments for mission selection—were consistently recognised as key contributors to mission success.

Our inventory has shown that unplanned changes between approaches have occurred for several missions during the phase B2CDE1 but exclusively in the direction away from independent prototype and operational processor coding ("**Cascade**") towards utilisation of the prototype code in the operational processor ("**Evolution**" or "**Hybrid**"). Such transitions have occurred due to:

- Development problems in the operational processors.
- Processor algorithm changes could not be incorporated fast enough into Operational Processor, or it was too expensive to do so.
- Development time constraints for full-mission reprocessing with improved algorithms.
- Identical winning bidders for both prototype and operational processor development contracts.

Since none of these transitions were intentional, and most came with extra cost and delay (first two bullets) or a loss of independent verification (last bullet), we recommend as anticipated above to proceed as follows: perform a trade-off before phase B2 to select the optimum development model on a case-by-case basis, where the following criteria are considered:

- Level of consolidation of algorithms in the domain of application.
- Complexity of algorithms.
- Level of difficulty to meet processing time constraints for Near Real Time processing (where applicable).
- Overall schedule and Time available for development.

- The presence of algorithm expertise monopolies in the domain of application.
- Cost/manpower constraints.
- Industrial setup including the E2ES.

R-22.
Perform an explicit trade-off of the model to be used for development and procurement of the processors.

For the E2ES procurement—which includes the L1 Processing Module—it is essential to ensure that the delivery from the Space Segment prime (if that approach is chosen) is adequate not only for meeting its space-segment performance-verification obligations toward ESA, but also for serving as a reference or basis for the Operational Processor (in Cascade or Hybrid development models) or, in the Evolution model, for serving directly as the Operational Processor itself. To ensure that the necessary requirements are properly captured, the use of [GE2ERD], suitably tailored, is recommended.

As noted in Section 3.1 of [GE2ERD], the dual purpose of the space-segment E2ES delivery—supporting both space-segment verification and ground-segment reference—has in multiple cases produced negative outcomes, because the prime contractor is a stakeholder for payload-performance verification but not for ground-segment needs. Although not yet common practice, the SMOS example shows that parallel developments—one under responsibility of the Space Segment prime and one under ESA contract to expert teams—with appropriate exchange of characterisation information, avoid these issues (see Recommendation R-18) and it is being used in newest EE (partially Forum and Harmony)

5.2 Cost estimates for Procurement of the E2ES

The costliest modules in the E2ES are the Level-1 and Level-2 data processors. For passive UV/VIS/NIR/TIR sensors, the Level-2 processor typically costs significantly more than the Level-1 processor. For active optical sensors the cost ratio is more balanced, and for active microwave missions it is often inverted.

The main cost drivers are:

- Number of products, number of instrument calibration modes, number of measurement modes.
- Processing level (1, 2, synergetic).
- Application (prototype, operational, reference* or not).
- Technical/Geophysical Domain (e.g. altimetry, atmospheric sensing).
- Heritage.
- (Lack of) Competition, Intellectual Property Rights.
- Hardware included or not.
- Other activities (Commissioning support, Cross-verification support).

*= "reference" indicates that the processor is to be accompanied by a very detailed specification allowing independent coding of an operational processor with the same output quality. This corresponds to the "**Cascade**" implementation model in section 3.1.

Typical ballpark figures at time of initial writing of this document (2020) are as follows:

- ISM and all modules except L1PP, L2PP: 300k to 800k.
- L1PP 300k to 1.5M per instrument.
- L2PP 300k to 4.5M per instrument.

A cost modelling spreadsheet for these elements exists with ESTEC Cost division (TEC-SYC) although it has not been recently updated.

6 Conclusion

Concepts related to E2ES have been introduced and discussed. Significant experience in harmonisation and common practices has been gained in Phases 0/A/B1 and applied to several missions in Phases B2/C/D/E1. This document outlines the initiatives for further harmonisation and identifies lessons learned from E2ES and processor-development activities. Building on these lessons, additional best practices are proposed, together with supporting templates and tools, with the aim of fostering a bottom-up adoption. Continued feedback is encouraged to refine these resources and support broader consensus on these recommendations as more missions are addressed.

7 Annex A: Terminology

ESA develops missions in a variety of contexts, and changes nomenclature to adapt to customers and propagates these different nomenclatures to industry. This can lead to ambiguities We propose to promote standard nomenclature within ESA and towards other agencies. In particular, we propose to adopt the following definitions (see also section 1.1 of [GE2ERD] for other standard acronyms):

E2E = End-to-End
E2ES = End-to-End Simulator
GM= Geometry Module
SGM = Scene Generation Module
ISM= Instrument Simulation Module
ODGM= On-Board Data Generation Module
OSS = Observation System Simulator (to be preferred over the redundant OPSI, Observing Performance SImulator)
PEM= Performance Evaluation Module (same as PAM below)
PAM=Performance Assessment Module. If there is a need to distinguish the 'inner loop' that compares input radiances fed to the ISM and processed to Level 1 from the 'outer loop' that compares the input scene with the Level 2 retrievals, then we recommend using **PAM-L1** and **PAM-L2**.
CPM=Off-Line Calibration Processor Module. Missions where data are processed in the EUMETSAT ground segment generally have the offline calibration algorithms included in the (hybrid on/offline) Level 1 operational processor.
TDS= Test Data Set: full set of input and resulting output datasets with documentation and processor configuration information
IODS=Input-Output Data Specification, generated as part of the L1PP and CPM development. It describes content and initial format of the data used by the Prototype Processors.
PFS=Product Format Specification, which differs from the IODS in that it describes the format of the operational products that are produced in the ground segment and comes later. The geophysical and engineering parameters in both IODS and PFS are expected to be identical, as they both originate from the same algorithms. *NB: The development could be structured such the E2ES chain is compatible with both the initial IODS and the late PFS or that they become the same to allow direct exchange.*
L1PP = Level-1 Prototype Processor
L1OP= Level-1 Operational Processor
L2PP= Level-2 Prototype Processor
L2OP=Level-2 Operational Processor

It is suggested do avoid the redundant term GPP=Ground Processing Prototype (except for missions which have also an onboard Level 1 Processor). Also, it is suggested not to use the technically correct but excessively verbose, term of "Level-0-to-Level-1 Prototype Processor"

8 Annex B: Deliverable Items and Services List

ID	Item	Description	Notes
SW1	E2ES	Integrated packaged E2E Simulator (Binary and source code)	Includes: - Installer/uninstaller, - Framework, - Modules - Configuration data/files, (AUX, etc) - Input/output files
SW2	Modules: GM, simplified SGM ISM, ODGM,	Individual Modules component of E2ES as executable and source code	Includes: - Source code - Build scripts - Configuration data/files, (AUX, etc) - Input/output files
SW3	L1PP	Level-1 Processor Prototype and *on-line* calibration as executable and source code	Includes: - Source code - Build scripts - Configuration data/files, (AUX, etc) - Input/output files
SW4	PAM-L1	Performance evaluation for L1 as executable and source code	Includes: - Source code - Build scripts - Configuration data/files, (AUX, etc) - Input/output files Also input to the Monitoring Facility of the GS
SW5	scientific SGM*	scientific Scene Generator Module as executable and source code	Includes: - Source code - Build scripts - Configuration data/files, (AUX, RTM,etc) - Input/output files * Provided by L2 development
SW6	L2PP	Level-2 Processor Prototype as executable and source code	Includes: - Source code - Build scripts - Configuration data/files, (AUX, etc) - Input/output files
SW7	PAM L2	Performance evaluation for L2 as executable and source code	Includes: - Source code - Build scripts

			- Configuration data/files, (AUX, etc) - Input/output files Also input to the Monitoring facility of the GS
D1	L1 ATBD	Level-1 ATBD	Document. Includes both: - L1 processing ATBD - on-line calibration ATBD
D2	L1 IODS	Level-1 Input-Output Data Specification	Document (also called IODD)
D3	L2 ATBD	Level-2 ATBD	Document
D4	L2 IODS	Level-2 Input-Output Data Specification	Document (also called IODD)
D5	Scientific SGM ATBD	Scientific Scene Generator Module ATBD	Document
D6	L1 *off-line* calibration	L1 *off-line* calibration ATBD	Document
D7	Modules Design Specification	ECSS SDS for all Modules which component of E2ES of SW2	Document
D8	PAM-1 Definition	Specification of the algorithms to assess performance at L1	Document
D9	PAM-2 Definition	Specification of the algorithms to assess performance at L2	Document