

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

PE-RP-ESA-GS-0754  
Noordwijk, 08-03-2023  
V 1.2

## 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.

# 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 (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(1)

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

## 1.2 Reference documents

Ref	Title	Version
[GE2ERD]	Generic End-to-End Simulator and L1/L2 Processor Req. Document, PE-TN-ESA-GS-402	2.0
[EOFFST]	Earth Observation File Format Standard, PE-TN-ESA-GS-0001	3.0
[GENICD]	Generic E2E Simulator Interface Control Document, PE-ID-ESA-GS-0464	1.4.2
[EOCFI]	Earth Observation Mission Software CFI, <a href="http://eop-cfi.esa.int">http://eop-cfi.esa.int</a>	4.24
[RARCH]	EO E2ES Reference Architecture ARCHEO-TN-002	3.2
[OPENSF]	OpenSF software and Documentation, <a href="http://eop-cfi.esa.int/index.php/opensf">http://eop-cfi.esa.int/index.php/opensf</a>	4.1
[BIBLOS]	BIBLOS Technical Specification, <a href="https://gmv-biblos.gmv.com/">https://gmv-biblos.gmv.com/</a>	3
[GRAWL0]	EO generic RAW and L0 specification, PE-TN-ESA-GS-586	1.1
[DFDL4S]	DFDL4S binary R/W library <a href="https://eop-cfi.esa.int/index.php/applications/dfd4s">https://eop-cfi.esa.int/index.php/applications/dfd4s</a>	2.1
[ECL]	ESA Community licence, <a href="https://essr.esa.int/license/list">https://essr.esa.int/license/list</a>	2.4
[MSFFS]	Earth Observation Mission Software File Format Specification, PE-ID-ESA-GS-584	1.8

## 2 Contents

1	Purpose of the Document .....	2
1.1	Authors .....	2
1.2	Reference documents .....	3
2	Contents .....	4
3	Introduction to End-to-End simulator (E2ES) .....	6
3.1	Purpose of the E2ES in the Project life cycle .....	8
3.1.1	Phase O/A/B1: .....	8
3.1.2	Phase B2/C/D/E1 .....	8
3.1.3	Phase E2 .....	8
3.2	Top-level Generic Architecture of the E2E Simulator .....	9
3.2.1	The E2ES for Mission and System Requirement Verification .....	9
3.3	Responsibilities in the E2ES chain.....	10
3.3.1	Space Segment prime contractor .....	10
3.3.2	ESA Mission scientist .....	11
3.3.3	Scientific Algorithms’ developers .....	11
3.3.4	Instrument Performance engineer .....	11
3.3.5	Simulator and Processor Engineer.....	12
3.3.6	Ground Segment System engineer .....	12
4	Lessons Learned .....	13
4.1	Harmonisation of E2ES in ESA EOP .....	13
4.2	Technical and Procurement aspects.....	13
4.2.1	Requirements template for E2ES Mission Performance chain .....	13
4.2.2	Reference Architecture for Earth-Observation End-to-End simulators .....	14
4.2.3	Interfaces.....	14
4.2.4	E2ES Orchestration framework .....	14
4.2.5	File formats.....	15
4.2.6	Earth Observation Mission software CFI .....	16
4.2.7	Calibration and characterisation algorithms .....	16
4.2.8	Scene generation .....	16
4.2.9	Algorithms’ maturity .....	17
4.2.10	Terminology.....	17
4.2.11	Flow of Calibration and Characterisation data to the ground segment .....	18
4.2.12	Procurement organisation of L1 vs L2 E2E .....	19
4.2.13	Deliverable Items and Services.....	24
4.2.14	E2ES Deliveries and Development Process.....	24
4.2.15	SW environment preventive maintenance.....	25
4.2.16	Software Licensing.....	26
4.3	Organisation .....	28
4.4	2.2 E2ES components re-use decisions .....	29
4.4.1	Component re-use lifecycle.....	31
5	Development of the Data Processors.....	32
5.1	Development models for Prototype and Operational Processors.....	32
5.1.1	Discussion .....	32
5.2	Cost estimates for Procurement of the E2ES .....	34
6	Conclusion .....	35

7	Annex A: Terminology .....	36
8	Annex B: Deliverable Items and Services List .....	37

### 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.

A number of 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]

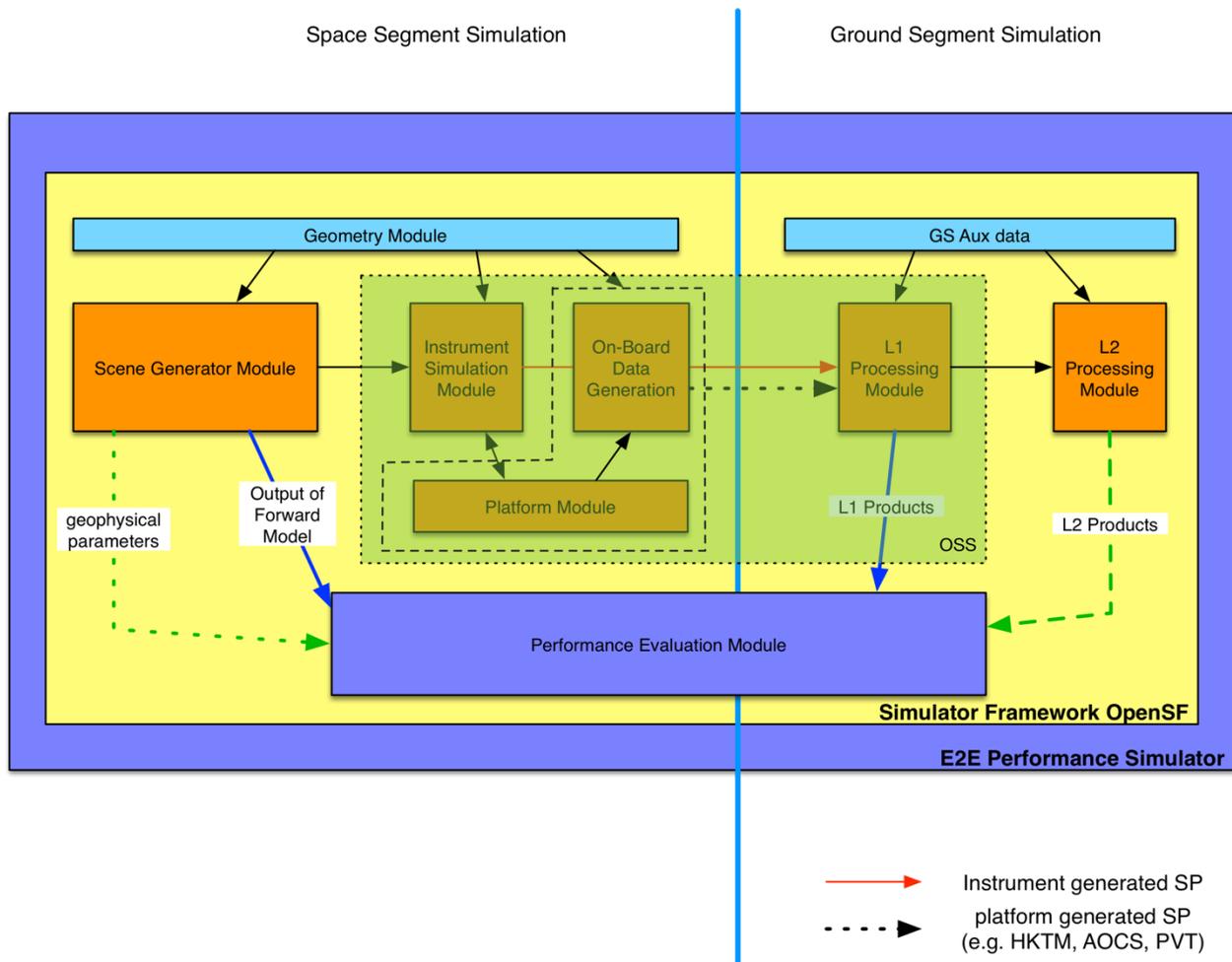


Figure 1 Generic layout of an End-to-End simulator with Scene Generator (SG), Instrument Simulator (IS), Level 1 processor (L1), Level 2 processor (L2), and Performance Assessment Model (PAM)



### **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. 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<sup>1</sup> and generates ancillary, measurements and calibration Level-1 data applying the necessary processing.

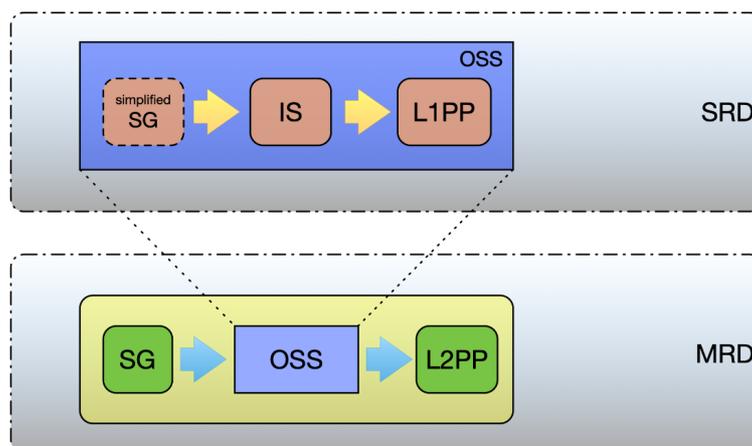
The Performance Assessment Module (PAM) confronts then input TOA stimuli with the outputs of the L1PP to perform sensitivity analysis and L1 performance verification (so called “*inner*” or Level 1 loop).

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.



<sup>1</sup> 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]

**Figure 2 E2EScomponent and verification level**

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)

### **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

In order 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 commonly responsible for the procurement and development of the instrument simulator and the L1 processor, which would enable it to demonstrate the fulfilment of the system requirements at Level 1. Noting the role that the Level 1 processing has to implement in the Ground Segment, the L1 algorithms also includes also the necessary calibrations and parameters for the level-2 processors which are not related to verification of Space Segment Level 1 requirements.

Although for some mission, ESA directly procured with a dedicated SW industry the elements of the E2ESchain (e.g. S5P, SMOS), the current common practice is to require that the L1 prototype and the instrument simulator are developed under responsibility of the prime contractor to also serve as either the operational Level 1 processor or its reference processor (whereby the detailed processing model and the test data generated by the prototype set the boundary conditions for independent development of the operational processor). The intention being that with this approach the necessary knowledge and data regarding instrument is directly available for E2E modelling and processing as well as to ensure that coherency between the developed algorithms used for space segment requirement verification with the ground segment processor implementation and that manpower of ESA side would be saved.

Considering the critical role that the chain plays in the Ground Segment, for a considerable number of recent missions (including the one with external partners like Eumetsat) this has instead led to undesirable results in part because an agile software development cycle is fundamentally different from a HW procurement one. Problems encountered included general lack of commitment, little or no flexibility, limited understanding of the Ground Segment needs, limitation in number of SW and data releases, inability to get quick bugfixes, slow turnaround and delays mostly resulting by the coupling with satellite milestones and paradoxically by the lack of competent manpower due to priority given to space segment development and testing activities, inadequate deliverables accepted by ESA just in order to complete major satellite milestone.

Furthermore, externally the limitation of this approach to satisfy timely the Ground Segment needs caused and causes consistently friction with the operational ground segment (be it either EOP-G or Eumetsat) and with the L2 scientific algorithm developers/institutions resulting, ultimately, in higher costs and more ESA management time required.

**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 in coordination with the Project to ensure mission objectives (generally at define at Level 2) are achieved. It convenes the (Mission Advisory Group) MAG and is the reference person for science and geophysical aspects. It also provides expertise and supports the Level 2 and Scientific Scene Generator development. In most missions is also the TO of the contract for procurement of Level 2 Processor prototype however it generally does not have a software expertise which therefore needs to be provided from within the Project team. To ensure coherency between the L1 and L2, seamless interfacing between products and adequate processing performance and SW quality of the Level 2 Processor Prototype and of the Scene Generator (SG), the Mission Scientist often (but not always) manages the contract with the help of the Ground Segment engineer or of 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 an ESA mission, with the highest exposure for the user community will be the L2 product. Typically, ESA places contracts with scientific algorithm's developers/institutions that are responsible for the definition and implementation of the L2 prototype algorithms and to support the review of L1 algorithms. Often, they are supported by software developers to build the prototype processor. In order to verify the entire performance chain up to L2, the algorithm's developers will have to generate geophysical realistic scenarios (as opposed to synthetic ones) by developing a dedicated Scene Generator (SGM).

### **3.3.4 Instrument Performance engineer**

The ESA Performance engineer within the ESA Project team is responsible of the Instrument performance. As such is/her primary role is to follow the HW procurement securing also the verification of Instrument level requirement performed by Space Segment prime. He/she is involved in the development of the E2ES up to Level 1 in order to ensure an adequate modelling of the instrument in the Instrument Simulator SW development and of the adequacy of L1 algorithms. The engineer

works hand-in-hand with the Simulator and Processor Engineer who is responsible of the SW engineering and coordination of the procured E2ESchain SW elements.

### 3.3.5 Simulator and Processor Engineer

The ESA (Simulator and) Processor engineer, within the Project team, has a strong software background and is responsible and focal point to support the definition, development, acceptance, testing and usage within the Project of the End-to End Mission Performance chain and of its components coordinating the various need into a coherent set of E2ES SW chain requirement to satisfy all needs. He/she provides dedicated support on the best practice of SW development working hand-in-hand with the Instrument Performance engineer at Level 1, with the Mission Scientist at Level 2 and with the GS engineer. It is responsible of the overall integration of the E2ES chain at L1 and at Level 2 either internally or by mean of industrial contract. It is also in charge of deploying the E2ES toolchain for use internally at ESA and enabling Performance Engineer and others to use it. It works together with the GS engineer to ensure that all needed deliverables to the operational GS are available in time and of adequate quality.

He/she manages the process of anomaly tracking wrt E2ES chain developers and the delivery of the E2ES chain to internal and external stakeholder. He/she can support multiple E2ES chains (e.g. in case of multiple instruments). He/she also ensure that consistent proven technical baselines and approach are applied coordinating with other similar development.

In a number of Project teams this critical role is not formally allocated/present and the duties above are either delegated to the GS Engineer, to the Instrument Performance Engineer or even outsourced to industry. This has resulted in inconsistent or unacceptable quality of deliverables.

#### 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 within the Project the requirements for the GS, its overall architecture, the interfaces with FOS and PDGS (or with relevant external entities) for definition and agreement of data and operational interface, define and implements the overall Ground Segment testing and provide general ground SW expertise within the project including for the E2ES Mission Performance chain if no dedicated Processor engineer are present in the Project team although this approach is not optimal as might lack hands-on low level SW experience to perform the 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 built on the inputs, experience and mistakes from missions already flying or under development and on activities and studies run by TEC, from participation to TEB, ITTs, technical progress meeting, and carefully compiled a set of general procurement practices and technical and architectural requirements that it recommends for cross-cutting application to all upcoming missions. To support this process EOP-PEP has furthermore taken over the operationalisation of SW simulation framework (openSF) originating in EOP-F, ensuring that internal and industry support is readily available to all missions, a seamless interfacing with the Mission Software CFI libraries for orbit and attitude calculation and overall end-to-end system coherence at architectural and framework level.

The various aspects are listed here below with corresponding recommendations.

Apart from these common elements, each mission needs additional solution, algorithms and requirements covering its specificity. Although also for those mission-specific solutions lessons have been learned and coordinated efforts between EOP and TEC are ongoing to develop generic processing SW component (BIBLOS, GRL, see [BIBLOS]), they are 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 (at the level of a ECSS Technical Specification) is available [GE2ERD] systematically covering all SW components of the reference architecture presented in Fig. 1 as well as all SW system aspects (e.g. interface, processing performance, log, dynamic behaviour, terminology, error handling, programming languages, installation, etc) and includes placeholder to the mission specific algorithms and ATBDs.

[GE2ERD] is designed to be tailored and should be used as the starting point for (a standardised set of) the E2ES Mission performance toolchain requirements to be included in the ITT. It has been used for several missions of both Copernicus and Earth Explorer and has made preparation work simpler and resulted in a structured set of proposals that were easier to evaluate and of higher quality.

#### **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 (termed ARCHEO [RARCH]) has been performed to define a reference architectures for End-to-End simulators, identifying commonalities across missions and also at a deeper level across mission families (grouped by active/passive sensing methods, frequency range etc.) whilst leaving flexibility for mission-specific needs. The generic User Requirement Document [GE2ERD] takes into account the ARCHEO reference architecture which has been the stepping stone for the ongoing development of open source SW components to be used for implementation of E2ES 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 has numerous disadvantages among others it prevents the direct integration in a common orchestration framework, it causes an increase in development and testing time and cost, causes vendor lock-in as the component cannot be swapped with functionally similar one without adaptation and makes comparison with similar missions difficult.

For this reason, a simple, language agnostic, generic and lightweight file-based software interface, the “ESA generic E2E ICD” [GENICD], has been defined to allow configurability while supporting standard invocation process, logging and data definition; by doing so it also simplifies:

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

Use of [GENICD] for the specification of the modules’ interfaces ensures out-of-the-box compatibility with the orchestration framework made available by ESA *openSF* and open source multi language SW 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

In order 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-development 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 subject of a long-term maintenance and development contract, it is continuously adapted in response to feedback from performance engineers and module/processor developers. It is used for all Earth Explorer missions presently in Phase 0/A/B1, and is also already used for several Phase B2/C/D/E1 missions (both Copernicus Sentinels and Earth Explorer).

It is to be noted that the use of openSF orchestration framework is not constraining for the development since in case need is identified it does not prevent to use and orchestrate directly from the command line any of the SW modules (e.g. via a script).

#### **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 (PDGS) 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/Satellite Prime have been seen in a number of cases like not fully adequate and demonstrating little commitment, this included lack of flexibility, of knowledge of the actual operation scenario or of the functioning of a ground segment especially (but not only) for external calibration (e.g. via transponders).

### **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 that needs 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

A number of Projects in EOP have started their E2ES Mission Performance Chain development with an incoherent nomenclature (across missions) for the E2ES components (e.g. same term for a different functionality or different name for the same or even new invented bespoke names) as well as for data Products (examples are OSS vs OPSI vs ISM vs FrontEnd, GPP vs L1PP/L2PP, GM vs GMd RAW vs L0, etc). This causes obvious problems with reuse of software and of documentation both for ESA as well (and particularly) for industry which is forced to name things differently in proposal, software and documents and is a source of 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.

##### **R-14.**

*Decouple space and ground segment responsibility and tasks 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 poses a challenge not only as the content and details of the simulation and processing increases but also as the responsibilities for L2 and L1 aspects are separated in different contracts. This section describes two different organisation modes aiming to ensure seamless integration and coordination to avoid ending up with two separate incompatible SW products: the L2 E2ES and the L1 E2E, that require additional effort for integration and pose the risk of late identification of discrepancies between the L1 and Space segment activities and L2 and scientific retrieval.

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

NB The approach where the complete (L1) E2ES is directly procured within the Satellite Prime contract is not discussed (as it is not recommended), however if programmatically necessary it can directly substitute step (b) below.

### 4.2.12.1 Integrated Approach

This is the favourite 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).

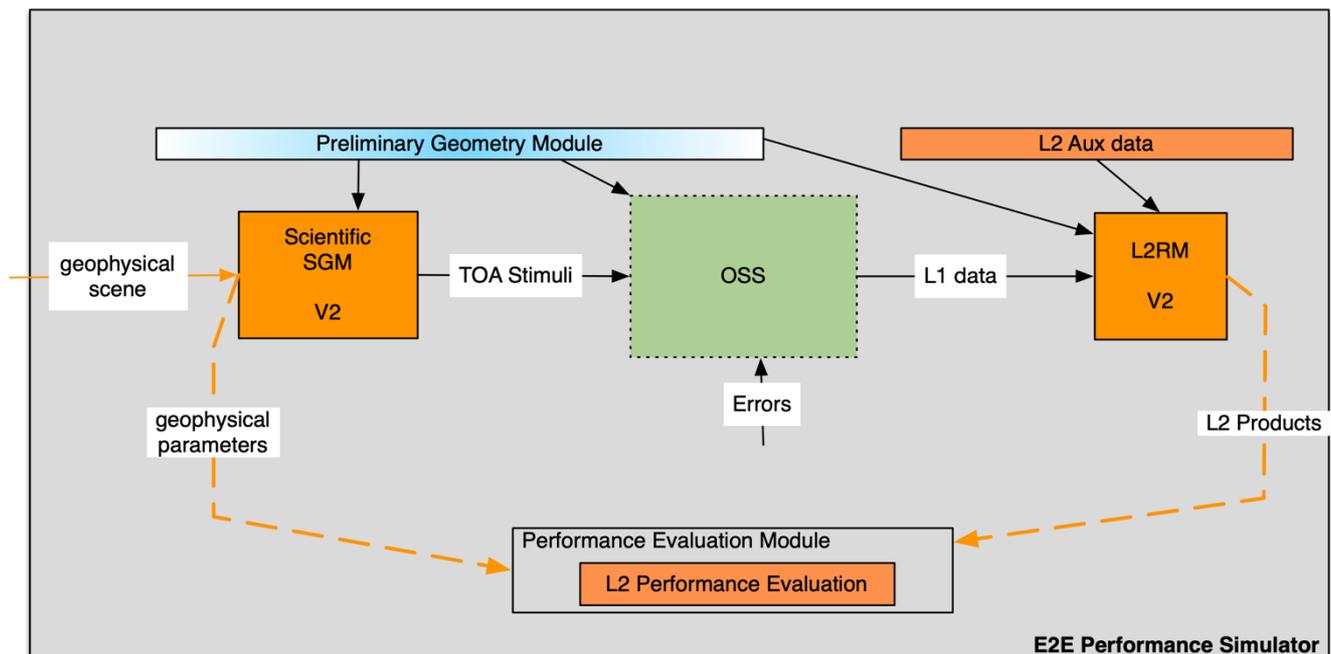


Figure 3 Evolution of L2 E2ES from Phase A/B1

- 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.

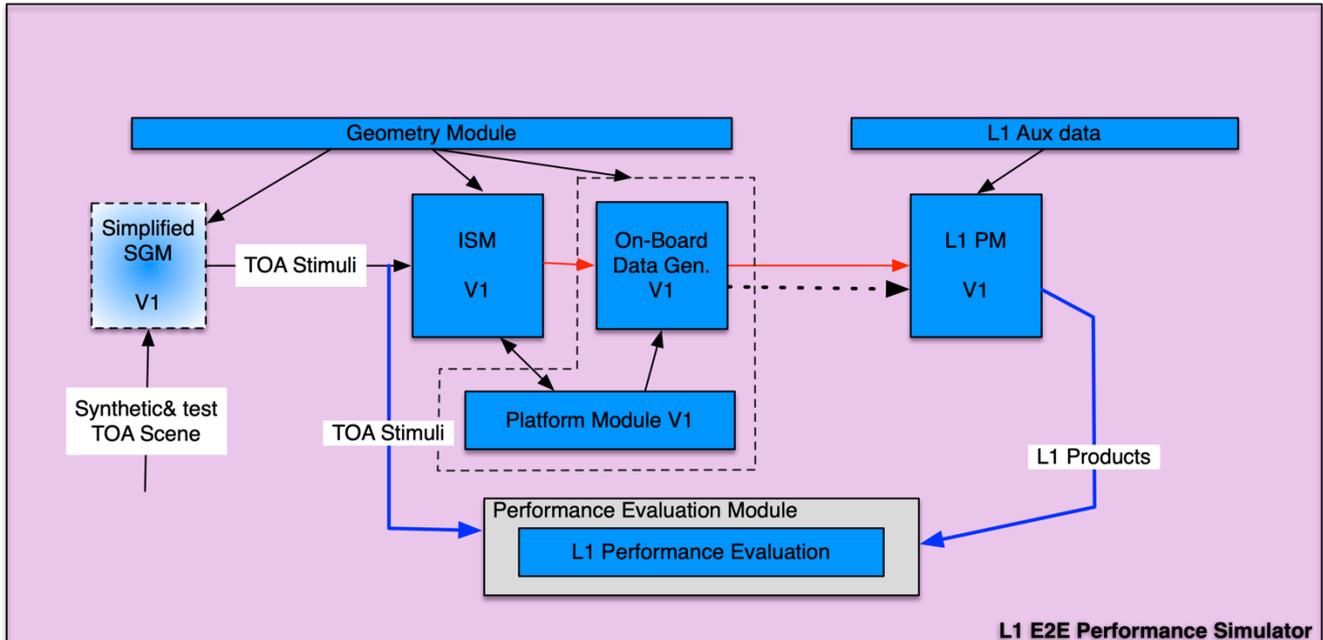


Figure 4 L1 E2ES SW Product

- 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.

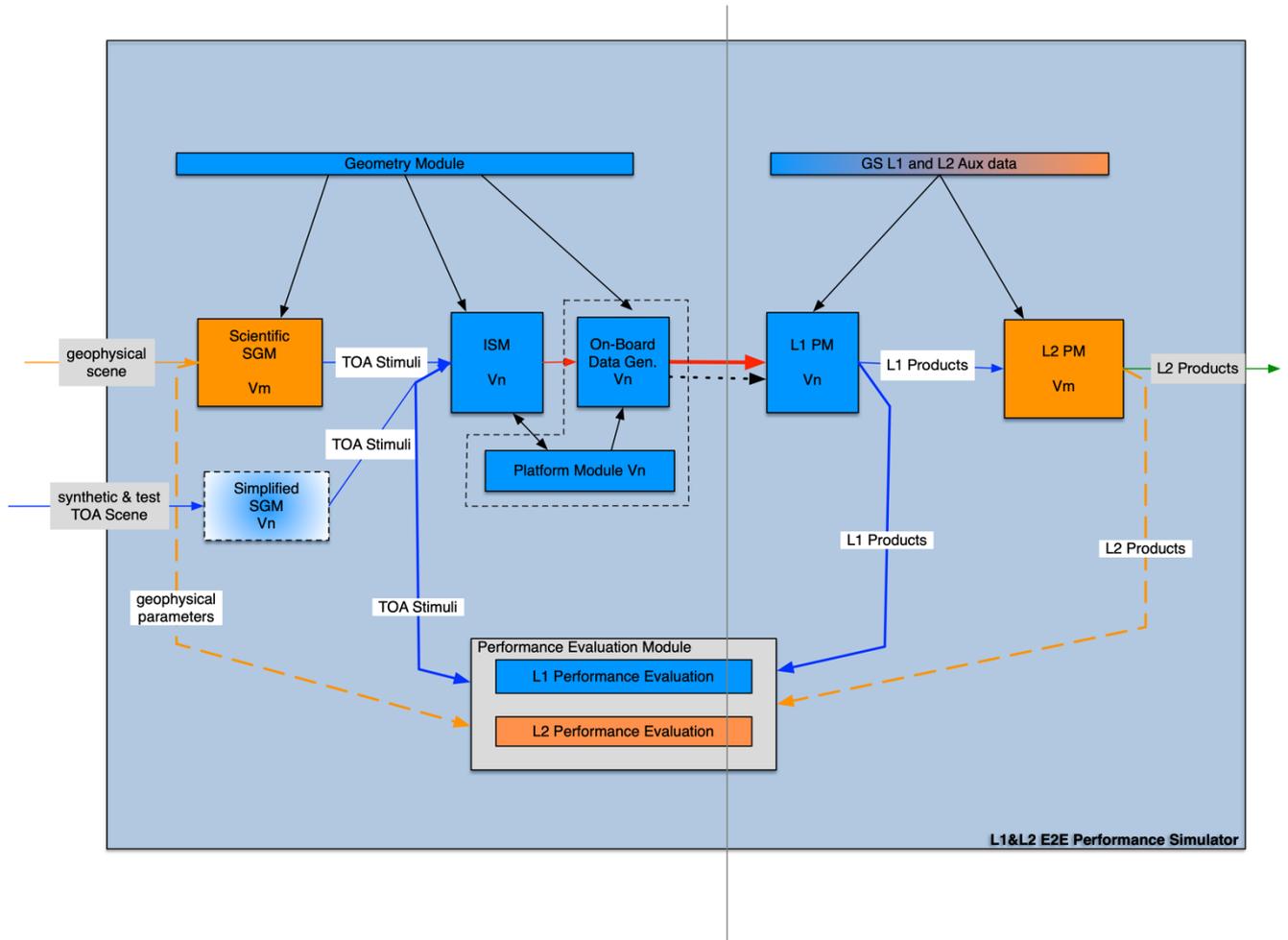


Figure 4 Integrated L1 and L2 E2ES (Blue L1, Orange Science/L2)

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 any moment 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 (depicted in next page Figure 5 Parallel L1 and L2 development approach) maintains 2 separate L1 E2ES and L2 E2ES contracts, SW products and development with an *off-line* coordination executed, not by producing a single E2ES L1 and L2 chain, but rather exchanging between the teams ad-hoc data (e.g. L1 product generated with the L1 provided to L2, or other TDS) and comparing and evaluate results and performance (at L1 and at L2) offline separately.

The advantage is that the 2 developments do not need to be coordinated (or only very loosely) and can be on different SW and data different baselines with different requirements, OS, logging, etc.

The disadvantage is that it allows/foresees only a limited manual test data set exchange of L1 data on specific scenarios (e.g. specified by L2), does not ensure an out-of-the box single stream verification/validation of the E2E performance that become laborious and is more affected by delay/dependency and cost increases for any change (e.g. interface) as they can only be evaluated/fixed by the other team.

This approach is workable but not recommended.

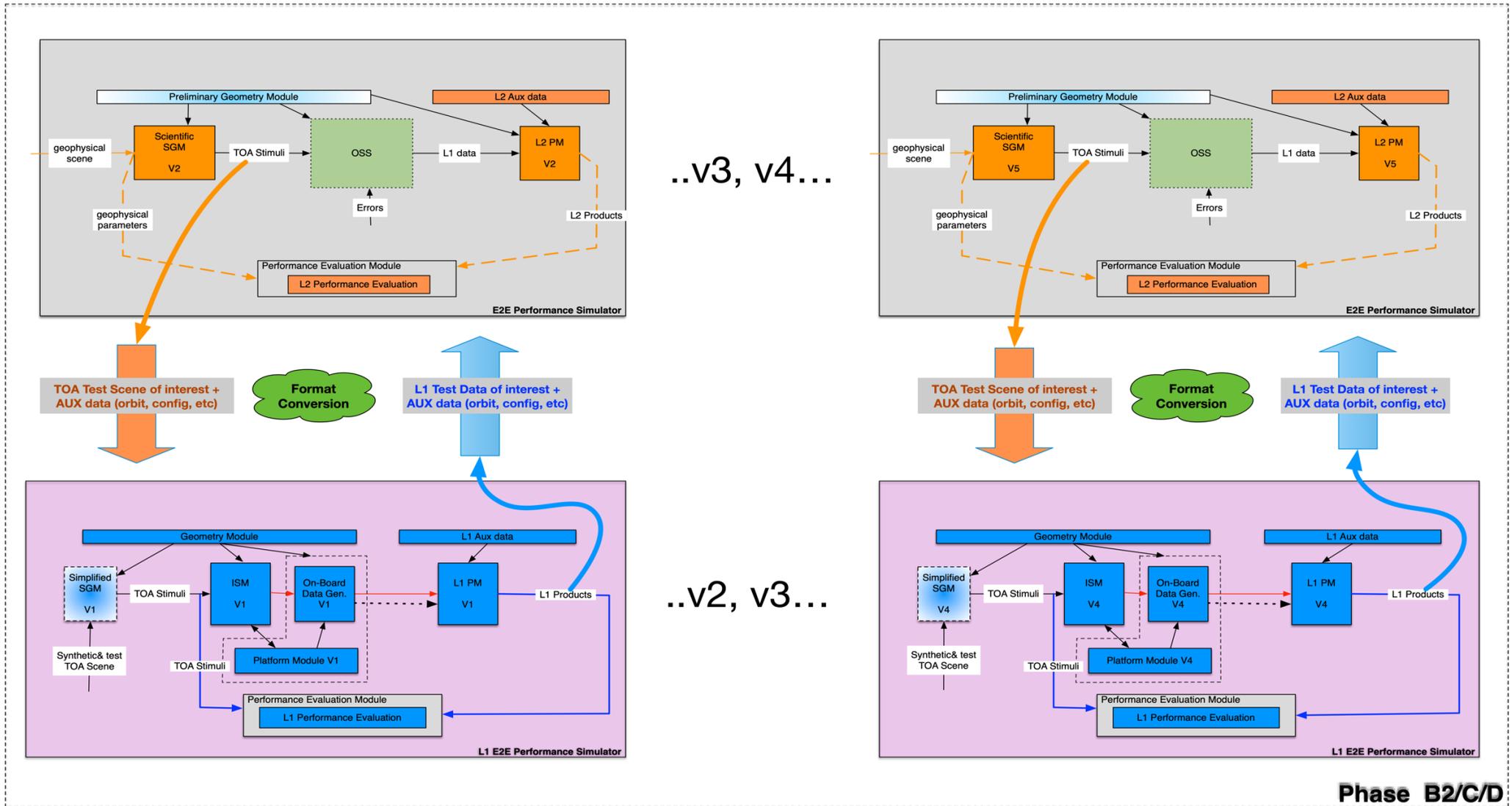


Figure 5 Parallel L1 and L2 development approach

#### 4.2.13 Deliverable Items and Services

There are different stakeholders relying on the timely availability of the E2E mission performance toolchain and of its component and generated data to perform their tasks. The timing and schedule of these actors is different and what is critical for one can be not so important for another. Stakeholders vary from the space segment designers, ground segment integrators, mission performance engineers and scientists. To this extent it is necessary to ensure that all elements developed as part of the E2ES toolchain are fully deliverable to ESA and to any external party with no restriction.

There have been cases where this was not the case due to incoherency of the DISL (which is contractually binding) and the task definitions (e.g. as specified in [GE2ERD]) so key elements of the E2ES were developed but not deliverables. This is typically a problem in case [GE2ERD] is not followed while the E2ES is procured via the satellite prime.

##### R-15.

*It is recommended to 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 definitions at the start of the contract of the SW deliveries for the E2ES and their date is speculative and experience has demonstrated that these deliveries need regularly to be changed/delayed to cope with requirements/inputs that are coming from different actors (Space Segment, Scientist, SW developers, Operational Ground Segment, etc). This negatively affects functionalities, cost and schedule so there is an interest in defining a mechanism that is built from start to flexibly provide a stream of deliveries when needed in response to external constraints.

Furthermore, while the exact schedule and number of deliveries is mission dependent, the E2ES users are different and with different schedules. It is therefore desirable to interleave formal deliveries with periodic intermediate ones, not hard-wiring the first delivery to the Satellite/Instrument Critical Design Review but defining instead intermediate ones well before, so that the Space Segment, the Ground Segment and the L2/Science can all be timely satisfied without dependencies/interlocks. It is finally critical to allow availability of deliveries to all parties also in potential absence of some missing/delayed input.

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 with 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:** Move away from a concept of Deliveries defined in advance by its pre-defined content to a timeline of Deliverables regularly spaced (typically every 6-9 months) that include and consolidate all inputs available at the moment in time.

The starting point is to have some of the E2ES deliverables dates associated to Space Segment Milestones and some related to Ground Segment/Science Milestones (see Figure 6 below) with the understanding that if some input is not available at a given point (e.g. an update of CCDB, or a product format) it will be considered for the next upcoming delivery while all the rest developed and tested at the point in time will be made available. Additionally, it will be possible if needed to introduce at any time additional informal deliveries with minimal effort (e.g. to fix critical bugs)

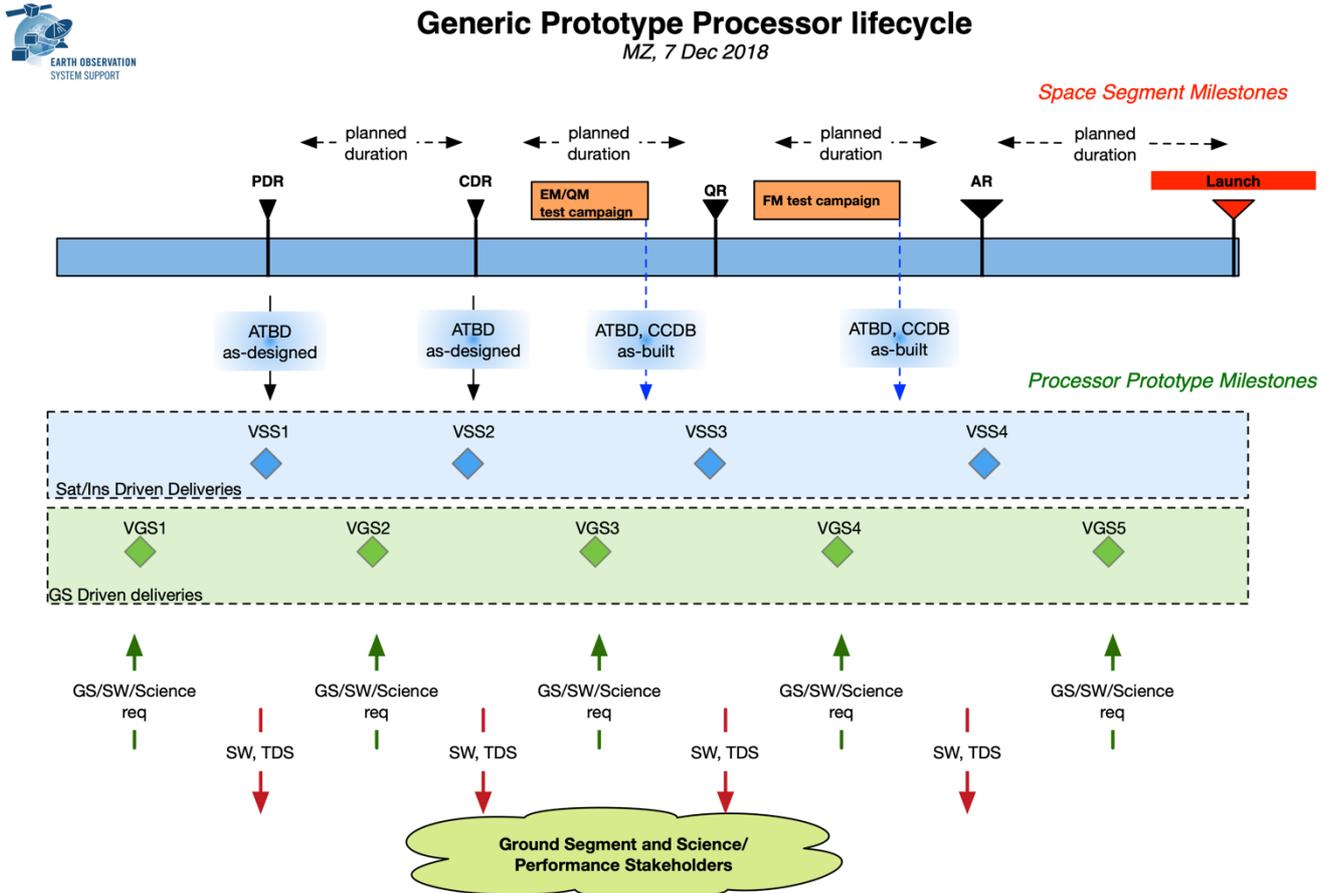


Figure 6 Processor deliveries' schema

#### 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 for development of on-board flight systems where a strong interest exists in protecting the industrial parties owning the IPR.

In the past clauses have been put to limit the distribution only to the Project itself, to the Programme (e.g. Copernicus or Meteo) or to specific parties.

This approach is inadequate to the needs and use of the E2ES mission performance simulator toolchain that needs to be made widely distributed to variety of internal stakeholder and that benefits, especially for the processing elements, of the scientific and engineering community contributions (and “free” troubleshooting) according to a modern open development model where budget is spent on added value functionality and refinement rather than on recoding over and over proprietary code.

The possibility of making freely available the developed software allows internal and external entities to easily exchange, modify and compare algorithms and data across missions as well as to form a starting step for new missions of similar kind being developed by allowing re-use with evident cost saving and de-risking of the activity, bringing in this way benefits from one EO Project to the next.

BIBLOS libraries [BIBLOS] are also being developed as open sources and will be made available by ESA, however this approach must be extended also to any new development and many new Earth Explorer development as well as some Copernicus has already included modified clauses to this extent. With this approach the IPR (or BIPR) can still reside with the developer as long as an open binary and source code permissive licensing for wide redistribution (i.e. for any Agency purpose also outside the specific program) and sublicensing is granted. In order to safeguard Member States interest and allow wide distribution ESA has developed a permissive software licence in the open-source spirit but limited to Member States, termed 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 #.

### 4.3 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 7 below.

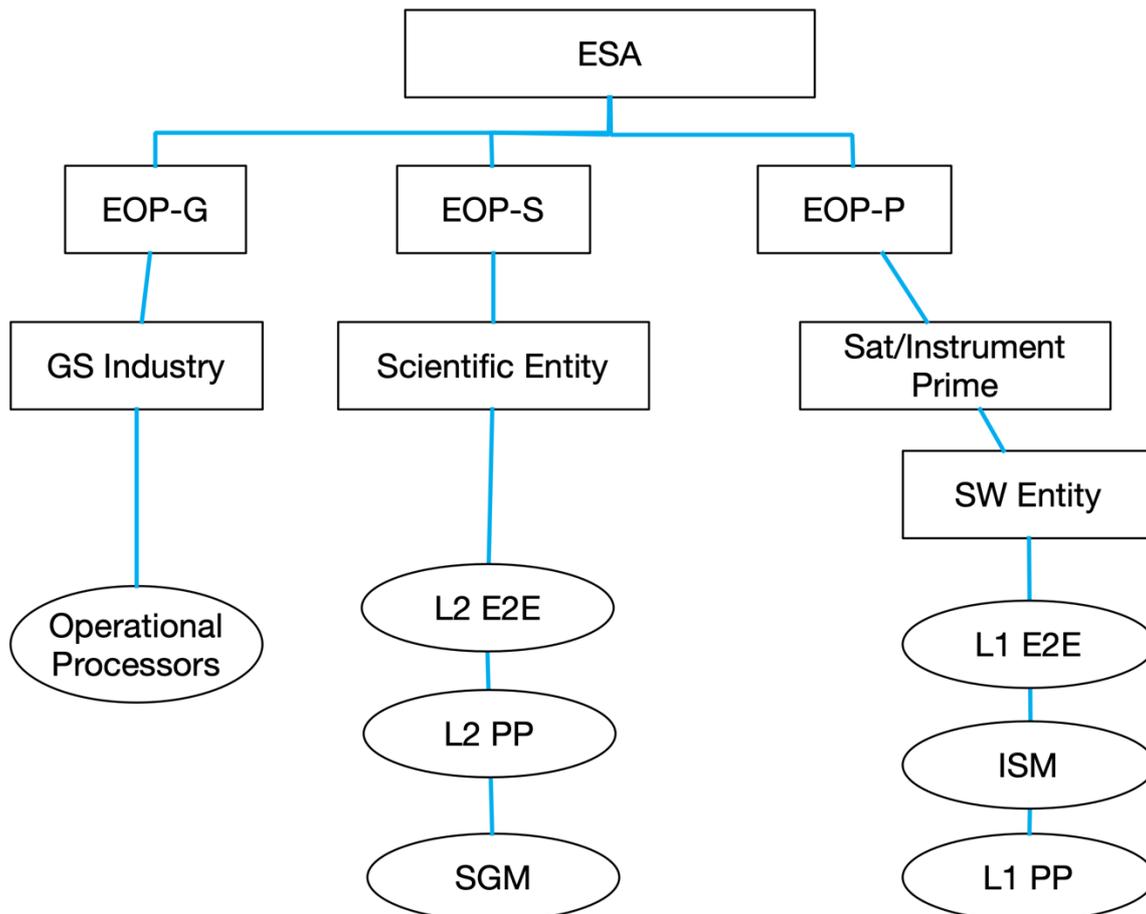


Figure 7 traditional organisation

Lesson learned from current EOP Projects has shown that :

- 1) Embedding the E2ES procurement within the main Space Segment contract **consistently** results in problems since the coordination with other actors including ESA, the GS and the Science Consortiums is not possible, the priorities are dictated mostly from the need of the Space Segment, there is little to no flexibility and the necessary system level ownership and expertise for E2ES is poor or not present. These result in delays, additional costs for coordination/reimplementation and marginally compliant technical solutions.
- 2) The development and quality of the L2 deliverables is not adequate and late; generally with no SW engineering practices/requirements present and with no or little handshake with the L1.

This has repeatedly resulted in late discovery of lacking SW, products and algorithm coherency between L1 and L2 within the E2E as well as in the operational GS e.g. wrt the L2PP

A revised organisation ensuring an effective coordination with L2/Science and avoiding these pitfalls is proposed.

**R-18.**

*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 8.*

**R-19.**

*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 8.*

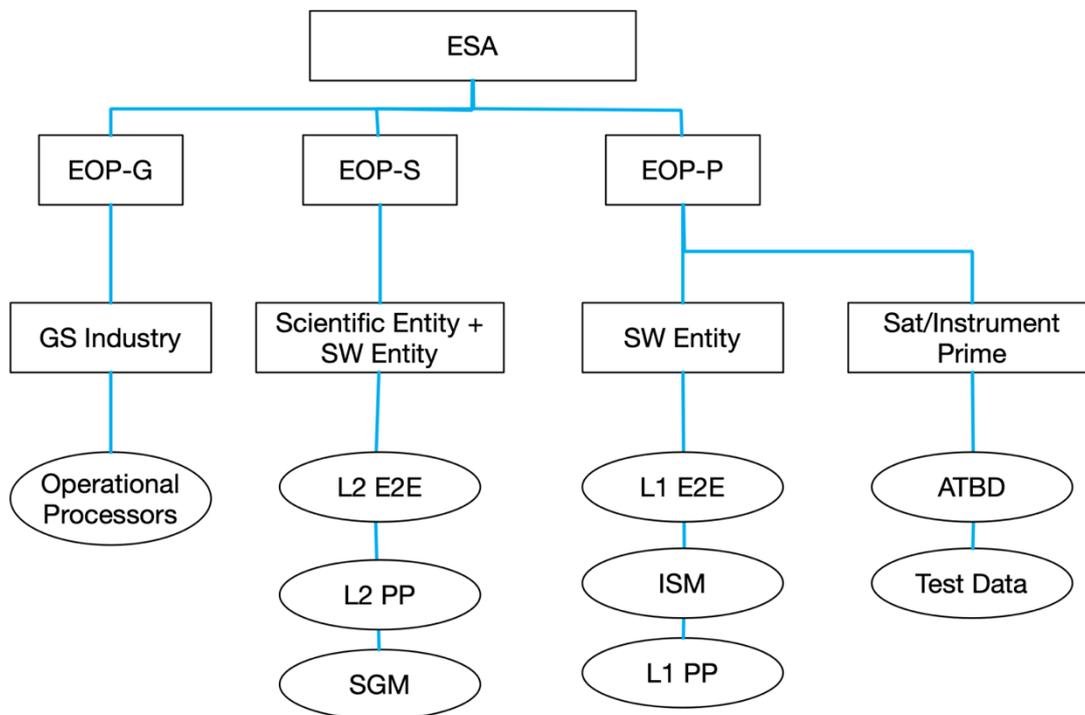


Figure 8 proposed E2E organisation

**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:

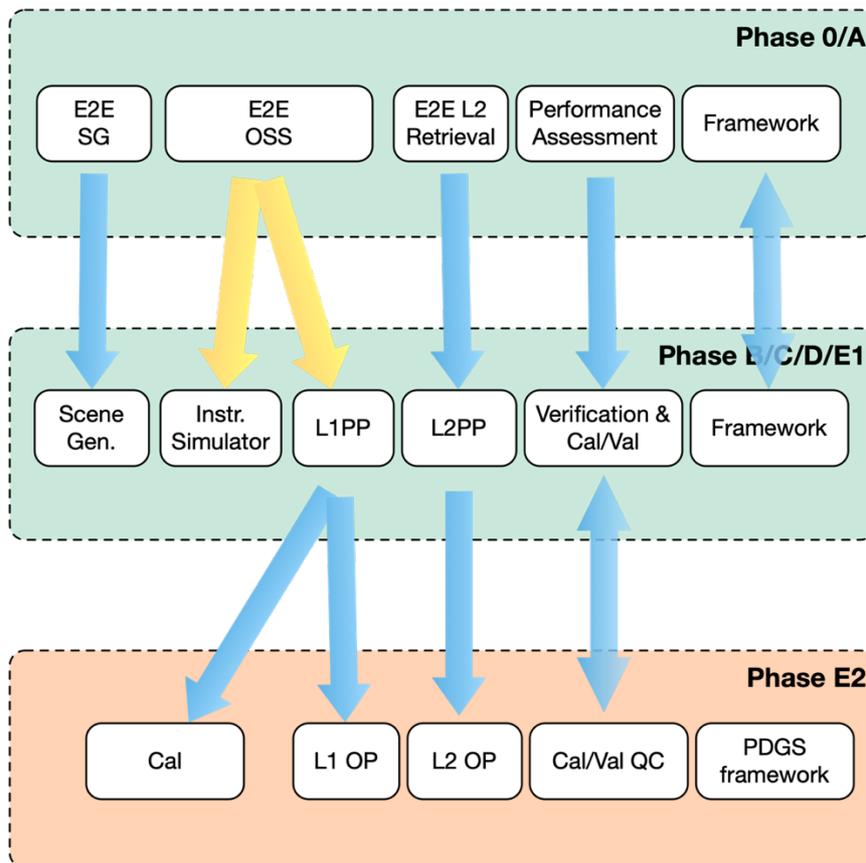
- Reuse of models and algorithms from Phase 0 to Phase E2.
- Reuse of models and algorithms within ESA projects.
- Need for automated operation.
- Need for automated input/output comparison, i.e., scenarios and expected L1/L2 output.
- Provision of in-house maintenance and support.

**R-20.**

*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 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 options is generally selected:

**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 (a special case of option 1 in 4.4.1 above, that is chosen to save time at the expense of independent verification).

#### 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 although constraints related to the selected industrial consortium play an important role as well as costs.

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 2 separate developments and the long and expensive update cycle from Prototype to Operational for all changes.
- 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.

e)

The disadvantages of the “**Evolution**” model wrt “**Cascade**” are:

- it requires an industrial contractor of high SW expertise in addition to algorithmic one in order 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).

We suggest to consider the **Evolution** model to be the default unless there are specific reasons not to.

Only in case that processors form the basis of an operational services requiring to execute within a specialised infrastructure (e.g. Message based interfaces, direct linking of the processor code within the infrastructure SW component) or hardware (GPU farm) then it might be more suitable to use the **Cascade** model.

The “**Hybrid**” model is a shortcut that is sometimes used to gain implementation time at the expense of independent verification and performance improvement.

The selection between the “**Cascade**” or “**Hybrid**” models are sometime driven by programmatic considerations e.g. when responsibility for the operational ground segment lies outside ESA, like EUMETSAT, independently of the level of algorithm consolidation in the domain that the mission addresses. An assessment addressing not only cost elements but also as technical risks of 2 (Cascade) vs 1 (Evolution) development and of the integration of a OP CFI into an externally provided infrastructure (standard vs non-standard) has to be performed.

We would recommend that even for innovative missions where the operational ground segment function is provided by another party than ESA, and the “Cascade” model is selected, it is critical that ESA arranges since the very beginning with the other party and with industry to allow for frequent updates of the Processing Baseline.

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 solid E2ES able to evolve during the whole lifecycle of the project (as opposed to be a one-off for mission selection) where unanimously judged as important contributions to success.

Our inventory has shown that unplanned transitions 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.

f)

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.

g)

**R-21.**

*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 important to ensure that the delivery from the space segment prime (if that is the approach selected) is adequate not only to meet its space-segment performance verifications obligations towards ESA, but also to verify that is suitable to meet its requirements as a reference for the Operational Processor (for processors' developments according to **Cascade** or **Hybrid** model) or suitable to be the actual Operational processor itself (**Evolution** model). To ensure relevant requirements are in place we recommend using [GE2ERD] and tailor it accordingly.

As mentioned in section 3.1 of [GE2ERD], the dual purpose of the space-segment E2ES delivery (both space-segment verification and ground segment reference) has in several instances led to adverse outcomes, as the prime contractor is a stakeholder in the payload performance verification, but not in the ground segment. Although not common practice yet, the example of SMOS has shown that parallel developments (one under responsibility of the space segment prime, one under ESA contract to expert teams) with the appropriate exchange of information (characterisation information) does not present such drawbacks (see Recommendation R-18)

## **5.2 Cost estimates for Procurement of the E2ES**

The costliest modules in the E2ES are the level-1 and level-2 data processors, with typically the cost of the level-2 processor significantly exceeding that of the level-1 processor for passive UV/VIS/NIR/TIR sensors, whilst the cost ratio is more balanced for active optical sensors and typically inverted for active MW.

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).

h)

\*= "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 writing 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 in E2ES have been introduced and discussed. Experience with harmonisation and common practices has been extensive in Phase 0/A/B1 and been applied to some missions in Phase B2/C/D/E1. This document presents the initiatives for further harmonisation and identifies lessons learned from the E2ES and processor development processes. Based on these lessons learned further best practices are proposed and corresponding templates and tools are provided, with the hope of achieving a ‘bottom up’ uptake. Continued feedback is welcomed to further develop all of the resources discussed and seek consensus on these recommendations when more missions are considered.

## 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: <ul style="list-style-type: none"> <li>- Installer/uninstaller,</li> <li>- Framework,</li> <li>- Modules</li> <li>- Configuration data/files, (AUX, etc)</li> <li>- Input/output files</li> </ul>
SW2	Modules: GM, simplified SGM, ISM, ODGM,	Individual Modules component of E2ES as executable and source code	Includes: <ul style="list-style-type: none"> <li>- Source code</li> <li>- Build scripts</li> <li>- Configuration data/files, (AUX, etc)</li> <li>- Input/output files</li> </ul>
SW3	L1PP	Level-1 Processor Prototype and <i>on-line</i> calibration as executable and source code	Includes: <ul style="list-style-type: none"> <li>- Source code</li> <li>- Build scripts</li> <li>- Configuration data/files, (AUX, etc)</li> <li>- Input/output files</li> </ul>
SW4	PAM-L1	Performance evaluation for L1 as executable and source code	Includes: <ul style="list-style-type: none"> <li>- Source code</li> <li>- Build scripts</li> <li>- Configuration data/files, (AUX, etc)</li> <li>- Input/output files</li> </ul> <p>Also input to the Monitoring Facility of the GS</p>
SW5	scientific SGM*	scientific Scene Generator Module as executable and source code	Includes: <ul style="list-style-type: none"> <li>- Source code</li> <li>- Build scripts</li> <li>- Configuration data/files, (AUX, RTM,etc)</li> <li>- Input/output files</li> </ul> <p><i>* Provided by L2 development</i></p>
SW6	L2PP	Level-2 Processor Prototype as executable and source code	Includes: <ul style="list-style-type: none"> <li>- Source code</li> <li>- Build scripts</li> <li>- Configuration data/files, (AUX, etc)</li> <li>- Input/output files</li> </ul>
SW7	PAM L2	Performance evaluation for L2 as executable and source code	Includes: <ul style="list-style-type: none"> <li>- Source code</li> <li>- Build scripts</li> <li>- Configuration data/files, (AUX, etc)</li> <li>- Input/output files</li> </ul> <p>Also input to the Monitoring facility of the GS</p>
D1	L1 ATBD	Level-1 ATBD	Document. Includes both: <ul style="list-style-type: none"> <li>- L1 processing ATBD</li> <li>- on-line calibration ATBD</li> </ul>

D2	L2 ATBD	Level-2 ATBD	Document
D3	Scientific SGM ATBD	scientific Scene Generator Module ATBD	Document
D4	L1 <i>off-line</i> calibration	L1 <i>off-line</i> calibration ATBD	Document
D5	Modules Design Specification	SDS for all Modules which component of E2ES of SW2	Document
D6	PAM-1 Definition	Specification of the algorithm to assess performance at L1	Document
D7	PAM-2 Definition	Specification of the algorithm to assess performance at L2	Document