N-FEE Simulator User Requirement Document

Prepared by:	Function:	Signature:
Philippe Plasson	N-DPU Application Software Project Manager (LESIA)	Signature numérique de Philippe Plasson Date : 2019.12.20 16:03:56 +01'00'
Loïc Gueguen	N-DPU Application Software GSE Manager (LESIA)	Gueguen Loïc 2019.12.20 16:07:21 +01'00'
Verified by:	**Function:**	**Signature:**
Vanderlei Parro	SIMUCAM Project Manager (MIT)	
Authorized by:	**Function:**	**Signature:**
Philippe Plasson	N-DPU Application Software Project Manager (LESIA)	Signature numérique de Philippe Plasson Date : 2019.12.20 16:04:14 +01'00'

CI No. 16130000

DRD No. N/A

Model BB, EM, EQM, FM, PFM, QM

l'Observatoire de Paris | LESIA

Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

CNRS-Observatoire de PARIS
Section de MEUDON – LESIA
5, place Jules Janssen
92195 Meudon Cedex – France

CHANGE RECORD

Issue	Rev.	Date	Author	Modifications
0	1	2017-03-28	Philippe Plasson	First draft
1	0	2017-12-15	Philippe Plasson	Added: - Section "Coordinate system" - NFEESIM-UR-858 Changed: - Section 2.1 - Section 3.3.1 - Section 4.6.2 - Section 5 - NFEESIM-UR-547, NFEESIM-UR-534, NFEESIM-UR-549, NFEESIM-UR-448, NFEESIM-UR-583, NFEESIM-UR-728, NFEESIM-UR-730, NFEESIM-UR-731, NFEESIM-UR-435, NFEESIM-UR-451, NFEESIM-UR-550, NFEESIM-UR-679, NFEESIM-UR-551, NFEESIM-UR-440, NFEESIM-UR-655, NFEESIM-UR-719, NFEESIM-UR-444, NFEESIM-UR-449, NFEESIM-UR-650, NFEESIM-UR-674, NFEESIM-UR-511, NFEESIM-UR-843, NFEESIM-UR-554, NFEESIM-UR-555, NFEESIM-UR-673, NFEESIM-UR-833, NFEESIM-UR-831, NFEESIM-UR-675, NFEESIM-UR-788, NFEESIM-UR-516, NFEESIM-UR-517, NFEESIM-UR-513, NFEESIM-UR-844, NFEESIM-UR-518, NFEESIM-UR-791, NFEESIM-UR-574, NFEESIM-UR-579, NFEESIM-UR-546, NFEESIM-UR-547 Deleted: - NFEESIM-UR-838
1	1	2018-09-18	Philippe Plasson	The label CALIBRATION used in the mode names has been changed in FULLIMAGE in order to be compliant to [AD2] and [AD5]. The label TEST used in the mode names has been changed in PATTERN in order to be compliant to [AD2] and [AD5].

Issue	Rev.	Date	Author	Modifications
				Added: - New explanations added to NFEESIM-UR-571 - A note has been added to clarify NFEESIM-UR-597 - NFEESIM-UR-902, NEESIM-UR-903, NEESIM-UR-907, NEESIM-UR-908, NEESIM-UR-909, NEESIM-UR-910 Updated: - NFEESIM-UR-407, NFEESIM-UR-440, NFEESIM-UR-481, NFEESIM-UR-449, NFEESIM-UR-458, NFEESIM-UR-532, NFEESIM-UR-555, NFEESIM-UR-576, NFEESIM-UR-621, NFEESIM-UR-643, NFEESIM-UR-644, NFEESIM-UR-645, NFEESIM-UR-673, NFEESIM-UR-704, NFEESIM-UR-788, NFEESIM-UR-791, NFEESIM-UR-830, NFEESIM-UR-833. Deleted: - NFEESIM-UR-728, NFEESIM-UR-729 - The sentence "for which the time-code value modulo 4 is equal to 0" has been removed from: NFEESIM-UR-449
2	0	2019-06-25	lgueguen	Main modifications - Mode evolution (new Mode on, removing partial readout mode, new trap pumping mode) - Modification of mode transitiion principle - Add partial readout feature in all modes - CCD order management - Warm-up simulation - Packet order list management - Requirement on naming convention - Error injection updated - Use of unused simulation for synchronization propagation to GSE Added: - NFEESIM-UR-925, NFEESIM-UR-928, NFEESIM-UR-929, NFEESIM-UR-930, NFEESIM-UR-931, NFEESIM-UR-932, NFEESIM-UR-933, NFEESIM-UR-934, NFEESIM-UR-935, NFEESIM-UR-936,

Issue	Rev.	Date	Author	Modifications
				NFEESIM-UR-937, NFEESIM-UR-938, NFEESIM-UR-939, NFEESIM-UR-943, NFEESIM-UR-944, NFEESIM-UR-945, NFEESIM-UR-946, NFEESIM-UR-947, NFEESIM-UR-948, NFEESIM-UR-951, NFEESIM-UR-953, NFEESIM-UR-954, NFEESIM-UR-962, NFEESIM-UR-971, NFEESIM-UR-972, NFEESIM-UR-973, NFEESIM-UR-974, NFEESIM-UR-975, NFEESIM-UR-976, NFEESIM-UR-977, NFEESIM-UR-978, NFEESIM-UR-979, NFEESIM-UR-982, NFEESIM-UR-983, NFEESIM-UR-985, NFEESIM-UR-986, NFEESIM-UR-989, NFEESIM-UR-990, NFEESIM-UR-992, NFEESIM-UR-993, NFEESIM-UR-994, NFEESIM-UR-996, NFEESIM-UR-997, NFEESIM-UR-999, NFEESIM-UR-1001, NFEESIM-UR-1002, NFEESIM-UR-1005, NFEESIM-UR-1006, NFEESIM-UR-1008, NFEESIM-UR-1050, NFEESIM-UR-1051, NFEESIM-UR-1052, NFEESIM-UR-1055, NFEESIM-UR-1056, NFEESIM-UR-1057, NFEESIM-UR-1064, NFEESIM-UR-1065, NFEESIM-UR-1066, NFEESIM-UR-1067, NFEESIM-UR-1070, NFEESIM-UR-1073, NFEESIM-UR-1074, NFEESIM-UR-1076, NFEESIM-UR-1078, NFEESIM-UR-1079, NFEESIM-UR-1080, NFEESIM-UR-1083, NFEESIM-UR-1085, NFEESIM-UR-1089, NFEESIM-UR-1094, NFEESIM-UR-1101, NFEESIM-UR-1103, NFEESIM-UR-1105, NFEESIM-UR-1106, NFEESIM-UR-1107, NFEESIM-UR-1109, NFEESIM-UR-1110, NFEESIM-UR-1112, NFEESIM-UR-1113, NFEESIM-UR-1115, NFEESIM-UR-1116, NFEESIM-UR-1117, NFEESIM-UR-1118, NFEESIM-UR-1119, NFEESIM-UR-1120, NFEESIM-UR-1121, NFEESIM-UR-1122, NFEESIM-UR-1125, NFEESIM-UR-1126, NFEESIM-UR-1135, NFEESIM-UR-1136, NFEESIM-UR-1135, NFEESIM-UR-1136, Updated: - NFEESIM-UR-444, NFEESIM-UR-447, NFEESIM-UR-449, NFEESIM-UR-458, NFEESIM-UR-481, NFEESIM-UR-485, NFEESIM-UR-511, NFEESIM-UR-512, NFEESIM-UR-516, NFEESIM-UR-517,

Issue	Rev.	Date	Author	Modifications
				NFEESIM-UR-518, NFEESIM-UR-534, NFEESIM-UR-549, NFEESIM-UR-579, NFEESIM-UR-615, NFEESIM-UR-621, NFEESIM-UR-629, NFEESIM-UR-630, NFEESIM-UR-631, NFEESIM-UR-632, NFEESIM-UR-650, NFEESIM-UR-655, NFEESIM-UR-673, NFEESIM-UR-674, NFEESIM-UR-675, NFEESIM-UR-791, NFEESIM-UR-831, NFEESIM-UR-843, NFEESIM-UR-844, NFEESIM-UR-902, NFEESIM-UR-910, . Deleted: - NFEESIM-UR-833, NFEESIM-UR-903
2	1	2019-11-20	lgueguen	Action 27336: The field "verification method" has been be filled. The section §5 "Deliveries and schedule" has been updated according to the last discussions with IMT. Postponed to R1: • NFEESIM-UR-405, NFEESIM-UR-474 Postponed to R1+: • NFEESIM-UR-447, NFEESIM-UR-925, NFEESIM-UR-655, NFEESIM-UR-631, NFEESIM-UR-632 and NFEESIM-UR-633, NFEESIM-UR-512, NFEESIM-UR-939, NFEESIM-UR-1065, NFEESIM-UR-455, NFEESIM-UR-456, NFEESIM-UR-548, NFEESIM-UR-600, NFEESIM-UR-659, NFEESIM-UR-419, NFEESIM-UR-723 and NFEESIM-UR-421. Postponed to R2: • NFEESIM-UR-931, NFEESIM-UR-938, NFEESIM-UR-425, NFEESIM-UR-546, NFEESIM-UR-547, NFEESIM-UR-1005, NFEESIM-UR-1006 and NFEESIM-UR-1008.

Issue	Rev.	Date	Author	Modifications
				Postponed to R2+: • NFEESIM-UR-932, NFEESIM-UR-933, NFEESIM-UR-934, NFEESIM-UR-939, NFEESIM-UR-951, NFEESIM-UR-953, NFEESIM-UR-954, NFEESIM-UR-962, NFEESIM-UR-971, NFEESIM-UR-972, NFEESIM-UR-973, NFEESIM-UR-974, NFEESIM-UR-975, NFEESIM-UR-976, NFEESIM-UR-977, NFEESIM-UR-978, NFEESIM-UR-979, NFEESIM-UR-982, NFEESIM-UR-984, NFEESIM-UR-1094, NFEESIM-UR-425, NFEESIM-UR-546, NFEESIM-UR-547, NFEESIM-UR-699, NFEESIM-UR-588, NFEESIM-UR-589 and NFEESIM-UR-590. Postponed to R3: • NFEESIM-UR-574. Postponed to R4: • NFEESIM-UR-475 and NFEESIM-UR-477, NFEESIM-UR-568, NFEESIM-UR-496, NFEESIM-UR-591, NFEESIM-UR-660, NFEESIM-UR-695, NFEESIM-UR-696, NFEESIM-UR-476, NFEESIM-UR-478, NFEESIM-UR-899, NFEESIM-UR-900, NFEESIM-UR-634, NFEESIM-UR-635, NFEESIM-UR-1066, NFEESIM-UR-1067, NFEESIM-UR-944, NFEESIM-UR-647, NFEESIM-UR-648, NFEESIM-UR-427, NFEESIM-UR-466 and NFEESIM-UR-465, NFEESIM-UR-418, NFEESIM-UR-420, NFEESIM-UR-481, NFEESIM-UR-480, NFEESIM-UR-571, NFEESIM-UR-1117, NFEESIM-UR-1118, NFEESIM-UR-1119, NFEESIM-UR-1120, NFEESIM-UR-1121, NFEESIM-UR-1122, NFEESIM-UR-989, NFEESIM-UR-990, NFEESIM-UR-992, NFEESIM-UR-993, NFEESIM-UR-994, NFEESIM-UR-996, NFEESIM-UR-997, NFEESIM-UR-999, NFEESIM-UR-1001, NFEESIM-UR-1002, NFEESIM-UR-

Issue	Rev.	Date	Author	Modifications
				726, NFEESIM-UR-727, NFEESIM-UR-469, NFEESIM-UR-470, NFEESIM-UR-724, NFEESIM-UR-467, NFEESIM-UR-468, NFEESIM-UR-693 Updated requirements: • NFEESIM-UR-423 acording PLGSC-157 • NFEESIM-UR-1118 acording PLGSC-174 • NFEESIM-UR-547 removing RMAP linked entries • NFEESIM-UR-1005 adding V-End max value • NFEESIM-UR-590 Echo on SpaceWire and not ethernet New requirements: • NFEESIM-UR-1164 • NFEESIM-UR-1166 • NFEESIM-UR-1167 Deleted : • NFEESIM-UR-702, NFEESIM-UR-703, NFEESIM-UR-704, NFEESIM-UR-706
2	2	2019-12-20	lgueguen	NFEESIM-UR-1078 updated. Adding the following registers to be simulated : • Trap_Pumping_Dwell_counter • Readout_pause_counter • Trap_Pumping_Shuffle_counter • ccd1_last_Epacket • ccd1_last_Fpacket • ccd2_last_Epacket • ccd2_last_Fpacket • ccd3_last_Epacket • ccd3_last_Fpacket • ccd4_last_Epacket • ccd4_last_Fpacket NFEESIM-UR-1079 and NFEESIM-UR-1080 updated

Issue	Rev.	Date	Author	Modifications
				(for readibility) NFEESIM-UR-788 and NFEESIM-UR-791 updated: the maximum number of windows per CCD has been corrected to be compliant with the N-DPU ASW SRS and memory budget (new value = 44123, old value = 41314)

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 The PLATO mission

PLATO ('PLAnetary Transits and Oscillations of stars') is a M-class mission of the European Space Agency's Science program Cosmic Vision 2015-2025 foreseen to be launched in 2026. PLATO aims to characterize exoplanetary systems by detecting planetary transits and conducting asteroseismology of their parent stars.

PLATO is the next generation planetary transit experiment; its objective is to characterize exoplanets and their host stars in the solar neighbourhood. While it builds on the heritage from CoRoT and Kepler, the major breakthrough to be achieved by PLATO will come from its strong focus on bright targets, typically with mV≤11. The PLATO targets will also include a large number of very bright and nearby stars, with mV≤8 (see also [RD1]).

The prime science goals of PLATO are:

- the detection and characterization of exoplanetary systems of all kinds, including both the planets and their host stars, reaching down to small, terrestrial planets in the habitable zone;
- the identification of suitable targets for future, more detailed characterization, including a spectroscopic search for biomarkers in nearby habitable exoplanets;
- a full characterization of the planet host stars, via asteroseismic analysis: this will provide us with the masses, radii and ages of the host stars, from which masses, radii and ages of the detected planets will be determined.

These ambitious goals will be reached by ultra-high precision, long (few years), uninterrupted photometry monitoring in the visible of very large samples of bright stars, which can only be done from space. The resulting high quality light curves will be used on the one hand to detect planetary transits, as well as to measure their characteristics, and on the other hand to provide a seismic analysis of the host stars of the detected planets, from which precise measurements of their radii, masses, and ages will be derived. For the brightest targets, planets are also expected to be detectable through the modulation of stellar light reflected on the planet surface, and/or through the astrometric wobble induced on the star by the planet orbital motion.

The PLATO space-based data will be complemented by ground-based follow-up observations, in particular very precise radial velocity monitoring, which will be used to confirm the planetary nature of the detected events and to measure the planet masses.

The full set of parameters of the systems with detected exoplanets will thus be measured, including all characteristics of the host stars and their orbits, radii, masses, and ages of the planets. Measurements of the radii and masses will be used to derive the planet mean densities and therefore will give insight on their internal structure and composition. The orbital parameters, together with the precise knowledge of all characteristics of the host star, will enable us to estimate the temperature and radiation environment of the planets. Finally, the knowledge of the age of the exoplanetary systems will allow us to put them in an evolutionary perspective.

See [RD1] for further details on the PLATO mission.

1.2 The PLATO payload

1.2.1 Cameras and FEE

The PLATO payload is made up of 24 normal camera (N-Camera) acquiring the data at the cadence of 25 seconds and 2 F-Camera acquiring the data at the cadence of 2.5 seconds.

Each N-camera has a focal plane array (FPA) made up four 20.3-million pixel CCD and is responsible for the acquisition of about 115000 stars. Each N-camera is controlled by a N-FEE which is responsible for performing the following operations:

- pixel read-out
- pre-windowing on the data stream
- HK acquisition
- data transmission via a SpaceWire link

Each F-camera is responsible for the acquisition of about 600 stars. The F-camera are involved in the loop of the fine guidance system. Each F-camera is controlled by a F-FEE having similar functions to the N-FEE.

1.2.2 Data Processing System (DPS)

1.2.2.1 DPS overview

The PLATO Data Processing System, called DPS, is the PLATO sub-system in charge of the on-board data processing (data acquisition, data reduction, data compression, monitoring, etc.). The DPS is a set of several on-board computer boards connected via a SpaceWire network.

The DPS logical architecture is composed of:

- 12 N-DPU fully independent from one another,
- 2 F-DPU,
- 2 ICU.

Both N-DPU and F-DPU are linked via SpaceWire interfaces to respectively, the N-FEE and the F-FEE at one end and to the ICU at the other end.

The N-DPU and F-DPU are in charge of processing respectively the data of two normal cameras for each N-DPU and the data of one fast camera for each F-DPU.

The ICU are working in cold redundancy and are in charge of managing the payload, the communication with the Service Module (SVM) and the compression of scientific data before transmitting them as telemetry to the SVM.

1.2.2.2 N-DPU

During the Observation mode, each N-DPU unit is receiving window segments of observed stars from the two N-FEE as inputs. The window segments are transferred every 6.25 seconds (CCD exposure time is 25 seconds, but the 4 CCD are read sequentially every 25/4 = 6.25 seconds) through a single SpaceWire link. The windows segments are then reconstructed back into windows and are processed by the N-DPU ASW.

Once the star windows are reconstructed, the N-DPU ASW performs the following tasks:

- Offset, background and smearing correction.
- LC and centroids computation.
- Outlier detection and time averaging of the LC and centroids.
- Periodic update of the photometric masks (which requires to compute periodically the telescope attitude).

The science products (LC, centroids, raw imagettes) and housekeeping reports are transmitted as TM packets to the ICU. The processing cadence for N-DPU is 25 seconds (4 CCD are processed over 25 seconds).

1.2.2.3 F-DPU

There are 2 F-DPU units. Each F-DPU is linked to one F-FEE by way of 4 SpaceWire links in parallel. The main role of the F-DPU is to compute quaternions from star observations in order to perform attitude determination of the spacecraft.

The F-FEE sends to the F-DPU windows segments that are reconstructed back by the F-DPU software. The windowed stars are then used to compute quaternions for the AOCS to perform orbital and attitude control.

The windowed stars are also transferred as raw imagettes to the ICU to be downlinked to the ground. The processing cadence for the F-DPU is 2.5 seconds.

1.2.2.4 ICU

Regarding ICU, there are two ICU: a main ICU unit and a redundant ICU unit. The 2 ICU channels work in cold redundancy that can work independently or with crossed configuration. On a logical scope, the ICU is in charge of:

- Managing the communication with the spacecraft (SVM) through a SpaceWire link.
- Collecting the scientific data and HK from the N-DPU and the F-DPU via a SpaceWire interface.
- Managing payload commanding and monitoring through a SpaceWire network.
- Providing mode and status information of the ICU and the entire payload.
- Compressing scientific data before transferring to SVM.

The incoming scientific data is compressed with a factor 2 in the ICU. The compressed scientific data and the uncompressed HK telemetry from the entire payload are sent to the ground via a SpaceWire interface with the SVM at a rate no greater than 435 Gb/day.

1.2.2.5 DPS architecture

The figure below details the physical architecture of the DPS:

- The 12 N-DPU units are gathered in 2 MEU boxes which contain also two SpaceWire router units and one power supply unit.
- The 2 F-FPU units are gathered in 1 FEU box which contains also one power supply unit.
- The 2 ICU units contain one SpaceWire router sub-unit, one compression sub-unit, one processor sub-unit and one power supply unit.

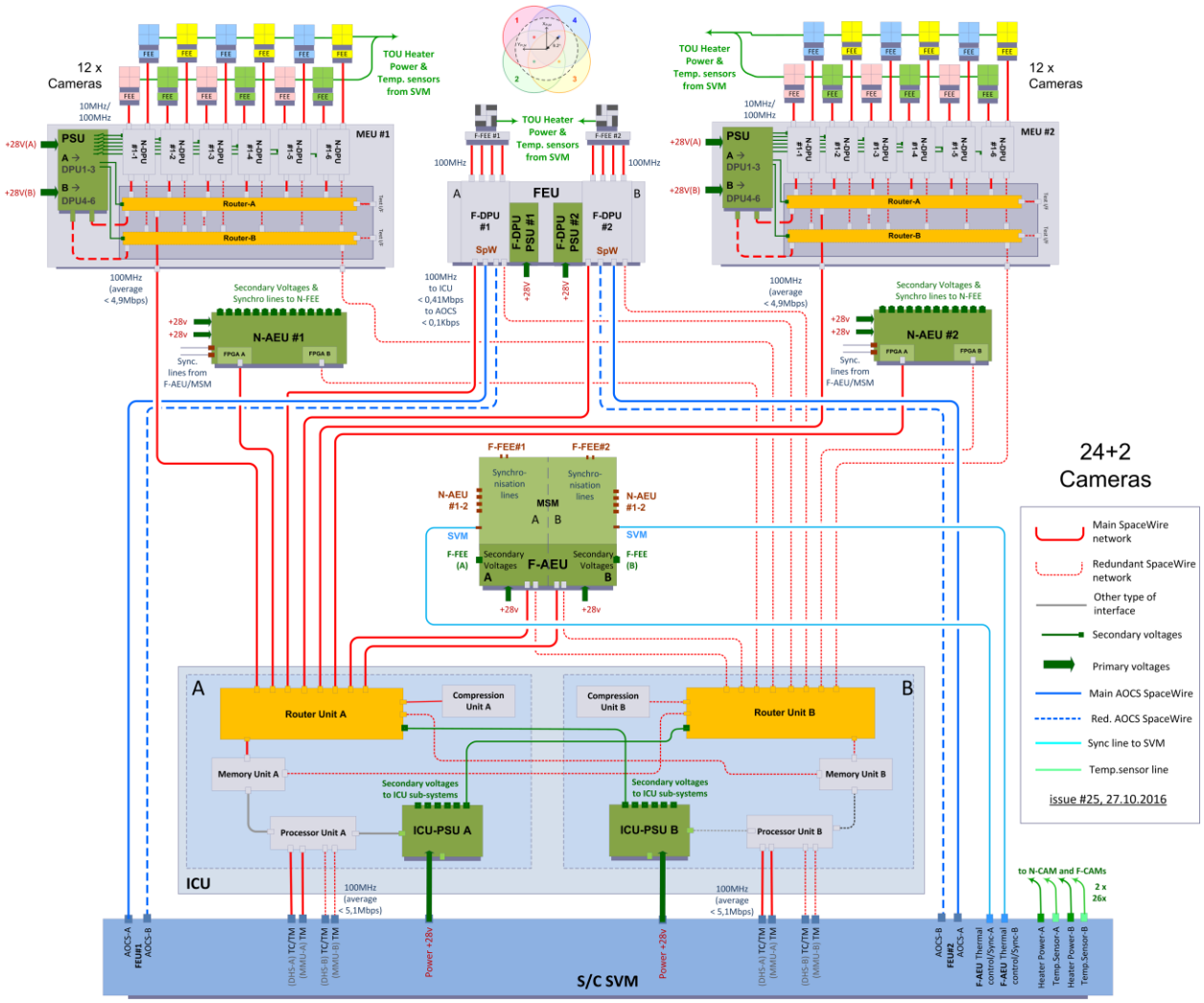

Figure 1 - PLATO DPS physical architecture

1.3 Purpose of the document

The validation process of the N-DPU / F-DPU / ICU Application Software, the DPS AIT activities and the payload AIT activities will require to have some powerful GSE and test systems. Among the various simulators which have to be produced, the N-FEE simulator will play a crucial role. The N-FEE simulator will allow to replace, during the test activities, a real N-FEE. In particular, the N-FEE simulator will allow to inject some test data produced by the DPA team in charge of the on-board algorithm studies and will be used for validating all the on-board algorithms implemented inside the N-DPU. The N-FEE simulator will allow also to perform some performance tests and to simulate failure cases in the context of robustness tests.

This document is the user requirement specification of the PLATO N-FEE simulator.

1.4 Application

This document shall be used as an applicable document for the development of the PLATO N-FEE simulator.

1.5 Requirement identification

The content of this document is prepared in and exported from an IBM DOORS module (/PLATO/DPS/MEU/N-DPU/DAS/SGSE/NFEESIM/NFEESIM-URD).

To identify the requirements written in this document each requirement has an unambiguous ID "NFEESIM-UR-#" which will not change during the project lifetime.

2. Reference documents

2.1 Applicable documents

	Title	Document Reference
AD1	N-FEE User Requirements Document	PLATO-OHB-PL-RS-0005
AD2	PLATO N-FEE to N-DPU Interface Requirement Document (IRD)	PLATO-DLR-PL-ICD-0010
AD3	SimuCam pattern requirement	PLATO-LESIA-PL-TN-023
AD4	PLATO CCD Definition	PLATO-MSSL-PL-TN-008
AD5	PLATO N-FEE ICD	PLATO-MSS-PL-ICD-0002
AD6	PLATO N-FEE Register map	PLATO-MSSL-PL-Register map

2.2 Applicable ECSS Standard

	Title	Document Reference
ES1	Software	ECSS-E-ST-40C
ES2	Ground systems and operations - Telemetry and telecommand packet utilization	ECSS-E-70-41A
ES3	SpaceWire - Links, nodes, routers and networks	ECSS-E-ST-50-12C
ES4	SpaceWire protocol identification	ECSS-E-ST-50-51C
ES5	SpaceWire - Remote memory access protocol	ECSS-E-ST-50-52C
ES6	SpaceWire - CCSDS packet transfer protocol	ECSS-E-ST-50-53C
ES7	Software product assurance	ECSS-Q-ST-80C

2.3 Reference documents

	Title	Document Reference
RD1	PLATO Mission Requirements Document	PLATO-ESTEC-MIS-RS-001, Issue 4.1
RD2	PLATO Instrument Design report	PLATO-OHB-PL-DD-001, Issue 2
RD3	FEE Windowing Technical Note	PLATO-DLR-PL-TN-018, Issue1
RD4	N-DPU ASW GSE Software System Specification	PLATO-LESIA-PL-SP-0002
RD5	PLATO (on-board) Software System Specification (SSS)	PLATO-DLR-PL-RS-0006
RD6	MEU User Requirement Specification	PLATO-DLR-PL-RS-0003

RD7	N-DPU Application Software Requirement Specification (SRS)	PLATO-LESIA-PL-SP-0001
RD8	PLATO N-DPU ASW Data Rate and Memory Budget (B Phase)	PLATO-LESIA-PL-RP-0031
RD9	Fullimage acquisition scenario	PLATO-LESIA-PL-TN-0046
RD10	Window acquisition scenario	PLATO-LESIA-PL-TN-0049
RD11	F-FEE Emulator (SimuCam) Requirement Specification	PLATO-DLR-PL-RS-0010

2.4 Acronyms

AD	Application Document
AEU	Ancillary Electronic Unit
AOCS	Attitude and Orbital Control System
APID	Application Process IDentifiers
ASW	Application Software
CCD	Charge-Coupled Device
CCSDS	Consultative Committee for Space Data Systems
COB	Centre Of Brightness
CPU	Central Processing Unit
DPA	Data Processing Algorithms
DPS	Data Processing System
DPU	Data Processing Unit
DSU	Debug Support Unit
ECSS	European Cooperation for Space Standardization
EDAC	Error And Detection Correction
EEPROM	Electrically-Erasable Programmable Read-Only Memory
ESA	European Space Agency
F-AEU	Fast Ancillary Electronic Unit
F-DPU	Fast Data Processing Unit
F-FEE	Fast Front End Electronics
FPU	Floating Point Unit
FIFO	First In First Out
FT	Fault Tolerant
GRFPU	Gaisler Floating Point Unit

GRSPW2	Gaisler SpaceWire IP core version 2
HK	HouseKeeping
ICU	Instrument Control Unit
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
kB	Kilo Byte. 1 kB = 1024 bytes
LC	Light Curve
LESIA	Laboratoire d'Etude Spatiale et d'Instrumentation en Astrophysique
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MB	Mega Byte. 1 MB = 1024 kB = 1048576 bytes
MEB	Main Electronic Box
MEU	Main Electronic unit
MU	Memory Unit
N-DPU	Normal Camera DPU
N-FEE	Normal Front End Electronics
OBCP	On-Board Control Procedure
PLATO	PLAnetary Transit and Oscillations
PROM	Programmable Read-Only Memory
PSU	Power Supply Unit
PU	Processing Unit
PUS	Packet Utilization Standard
RAM	Random Access memory
RD	Reference Document
RMAP	Remote Memory Access Protocol
RU	Router Unit
Rx	Reception
SDP	Software Development Plan
SDRAM	Synchronous Dynamic Random Access Memory
SoC	System on a Chip
SRS	Software requirement Specification
SSS	Software System Specification

TBC	To Be Confirmed
TBD	To Be Determined
TBW	To Be Written
TC	Telecommand
TM	Telemetry
UUID	Universally unique identifier

3. General description

3.1 N-FEE simulator and SimuCam

In order to reduce the number of simulators to be developed, it has been decided to have a common design for the N-FEE simulator and the F-FEE simulator: the same simulator hardware will be used for the N-chain activities and the F-chain activities. This common simulator is called 'SimuCam' and is developed by the Maua Institute of Technology (IMT) in Brazil, São Paulo.

SimuCam is a system allowing to handle simultaneously up to 8 SpaceWire links. It will allow to manage the simulation of up to 8 N-FEE in one box (or 2 F-FEE). We give in this chapter the N-FEE simulator user requirements which are applicable to SimuCam. This document, which is restricted to the PLATO N-FEE simulation capability, doesn't cover the PLATO F-FEE simulation requirements (another URD will be available for the user requirements related to the F-FEE simulation capability).

3.2 N-FEE simulator use cases

The various use cases of the SimuCam N-FEE simulators are listed below:

- Use case #1: N-DPU ASW development and validation process (user = LESIA)
- Use case #2: MEU functional and performance tests (user = LESIA + IAA)
- Use case #3: DPS AIT (user = DLR)
- Use case #4: Payload AIT (user = OHB + DLR)

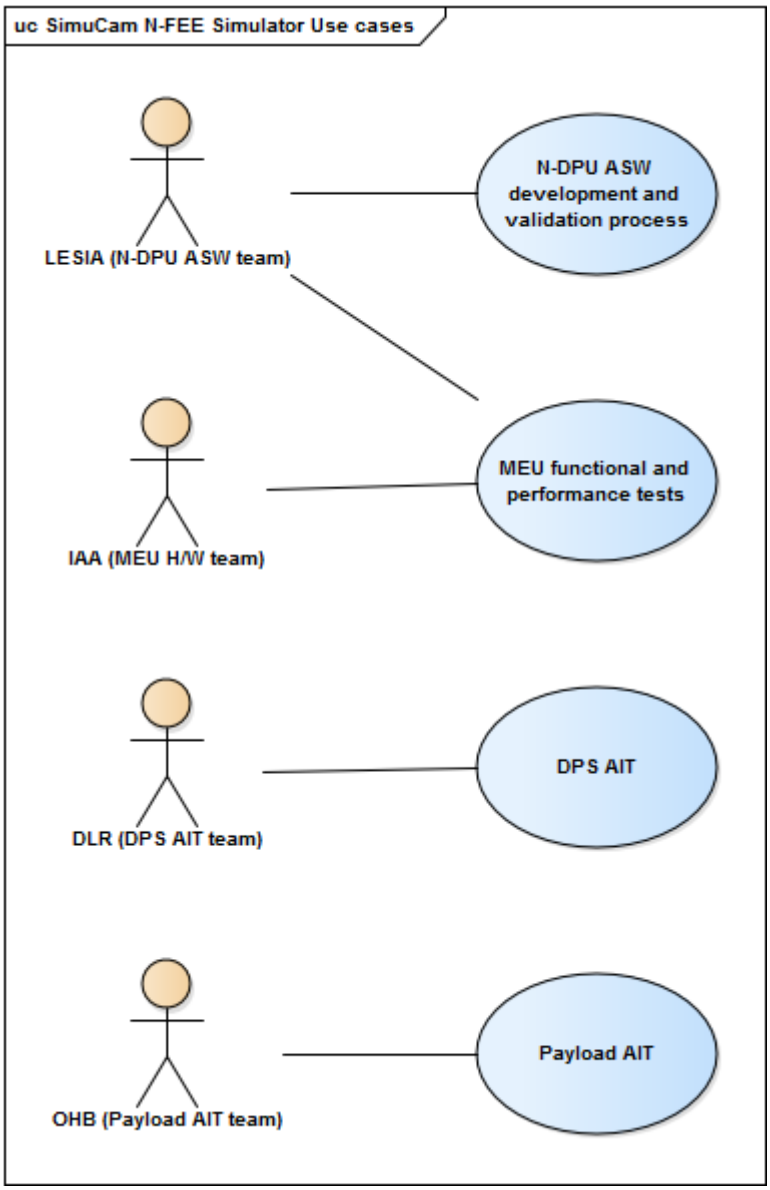

Figure 2 - SimuCam N-FEE simulator use cases

3.3 N-FEE simulator requirement overview

This section gives a quick overview of the main requirements of the N-FEE simulator. The detailed definition of the capability requirements is given in the chapter 4.

3.3.1 N-FEE functional simulation

SimuCam shall allow to simulate the N-FEE functional modes:

- Standby mode
- Calibration mode (full frame image mode)

- Windowing mode
- Partial readout mode

In the windowing and full frame image mode, the N-FEE simulators managed by the SimuCam transmit fullimages or window segment streams to the N-DPU via the SpaceWire link by fulfilling all the timing requirements of a real N-FEE and all the protocol requirements defined between the N-DPU and the N-FEE (see [AD1], [AD2] and [AD5]).

3.3.2 Simulation data feeding modes

SimuCam shall allow to feed the data sent to N-DPU (windows or full-frame images) with various types of data:

- Patterns (see [AD3]).
- Fullimages stored in SSD
- Window stacks stored in SSD

The pattern feeding mode will be used mainly for low level test cases, basic coupling tests or for performance tests (load tests).

For validating the algorithms implemented in the N-DPU application software, it will be needed to produce a huge amount of test data. The test data could be fullimages or stacks of windows. In order to have a robust architecture avoiding the realtime transfer of the data between SimuCam and the main test system, a local high volume storage facility (SSD) has to be managed by the SimuCam. The idea is to be able to pre-load in advance in the SSD the data (fullimages or windows) which will be sent by the N-FEE simulation entities during the simulation phases.

3.3.3 Other requirements

SimuCam shall also offer the following functionalities:

- N-FEE command management via RMAP
- N-FEE HK generation
- Window position list management
- Error injection (SpaceWire, RMAP, data content, etc.)
- Synchronization between N-FEE simulators
- Periodic status reporting
- Event reporting
- Log reporting

3.3.4 Requirement diagram

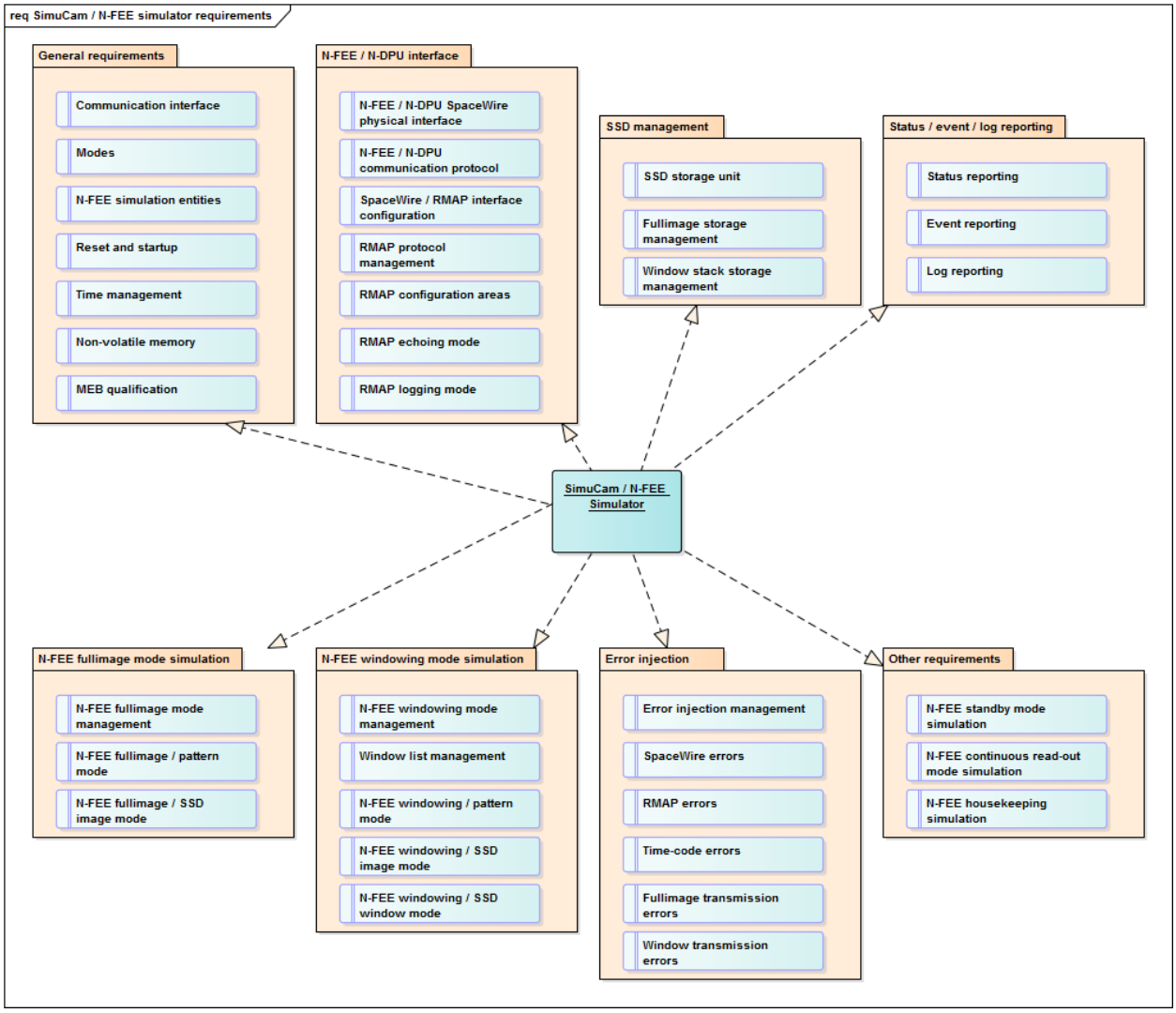

Figure 3 - SimuCam / N-FEE simulator requirements

3.4 Configuration examples

This section describes some test configurations in which the N-FEE simulator is used. These test configurations correspond to the use case #1 and use case #2 identified in the section 3.2.

3.4.1 Configuration #1 – Development bench (standalone N-DPU board)

The configuration #1, called 'Development bench (standalone N-DPU board)', has the following features:

- Target = any stand-alone N-DPU board (GR712RC evaluation board, BB, EM, EQM, …)
- Functional scope =
  - Booting and testing the N-DPU Application S/W with one N-DPU standalone board.
  - Simulation of 2 N-FEE (2 cameras)

- SimuCam use case: #1

The configuration is used by the N-DPU ASW development team. The N-DPU board is by default a GR712RC development board; however, depending on the test needs, a N-DPU breadboard or a N-DPU EM can be used instead of the GR712RC board. This configuration can be duplicated in order to have two development chains working in parallel: in this case, the SimuCam can be shared by both chains.

The figure below describes the configuration #1:

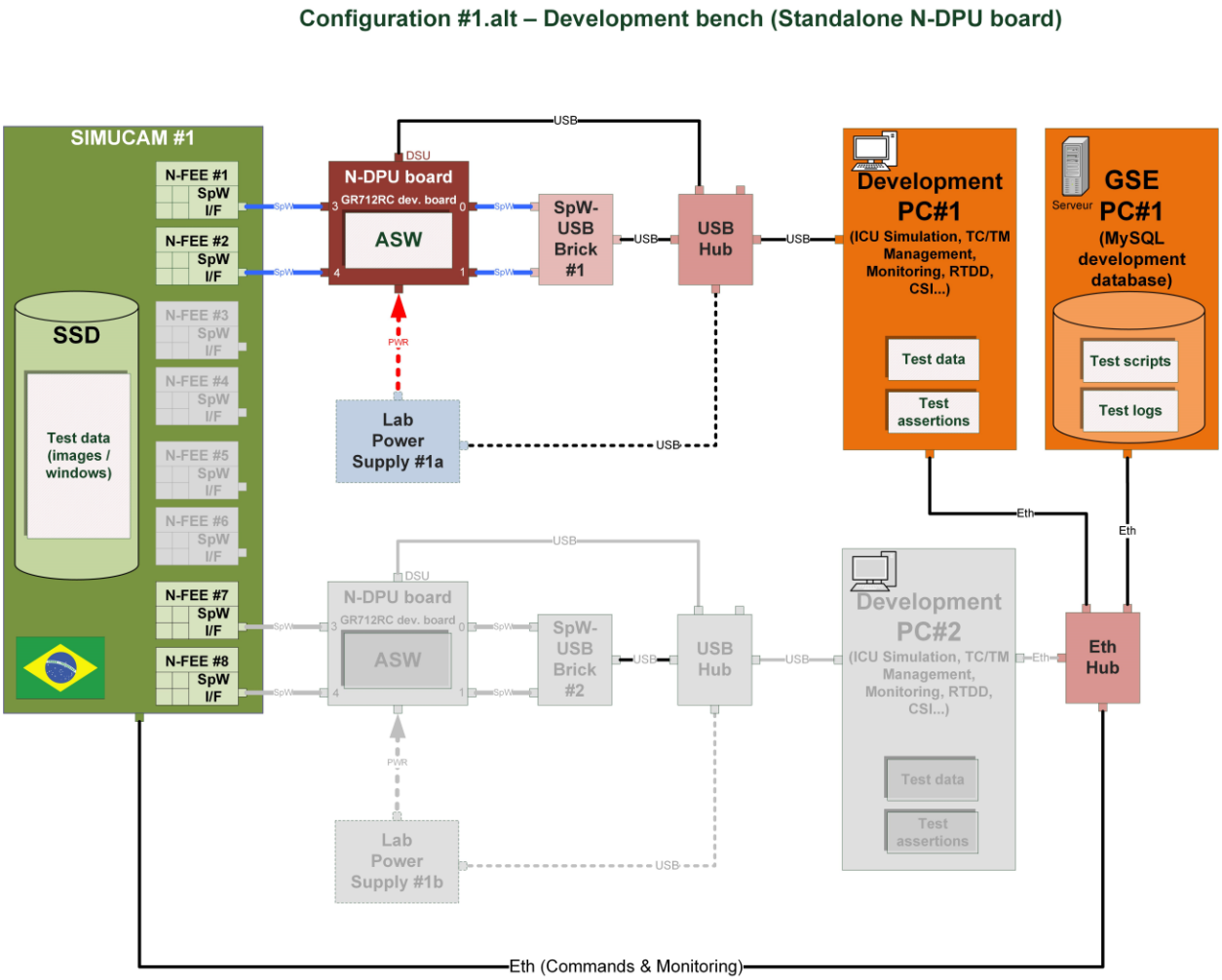

Figure 4 - Configuration #1 – Development bench (standalone N-DPU board)

3.4.2 Configuration #2 – Validation bench (MEU proxy configuration)

The configuration #2, called 'Validation bench (MEU proxy configuration)', has the following features:

- Target = any stand-alone N-DPU board (GR712RC evaluation board, BB, EM, EQM, …)
- Functional scope =
  - Booting and testing the N-DPU Application S/W with a MEU proxy.

- Simulation of 4 N-FEE (4 cameras)
- SimuCam use case: #1

This configuration #2 is used for:

- the SGSE test
- the N-DPU ASW validation campaign
- the preparation of the N-DPU ASW acceptance test inside the MEU
- the preparation of the MEU coupling tests and functional tests

The N-DPU boards are the EM board and breadboard. These both board can be replaced, if needed, by GR712RC development board.

The N-DPU ASW have a specific test mode allowing to simulate the TM packet generation of up to 6 N-DPU (by duplication of the TM packets). This mode is used for the full load test of SGSE.

The figure below describes the configuration #2:

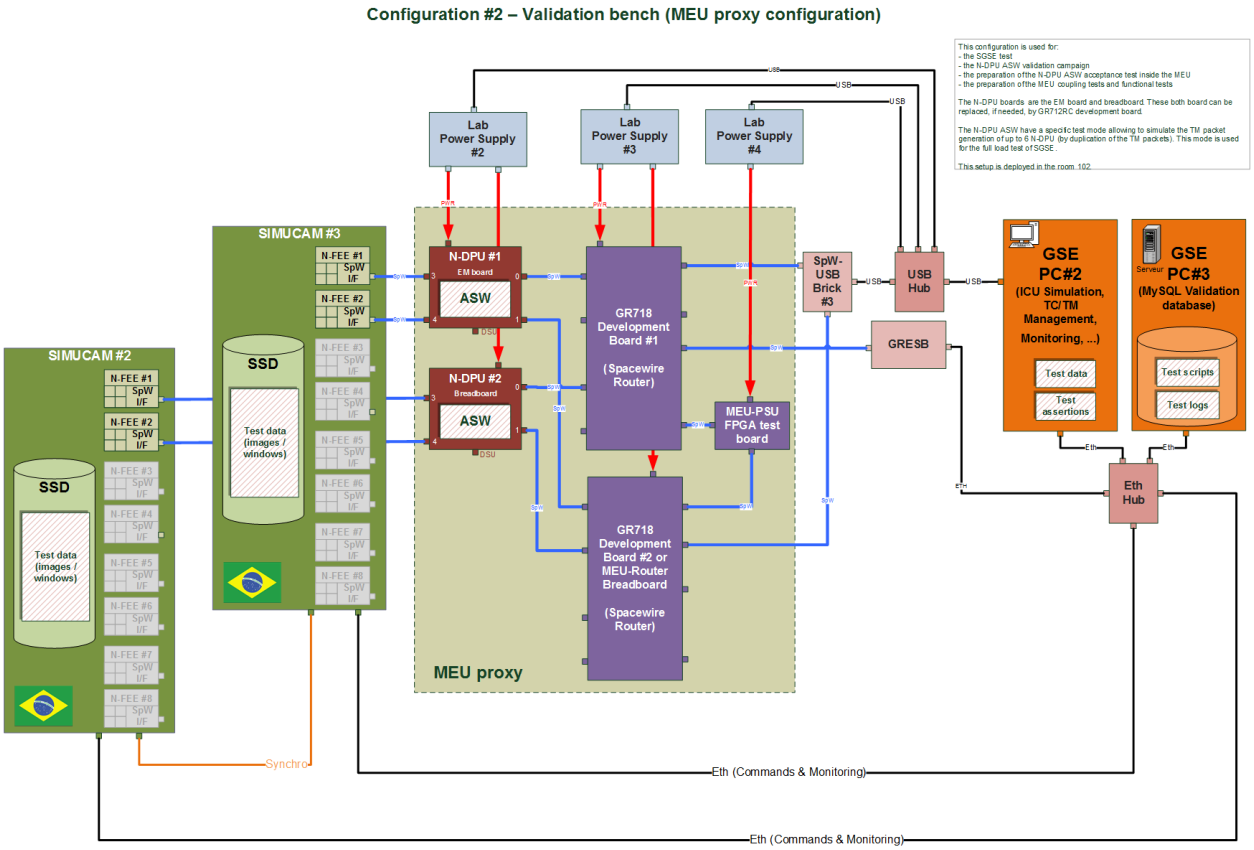

Figure 5 - Configuration #2 – Validation bench (MEU proxy configuration)

3.4.3 Configuration #3 – Acceptance bench (MEU configuration)

The configuration #3, called 'Acceptance bench (MEU configuration)' has the following features:

- Target = MEU QM, PFM, FM

- Functional scope =
  - Booting and testing the N-DPU Application S/W with a full MEU (x6 instances loaded by the GSE and running in the MEU)
  - Simulation of 12 N-FEE (12 cameras)
- SimuCam use case: #2

The figure below describes the configuration #3:

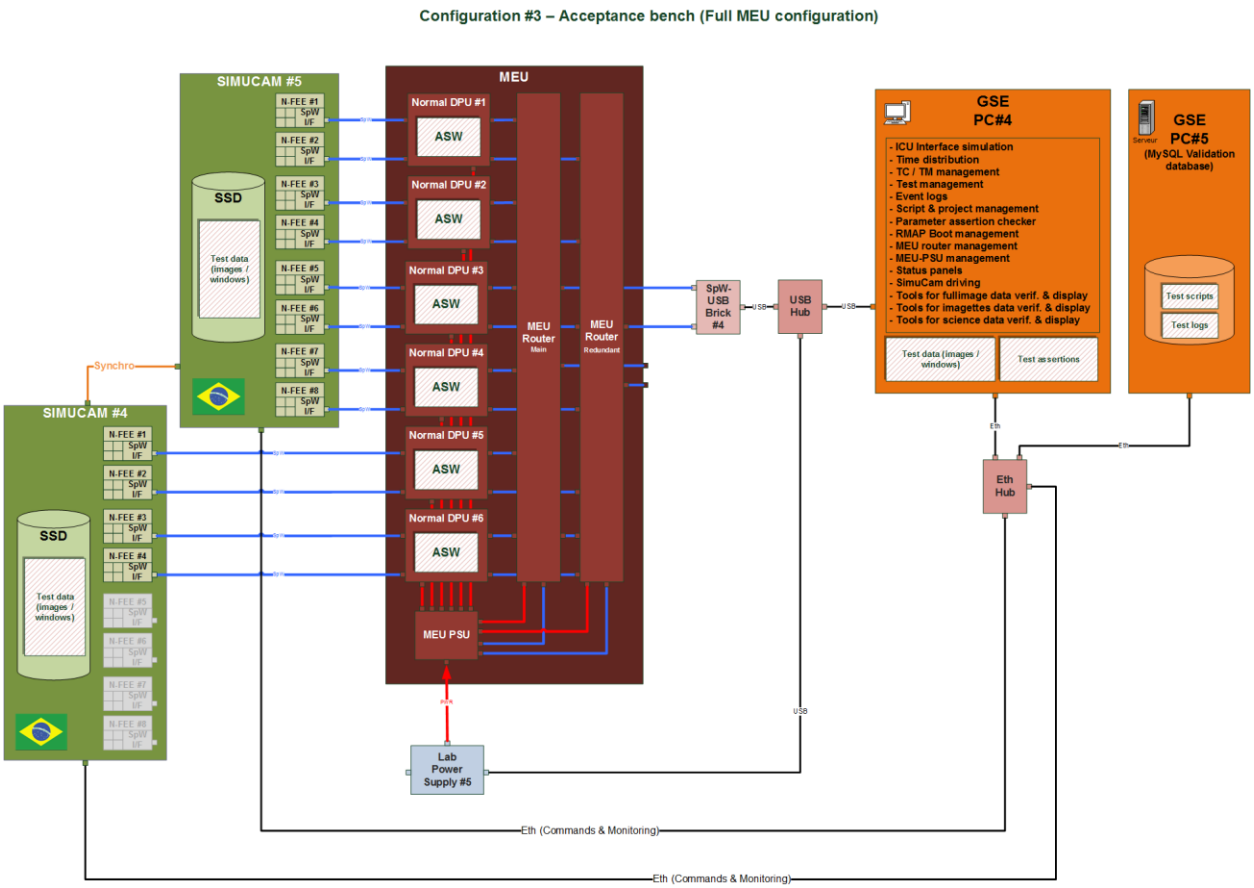

Figure 6 - Configuration #3 – Acceptance bench (MEU configuration)

4. Requirements

4.1 SimuCam general requirements

	Title:	**SimuCam general requirements**
	Type:	Constraint
	Verif.:	Inspection
	LastEdit:	2019-10-31
NFEESIM-UR-403	The SimuCam shall be made up two parts: · a main electronic box which is the core of the system and which manages all the real-time aspects of the N-FEE simulation, · a standalone test software client allowing to drive and monitor one or several main electronic boxes for test purposes.	

Hereafter, the main electronic box will be called 'SimuCam MEB' and the standalone test software 'SimuCam SSW'.

The term 'N-FEE simulation entity' is used to name an entity, managed by the SimuCam MEB, allowing to simulate one N-FEE independently of the others. The SimuCam MEB can be seen as a set of several independent N-FEE simulators (i.e. N-FEE simulation entities) gathered in the same box.

4.2 SimuCam SSW requirements

	Title:	**SimuCam SSW requirements**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-407	The SimuCam SSW shall allow to command all the functions of the SimuCam MEB.	

Note 1: Concerning the use case #1 and #2, in order to improve the automation process, LESIA intends to connect the SimuCam MEB directly to its central GSE system. That means that LESIA will develop its own SimuCam MEB software client in such a way it will be perfectly integrated to its test software system: the goal is to be able to drive several SimuCam MEB directly from the LESIA control management tools and to monitor them, in an integrated way, thanks to the LESIA monitoring tools. The SimuCam SSW will be used only for performing the SimuCam MEB acceptance tests and for debugging purpose in case of trouble. The SimuCam SSW won't be used in the N-DPU ASW validation process, nor for the MEU tests. For this reason, we don't give in this document detailed user requirements related to this sub-system.

Note 2: Concerning the use case #3, DLR for driving the N-FEE simulation can use either the SimuCam SSW or the LESIA SimuCam MEB client. In this second configuration, a command / control interface between the LESIA SimuCam MEB client and the DLR test system can be defined in order to improve the automation process: the specification of such an interface is not the purpose of this document.

4.3 SimuCam MEB general requirements

4.3.1 MEB communication interface

	Title:	**SimuCam MEB communication interface**
	Type:	Interface
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-558	The SimuCam MEB shall be commanded and monitored by means of an Ethernet interface.	

	Title:	**SimuCam MEB communication interface**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-06
NFEESIM-UR-1164	The ethernet interface protocol shall be compliant with EDEN protocole as define in [RD11]	

	Title:	**SimuCam MEB communication interface**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-559	All the SimuCam MEB shall be configured at startup with a different IP address provided by a static allocation mechanism.	

	Title:	**SimuCam MEB communication interface**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-690	The SimuCam MEB shall allow the user to configure the MEB IP address without the Ethernet communication.	

Note: A specific configuration file stored on the MEB non-volatile memory (SD card) could be an option for configuring this kind of parameters. The dynamic or automatic allocation mechanisms based on the use of a DHCP server shall not be used.

4.3.2 MEB Modes

	Title:	**SimuCam MEB modes**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31

NFEESIM-UR-531	The SimuCam MEB shall manage the following exclusive modes: · MEB_CONFIG · MEB_RUNNING

Note: The MEB_CONFIG mode is the mode in which the configuration operations common to all the N-FEE simulation entities have to be performed. In particular, the SSD image / window upload process is supported only in this mode. The MEB_RUNNING mode is the mode in which each N-FEE simulation entity can be started and used.

	Title:	**SimuCam MEB modes**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-537	The SimuCam MEB shall allow the user to change via an Ethernet command the active mode (MEB_CONFIG / MEB_RUNNING).	

	Title:	**SimuCam MEB modes**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-535	In MEB_CONFIG mode, the SimuCam MEB shall force all the N-FEE simulation entities in the NFEE_CONFIG state.	

Note: The N-FEE simulation entity states are described below.

	Title:	**SimuCam MEB modes**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-536	In MEB_CONFIG mode, the SimuCam MEB shall reject the commands setting a N-FEE simulation entity in one of the NFEE_RUNNING_* states.	

	Title:	**SimuCam MEB modes**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-31
NFEESIM-UR-543	In MEB_RUNNING mode, the SimuCam MEB shall accept the commands setting a N-FEE simulation entity in one of the NFEE_RUNNING_* states.	

4.3.3 N-FEE simulation entities

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Test

	LastEdit:	2019-11-21
NFEESIM-UR-405	The SimuCam MEB shall allow to manage at least 6 N-FEE simulation entities independently of each other.	

Note 1: In terms of SpaceWire configuration, the baseline at DPS level is to have one SpaceWire link per N-FEE, each N-DPU managing 2 N-FEE (i.e. 2 N-cameras). A MEU contains 6 N-DPU boards. With a SimuCam MEB simulating 6 N-FEE, the total number of SimuCam needed for testing a full MEU is limited to two.

Note 2: The requirement given above covers the need to command each N-FEE simulation entity independently of each other.

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Inspection
	LastEdit:	2019-10-31
NFEESIM-UR-526	One SpaceWire link shall be associated to each N-FEE simulation entity.	

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-672	Each N-FEE simulation entity shall be able to handle the data of four CCD: · CCD1 · CCD2 · CDD3 · CDD4	

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14

NFEESIM-UR-534	Each N-FEE simulation entity shall manage the following states: · NFEE_CONFIG: the N-FEE simulation entity is disabled (i.e. N-FEE is switched off) and can be configured · NFEE_RUNNING_ON: the N-FEE simulation entity is enabled (i.e. N-FEE is on) and simulates the N-FEE ON mode · NFEE_RUNNING_STANDBY: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE STANDBY mode · NFEE_RUNNING_FULLIMAGE: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE FULLIMAGE mode · NFEE_RUNNING_WINDOWING: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE WINDOWING mode · NFEE_RUNNING_FULLIMAGE_PATTERN: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE FULLIMAGE_PATTERN mode (same behaviour than NFEE_RUNNING_FULLIMAGE, but the pattern emission mode is forced) · NFEE_RUNNING_WINDOWING_PATTERN: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE WINDOWING_PATTERN mode (same behaviour than NFEE_RUNNING_WINDOWING, but the pattern emission mode is forced) · NFEE_RUNNING_PARALLEL_TRAP_PUMP_1: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE Parallel trap pumping mode 1 · NFEE_RUNNING_PARALLEL_TRAP_PUMP_2: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE Parallel trap pumping mode 1 · NFEE_RUNNING_SERIAL_TRAP_PUMP_1: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE Serial trap pumping mode 1 · NFEE_RUNNING_SERIAL_TRAP_PUMP_2: the N-FEE simulation entity is enabled (i.e. N-FEE is switched on) and simulates the N-FEE Serial trap pumping mode 2

The figure below is the N-FEE simulation entity statechart:

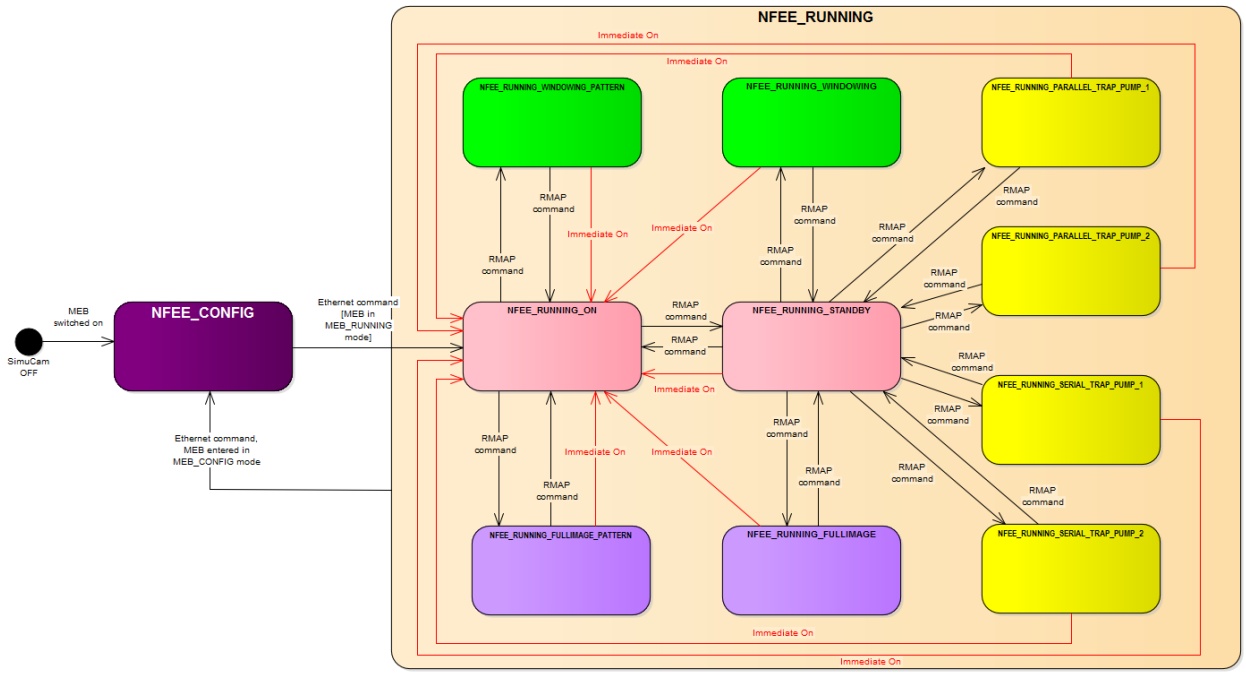

Figure 7 - FEE Simulation Entity Statechart

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-549	The SimuCam MEB shall allow the user to set individually, via an Ethernet command, each N-FEE simulation entity in one of the following states: · NFEE_CONFIG · NFEE_RUNNING_ON · NFEE_RUNNING_STANDBY · NFEE_RUNNING_FULLIMAGE_PATTERN · NFEE_RUNNING_FULLIMAGE · NFEE_RUNNING_WINDOWING_PATTERN · NFEE_RUNNING_WINDOWING · NFEE_RUNNING_PARALLEL_TRAP_PUMP_1 · NFEE_RUNNING_PARALLEL_TRAP_PUMP_2 · NFEE_RUNNING_SERIAL_TRAP_PUMP_1 · NFEE_RUNNING_SERIAL_TRAP_PUMP_20	

For example, among the 6 N-FEE simulation entities, 2 N-FEE simulation entities can be in NFEE_CONFIG state (i.e. N-FEE are off) and the 4 other can be active (i.e N-FEE are ON). Among the 4 active entities, one can simulate the N-FEE standby mode (NFEE_RUNNING_STANDBY state), one other can work in windowing mode (NFEE_RUNNING_WINDOWING state) while the 2 last work in fullimage mode (NFEE_RUNNING_FULLIMAGE state). Any other combination can be possible.

Note: normally, the transitions between the NFEE_RUNNING_STANDBY state and the NFEE_RUNNING_FULLIMAGE/WINDOWING states should be triggered via the SpaceWire/RMAP commands from the N-DPU. However, for test purpose, it could be useful to trigger these state transitions from the GSE software.

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-529	The SimuCam MEB shall allow the user to configure a N-FEE simulation entity (SpaceWire / RMAP interface parameters, ...) only when this entity is in the NFEE_CONFIG state.	

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-530	When a N-FEE simulation entity enters in the NFEE_CONFIG, the SimuCam MEB shall: · stop the N-FEE simulation on the associated SpaceWire link, · disable its associated SpaceWire link.	

	Title:	**N-FEE simulation entity**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-532	When a N-FEE simulation entity enters in one the NFEE_RUNNING states, the SimuCam MEB shall: · enable its associated SpaceWire link, · start the N-FEE simulation on the associated SpaceWire link.	

4.3.4 Reset and startup

	Title:	**SimuCam MEB reset and startup**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-567	The user shall be able to trigger a reset of the SimuCam MEB via a specific Ethernet command at any moment.	

	Title:	**SimuCam MEB reset and startup**

	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-562	At startup (after a reset of after powering on), the SimuCam MEB shall enter in MEB_CONFIG mode.	

	Title:	**SimuCam MEB reset and startup**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-568	At startup (after a reset of after powering on), the SimuCam MEB shall be configured using a default set of configuration parameters stored in its non-volatile memory.	

4.3.5 Non-volatile memory

	Title:	**Non-volatile memory**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-496	The SimuCam MEB shall manage a non-volatile memory in order to store the default set of configuration parameters which shall be used at startup (after reset or powering-on).	

4.3.6 Time management

	Title:	**Time management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-591	The SimuCam MEB shall provide a real-time clock.	

Note: This real-time clock will be used mainly for logging and time-stamping purpose.

	Title:	**Time management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-660	The SimuCam MEB shall maintain an internal time using its real-time clock.	

	Title:	**Time management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-695	The SimuCam MEB shall timestamp all the data packets sent via Ethernet.	

	Title:	**Time management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-696	The SimuCam MEB shall timestamp all the entries provided in its logs.	

	Title:	**Time management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-693	The SimuCam MEB shall be able to synchronize its internal time on an absolute time provided by the way of an Ethernet command.	

Note: Such a synchronization command could be sent at startup or periodically. The need is to be able to correlate the MEB logs and reports with the other events managed by the LESIA GSE system.

4.3.7 Endianness

	Title:	**Endianness**
	Type:	Constraint
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-658	All the data (HK, pixel data, commands) exchanged between the N-DPU and the N-FEE simulation entities via SpaceWire / RMAP packets shall be coded in big-endian order.	

	Title:	**Endianness**
	Type:	Constraint
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-708	All the data (commands, fullimages, windows, logs, dumps, etc.) exchanged via Ethernet between the SimuCam MEB and its remote client shall be coded in big-endian order.	

4.3.8 Coordinate system

	Title:	**Coordinate system**

	Type:	Constraint
	Verif.:	Test
	LastEdit:	2019-11-06
NFEESIM-UR-858	The SimuCam MEB shall use a coordinate system compliant to the one defined in [AD4].	

[AD4] defines the areas of the CCD, overscan and pre-scan and their dimensions and positions. The figure below depicts the various CDD zones and gives the coordinate system which shall be used:

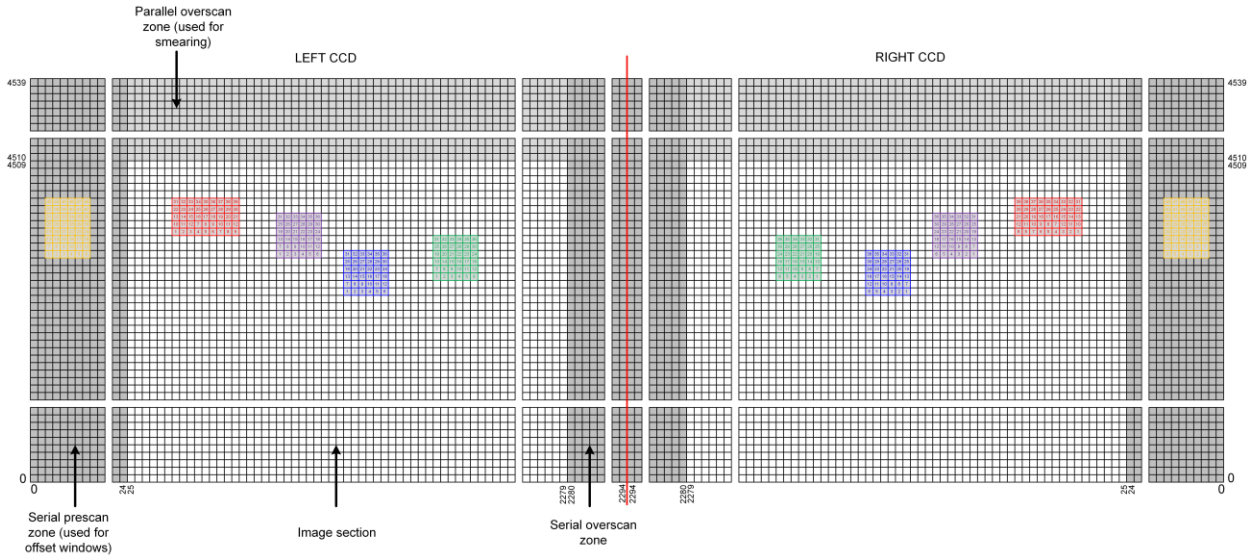

Figure 8 - CCD overview and coordinate system

The coordinate system for the windowing is derived from the CCD read-out scheme. Right and left side of the CCD have seperate coordinate-systems. The origin of both coordinate-systems is the first pixel read from the CCD.

4.3.9 MEB qualification

	Title:	**SimuCam MEB tests and qualification**
	Type:	Quality
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-822	The prevention of damage on the flight product due to SimuCam MEB failure shall be verified.	

	Title:	**SimuCam MEB tests and qualification**
	Type:	Quality
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-823	The compatibility of the interfaces of SimuCam MEB with flight products and facilities shall be verified by test.

4.4 N-FEE / N-DPU interface

4.4.1 N-FEE / N-DPU SpaceWire physical interface

	Title:	**N-FEE / N-DPU SpaceWire physical interface**
	Type:	Interface
	Verif.:	Inspection
	LastEdit:	2019-11-12
NFEESIM-UR-826	The SpaceWire interface shall be nine pin micro miniature D-type connectors in line with [ES3].	

	Title:	**N-FEE / N-DPU SpaceWire physical interface**
	Type:	Quality
	Verif.:	Inspection
	LastEdit:	2019-10-30
NFEESIM-UR-827	The SpaceWire electrical interface shall be compliant to the requirements given in [ES3].	

4.4.2 N-FEE / N-DPU communication protocol

	Title:	**N-FEE / N-DPU communication protocol**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-448	The SimuCam MEB shall implement the N-FEE / N-DPU communication protocol as defined in [AD2] and [AD5]: · the commands (read or write) sent by the N-DPU are sent as RMAP commands, · the pixels data and periodic HK data are sent by the N-FEE as SpaceWire packets according to a specific format specified in [AD2] and [AD5].	

4.4.3 SpaceWire / RMAP interface configuration

	Title:	**SpaceWire / RMAP interface configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-20

NFEESIM-UR-423	For each N-FEE simulation entity, the SimuCam MEB shall manage the following set of parameters configuring the SpaceWire / RMAP interface: · autoStart / linkStart mode · link speed (expressed in Mbps) · logical address (node address used by the N-DPU to build the SpaceWire packets sent to the N-FEE simulation entity) · destination node address (node address used by the N-FEE simulation entity to build the SpaceWire packets sent to the N-DPU) · time-code transmission (enabled / disabled) · RMAP key

	Title:	**SpaceWire / RMAP interface configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-475	For each N-FEE simulation entity, the SimuCam MEB shall manage a default SpaceWire / RMAP configuration parameter set.	

	Title:	**SpaceWire / RMAP interface configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-476	The SimuCam MEB shall store in its non-volatile memory the default SpaceWire / RMAP configuration parameter set of each N-FEE simulation entity.	

	Title:	**SpaceWire / RMAP interface configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-477	In the MEB_CONFIG mode, the SimuCam MEB shall allow the user to change the default SpaceWire / RMAP configuration parameter set of each N-FEE simulation entity.	

	Title:	**SpaceWire / RMAP interface configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-478	At startup, the SimuCam MEB shall configure each N-FEE simulation entity with the default SpaceWire / RMAP configuration parameter set stored in the MEB non-volatile memory.

	Title:	**SpaceWire / RMAP interface configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-474	When a N-FEE simulation entity is in the NFEE_CONFIG state, the SimuCam MEB shall allow the user to change its active SpaceWire / RMAP configuration parameter set.	

	Title:	**SpaceWire / RMAP interface configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-473	The SimuCam MEB shall allow the user to command the link enable/disable operation of each N-FEE simulation entity by a specific command at any moment when the N-FEE simulation entity is in one of the NFEE_RUNNING states.	

Note 1: The SpaceWire link is automatically enabled when entering one the NFEE_RUNNING state.

Note 2: The goal of the requirement above is to be able to cause a disconnection for testing the failure cases.

4.4.4 RMAP protocol management

	Title:	**RMAP protocol management**
	Type:	Interface
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-638	The SimuCam MEB shall implement the RMAP protocol as defined in [ES5].	

	Title:	**RMAP protocol management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13

NFEESIM-UR-640	The SimuCam MEB shall check incoming RMAP packets for: · Protocol ID · Instruction · Target address · Key · Header CRC

	Title:	**RMAP protocol management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-641	If the check shows no errors, the SimuCam MEB shall send a RMAP reply with a 'command executed successfully' status.	

	Title:	**RMAP protocol management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-06
NFEESIM-UR-642	Otherwise, the SimuCam MEB shall send an error code as specified in [ES5].	

4.4.5 RMAP configuration areas

	Title:	**RMAP configuration areas**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-583	The SimuCam MEB shall manage, for each N-FEE simulation entity, the following RMAP areas (see [AD2] and [AD5]): · verified configuration area · unverified configuration area · housekeeping area · window definition area	

	Title:	**RMAP configuration areas**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13

NFEESIM-UR-584	The SimuCam MEB shall make readable via RMAP read commands, for each N-FEE simulation entity, all the RMAP configuration areas.

	Title:	**RMAP configuration areas**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-585	The SimuCam MEB shall make writable via RMAP write commands, for each N-FEE simulation entity, all the RMAP configuration areas.	

Note: All the configuration areas of the real N-FEE will be readable and writable by the way of RMAP commands. It will be possible, for debugging purpose, to access to these areas using the memory management service implemented in the N-DPU. The memory management service implemented in the N-DPU will be extended to the N-FEE remote memory. To validate this service, the SimuCam shall make readable / writable via RMAP all the RMAP configuration areas.

	Title:	**RMAP configuration areas**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-699	The SimuCam MEB shall allow the user to dump via an Ethernet command the content of the RMAP configuration areas.	

Note: this dump functionality commanded via Ethernet will allow to perform some automatized verifications in the context of the N-DPU ASW validation according to the following scenario:

- The N-DPU writes the RMAP configuration areas.
- The test engineer gets, using the Ethernet command, the content of the RMAP configuration areas and checks that the actual content is compliant to the expected content.

4.4.6 RMAP echoing mode

	Title:	**RMAP echoing mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-588	The SimuCam MEB shall allow the user to enable / disable, for each N-FEE simulation entity, a RMAP echoing mode.	

	Title:	**RMAP echoing mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13

NFEESIM-UR-589	At startup, the SimuCam MEB shall disable the RMAP echoing mode for all the N-FEE simulation entities.

	Title:	**RMAP echoing mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-22
NFEESIM-UR-590	When a RMAP echoing mode is enabled, a N-FEE simulation entity shall echo, as a dedicated SpaceWire packet, each incoming packet (i.e. RMAP write / read commands) and each outgoing responses (i.e. RMAP write acknowledgements and RMAP read replies).	

Note: With this RMAP echoing mode, it will be possible to track the echo packets in the GSE monitoring tools and to perform some cross verifications in an automatic way.

	Title:	**RMAP echoing mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-06
NFEESIM-UR-1166	The echoed SpaceWire packet shall contain an entry that uniquely identify the original RMAP packet associated N-FEE simulation entity.	

This unique identification can simply be performed by replacing the echoed RMAP packet DLA entry by a different DLA value uniquely associated with each N-FEE simulation entity

	Title:	**RMAP echoing mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-06
NFEESIM-UR-1167	Each echoed SpaceWire packet shall be sent on the same SpaceWire link, a spare link not associated with any N-FEE simulation entity	

4.4.7 Deleted

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-702	Deleted	

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06

NFEESIM-UR-703	Deleted

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-704	Deleted	

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-706	Deleted	

Deleted

4.5 Timing and synchronization management

4.5.1 N-FEE timing configuration

	Title:	**N-FEE timing configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-447	The SimuCam MEB shall manage the following parameters defining the CCD readout sequence: · exposure period = EP (default value = 25 sec., possible values : 25 + 6.25xn with n in [0,4]) · CCD readout time = RT (default value = 4 sec.)	

In the rest of this document, the following elements will be used for timing signal identification

- FFS : First frame signal. Emitted each EP sec.
- OFS : Other frame signal. Emitted three times after FFS.
- FSS : Frame start signal. Either a FFS or a OFS.
- PFS : Periodical frame signal. The periodical signal combined from FFS and OFS

Between FFS and first OFS the duration is 6,25 sec.

Between each OFS the duration is 6,25sec.

Between last OFS of an EP and next FFS the duration is (n+1) x 6.25 sec.

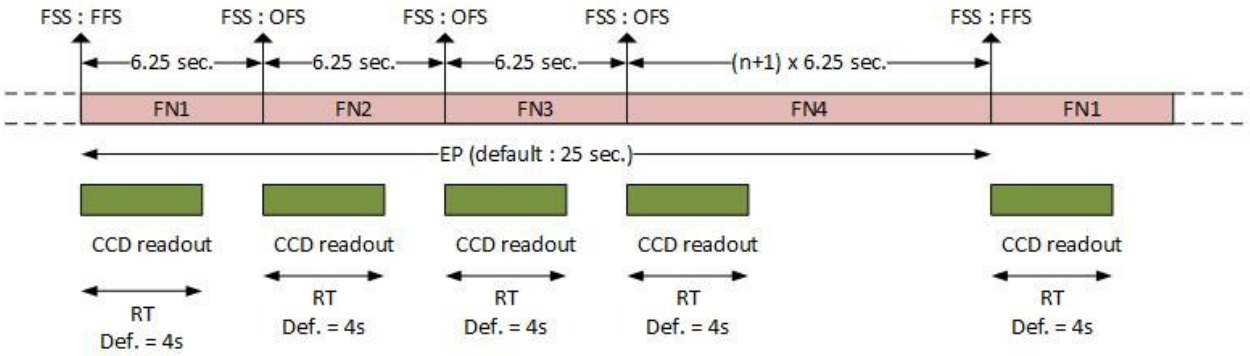

Figure 9 - N-FEE timing configuration

	Title:	**N-FEE timing configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-898	The SimuCam MEB shall manage a default set of CCD readout sequence parameters.	

	Title:	**N-FEE timing configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-899	The SimuCam MEB shall store in its non-volatile memory the default set of CCD readout sequence parameters.	

	Title:	**N-FEE timing configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-900	At startup, the SimuCam MEB shall use the default set of CCD readout sequence parameters stored in its non-volatile memory.	

	Title:	**N-FEE timing configuration**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-06

NFEESIM-UR-829	In the MEB_CONFIG mode, the SimuCam MEB shall allow the user to set, by the way of an Ethernet command, the parameters EP and RT.

Note 1: The exposure period EP corresponds to the PLATO N-Camera CCD exposure period. The four N-Camera CCD are not read out at the same time. Their readouts are shifted by 6.25 seconds.

Note 2: Since there is no buffering stage at the N-FEE level, the pixel transmission is performed at the same time than the readout: consequently, RT corresponds also to the data transmission duration.

Note 3: The EP and RT parameters are common to all the N-FEE simulation entities.

4.5.2 Synchronization signal

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-485	The SimuCam MEB shall be able to generate internal synchronization signals corresponding to PFS.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-06
NFEESIM-UR-636	The jitter of the internal synchronization signal shall be lower than 1µs.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-631	The SimuCam MEB shall be able to use as PFS synchronization signal an external synchronization signal provided by another SimuCam MEB system.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-632	The SimuCam MEB shall be able to use as PFS synchronization signal an external synchronization signal provided by a N-AEU.	

	Title:	**Synchronization signal**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-633	The SimuCam MEB shall allow the user to select, via an Ethernet command, which synchronization source (internal / external) has to be used.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-634	At startup, the SimuCam MEB shall configure the synchronization source with a default value stored in its non-volatile memory.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-635	The SimuCam MEB shall allow the user to change the default synchronization source stored in its non-volatile memory.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-06
NFEESIM-UR-630	The SimuCam MEB shall be able to propagate its internal PFS synchronization signal to other SimuCam MEB systems.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-835	The SimuCam MEB shall generate an external synchronization signal with timing similar to the Clk_N_ccdread signal timing specified in [AD1], PLT-N-FEE-221 and PLT-N-FEE-224.0	

Note: The CCD1 start readout pulse width is twice the CCD2 to 4 start readout pulse width.

	Title:	**Synchronization signal**
	Type:	Interface
	Verif.:	Inspection
	LastEdit:	2019-11-12

NFEESIM-UR-629	The SimuCam MEB shall provide two HW interfaces (input and output) for PFS synchronization signal retrieving and propagating.

	Title:	**Synchronization signal**
	Type:	Interface
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-910	The SimuCam MEB shall maintain, for each N-FEE simulation entity, a frame number register (FN) readable by RMAP. · The frame number is set to 0 when the FFS sync-impulse (CCD1 start readout pulse) is received. · The frame number is incrementented at each OFS sync-impulse (CCD2 to 4 start readout pulse).	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-1064	The SimuCam MEB shall simulate the parameter EP and use it for synchronization signal generation	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-1066	At startup the SimuCam MEB shall configure the EP value with a default value stored in its non-volatile memory.	

The default value stored should be 25 sec.

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-1067	The SimuCam MEB shall allow the user to change the default EP value stored in its non-volatile memory.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-1065	The SimuCam MEB shall allow the user to modify, via an ethernet command, the value of EP (with respect to EP constraints)

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1125	The SimuCam MEB shall offer to use one unused N-FEE simulation entity as synchronization provider.	

	Title:	**Synchronization signal**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-06
NFEESIM-UR-1126	When used as a synchronization provider, a N-FEE simulation entity shall only · transmit timecode · manage frame number register (FN) · response to RMAP read operation on frame number register	

This feature will be necessary to synchronize N-DPU GSE with the internal synchronization signal generated by the SimuCam MEB.

4.5.3 Time-code management

	Title:	**Time-code management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-486	At the release of the synchronization signal (internal or external), the SimuCam MEB shall send a SpaceWire time-code on all the SpaceWire links for which the time-code transmission has been enabled.	

Note: The N-FEE notifies the N-DPU that a new CDD readout has started by transmitting a time-code.

	Title:	**Time-code management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-488	The SpaceWire time-code sent by the SimuCam MEB shall be the same for all the links.	

	Title:	**Time-code management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-489	The SpaceWire time-code shall be incremented after sending.	

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-838	Deleted	

4.5.4 Synchronization reset

	Title:	**Synchronization reset**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-709	The SimuCam MEB shall allow the user to reset by an Ethernet command the internal synchronization signal.	

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-728	Deleted	

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-729	Deleted	

	Title:	**Synchronization reset**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05

NFEESIM-UR-710	Upon reception of a synchronization reset command, the SimuCam MEB shall: · stop the data transmission on all the N-FEE simulation entities · put all the N-FEE simulation entities in the NFEE_RUNNING_STANDBY state excepted if the N-FEE simulation entity is in the NFEE_CONFIG mode · wait SYNCH_DELAY milliseconds where SYNCH_DELAY is a parameter of the synchronization reset command (SYNCH_DELAY is in [0,60000]) · release a synchronization signal · reset the time-code counter (set to 0) · start a new synchronization and acquisition cycle

Note: The reset synchronization functionality is needed to be able to control the start of the synchronization and acquisition cycle during the test activities. Without this function, depending when the tests are started, a non-deterministic delay of up to EP seconds can occur before receiving the first image packet: such a situation can make difficult the test timing management.

The test script below shows how this function could be used:

- SIMUCAM_MEB.enterMode(MEB_RUNNING)
- wait 1 s
- SIMUCAM_MEB.NFEESIM01.enterMode(NFEE_RUNNING_STANDBY)
- ICU_SIM.boot(NDPU01_software)
- wait 10s
- SIMUCAM_MEB.resetSynchro(2000)
- TC_NCAM01_ACQUIRE_IMAGE //--- this command puts the N-DPU01 in the CALIBRATION mode and the N-FEE01 in the CALIBRATION mode

With the scenario above, the NDPU01 will receive the time-code and the first data pixels from the NFEE01 two seconds after having entered the CALIBRATION mode.

The scenario below uses the auto-reset mode:

- SIMUCAM_MEB.enterMode(MEB_RUNNING)
- wait 1 s
- SIMUCAM_MEB.NFEESIM01.enterMode(NFEE_RUNNING_STANDBY)
- SIMUCAM_MEB.enableAutoSynchroReset(NFEE01)
- ICU_SIM.boot(NDPU01_software)
- wait 10s
- TC_NCAM01_ACQUIRE_IMAGE //--- this command puts the N-DPU01 in the CALIBRATION mode and the N-FEE01 in the CALIBRATION mode

With the scenario above, the NDPU01 will receive the time-code and the first data pixels from the NFEE01 immediately after having entered the CALIBRATION mode.

4.6 SSD management

4.6.1 SSD storage unit

For each simulated exposure (i.e. every 25 seconds nominally), the N-FEE simulation entities managed by the SimuCam MEB have to be fed with a big amount of data (fullimages or windows): in order to have

a robust architecture avoiding the realtime transfer of the data between the LESIA main GSE system and the SimuCam MEB, a local high volume storage facility (SSD) has to be managed by the MEB. The idea is to be able to pre-load in advance in the SSD the data (fullimages or windows) which will be sent by the N-FEE simulation entities during the simulation phases.

	Title:	**SSD storage unit**
	Type:	Non-Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-412	The SimuCam MEB shall manage locally a SSD storage unit of at least 1 Tb for storing the content of the fullimages and windows to be sent.	

The size of one fullimage (4510 x 4510 16-bit pixels) is about 330 Mb.

1 Tb allows to store up to 3030 fullimages.

The size of the windows produced by the LESIA PLATO Image Simulator (PIS) is about 10 x 10 16-bit pixels = 160 bits

A simulation involving 20000 stars and a duration of 6 hours will require 20000 x 6 x 3600 x 160 / 25 = 2.76 Gb

The size of 1 Tb seems to properly fulfils the needs of simulation data storage.

	Title:	**SSD storage unit**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-430	The SimuCam MEB shall allow the user to erase the full SSD content via an Ethernet command.	

	Title:	**SSD storage unit**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-597	The SimuCam MEB shall handle a SSD storage status containing the following information: · number of stored fullimages · volume occupied by the fullimages / total volume · number of stored windows · volume occupied by the windows / total volume · volume occupied / total volume	

Note: The SSD storage status is part of the MEB global status which is periodically reported and which can be reported on demand (see NFEESIM-UR-546 and NFEESIM-UR-465).

4.6.2 Fullimage storage management

	Title:	**SSD fullimage storage management**

	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-431	The SimuCam MEB shall identify each fullimage written in the SSD by an ID. · A fullimage corresponds to a full CCD image including both sides (right and left), the prescan columns and the overscan lines.	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-714	The ID allowing to identify each fullimage shall be a 128-bit UUID.	

Note 1: Identifying the images stored in the MEB via an UUID is cleaner for quality process management. Several independent users can store fullimages in the MEB SSD without having to check if a given ID is already used or not by another user. The ID allocation process is really simplified with this approach.

Note 2: If managing UUID is too much complex, they could be replaced by 32-bit integer ID.

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-439	The SimuCam MEB shall allow the user to add a new fullimage in the SSD via Ethernet commands.	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-676	The SimuCam MEB shall allow to handle fullimages whose the size is up to 48 MBytes.	

The current baseline for the CCD size is:

- CCD half-line size = 2255 pixels
- CCD serial prescan pixels = 25 pixels
- CCD serial overscan pixels = 15 pixels
- CCD image section lines (without overscan lines) = 4510
- CCD overscan lines (smearing lines) = 30

Knowing that a pixel is coded on 16 bits, this leads to a size of (2255+25+15)*2*(4510+30)*2/1024/1024 = 39.8 MBytes. Applying a 20% margin, we get the requirement of 48 MBytes.

	Title:	**SSD fullimage storage management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-432	The SimuCam MEB shall allow the user to delete a fullimage identified by its ID in the SSD via an Ethernet command.	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-454	The SimuCam MEB shall notify the user that there is no more room to store a new fullimage in the SSD.	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-455	The SimuCam MEB shall reject the storing of a fullimage if an image having the same ID is already stored in the SSD.	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-456	The SimuCam MEB shall notify the user that an image having the same ID is already stored in the SSD.	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-548	The SimuCam MEB shall accept the fullimage write / delete commands only in the MEB_CONFIG mode.	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05

NFEESIM-UR-435	Each fullimage transferred by a MEB client shall be associated with a set of properties defining the following features: · ID · Creation Date · Name: a string up to 256 characters · Category: a string up to 256 characters · Tag: a string up to 256 characters · Comment : a string up to 256 characters · Width in pixels (number of columns) · Height in pixels (number of lines)

The information available in the fullimage property set will allow a SimuCam MEB client to have a SSD fullimage browser with filtering capabilities. Filtering capabilities are not mandatory features for SimuCam. SimuCam client will retrieve the whole set of properties for all available images and perform filtering at client level (e.g image produced in a given date range, image whose name matches with a given regular expression, ...).

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-433	The SimuCam MEB shall be able to report on demand the full list of the fullimages available in the SSD (ID, Creation Date, Name, Category, Tag, Comment, Width and Height).	

	Title:	**SSD fullimage storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-600	The SimuCam MEB shall reject the storing of a fullimage if: · the fullimage size is inconsistent with the information given in the header · the width is odd	

4.6.3 Window stack storage management

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-601	The SimuCam MEB shall be able to handle window stacks as defined below: · A window stack is a pile of several windows corresponding to the same portion of a CCD taken a different time. · The maximum number of windows in a stack is: 65536 · The maximum window size is: 400 pixels · The maximum stack size is: 400*65536*2/1024/1024 = 50 MBytes

Note: All the figures given in the requirement above could be adjusted depending on the MEB hardware limitations. With a stack depth of 65536, it will be possible to have a simulation with a duration of 18 days.

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-678	The SimuCam MEB shall identify each window stack written in the SSD by an ID.	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-717	The ID allowing to identify each fullimage shall be a 128-bit UUID.	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-602	The SimuCam MEB shall allow the user to add a new window stack in the SSD via Ethernet commands.	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-603	The SimuCam MEB shall allow the user to delete a window stack identified by its ID in the SSD via an Ethernet command.	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-604	The SimuCam MEB shall notify the user that there is no more room to store a new window stack in the SSD.

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-605	The SimuCam MEB shall reject the storing of a window stack if a window stack having the same ID is already stored in the SSD.	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-606	The SimuCam MEB shall notify the user that a window stack having the same ID is already stored in the SSD.	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-607	The SimuCam MEB shall accept the window stack write / delete commands only in the MEB_CONFIG mode.	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-608	Each window stack transferred by a MEB client shall contain a header defining the following features: · Creation Date · Name: a string up to 256 characters · Category: a string up to 256 characters · Tag: a string up to 256 characters · Comment : a string up to 256 characters · Width in pixels · Height in pixels · Number of entries in the stack	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-609	The SimuCam MEB shall be able to report on demand the full list of the window stacks available in the SSD (ID, Creation Date, Name, Category, Tag, Comment, Width, Height, number of exposures).	

	Title:	**SSD window stack storage management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-610	The SimuCam MEB shall reject the storing of a window stack if: · The window size or the stack size are inconsistent with the information given in the header.	

4.7 N-FEE Digitalization simulation

4.7.1 Digitalization area definition

Whatever the source data used for simulation (SSD, pattern) and the output type (full image, windows ...) the N-FEE simulation shall simulate part of the digitalization process to define the actual pixel to be produced.

	Title:	**Digitalization area definition**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1005	The N-FEE simulation shall manage the following parameter : · v-Start : First row to digitized. Default value 0 · v-End : Last row to digitized. Default 4539. Max value : 4559	

Note: Given a V-Start at 0, the value of 4539 corresponds to 4510 visible rows plus 30 over-scan rows whereas the value of 4559 corresponds to 4510 visible rows plus 50 over-scan rows.

	Title:	**Digitalization area definition**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-1006	Any pixel whose coordinate is not into the range defined by v-Start, v-End shall not be simulated.	

Given a pixel (x,y) the pixel is simulated only if v-start <=y<=v-End.

Any windows defined outside this range will not be extracted nor transmitted.

Any windows partially defined outside this range will be partially extracted and transmitted.

	Title:	**Digitalization area definition**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-14
NFEESIM-UR-1008	The value set to v-Start and v-End shall be retrieved from associated RMAP register defined in [AD5]	

We will use in the rest of the document the following values:

- FLN : First line number, FLN = v-Start
- LLN : Last line number, LLN = v-End
- OLN : overscan lines number, OLN = MAX(0 , LLN-4509)
- ILN : image lines number, ILN = LLN - FLN +1 - OLN

4.7.2 Read-out order

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-839	Deleted	

	Title:	**Read-out order**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-1050	The N-FEE simulation shall simulate CCD readout order configuration as defined in [AD5].	

	Title:	**Read-out order**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-15
NFEESIM-UR-1051	The actual CCD simulated for each frame shall be retrieved from dedicated RMAP register.	

	Title:	**Read-out order**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-1052	The N-FEE simulation shall authorize the same CCD to be readout more than once.

Note: The system allows to configure the same CCD for different frame.

	Title:	**Read-out order**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-842	The N-FEE simulation entity shall use the readout order configuration to put in the Spacewire data packet header sent to the N-DPU the CCD ID corresponding to the current acquisition cycle.	

4.7.3 Warm-up simulation

	Title:	**Warm-up simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-1055	The N-FEE simulation shall manage the Digitised parameter.	

	Title:	**Warm-up simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-1056	The value set to Digitised parameter shall be retrieved from associated RMAP register as defined in [AD5].	

	Title:	**Warm-up simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-1057	The N-FEE simulation shall produce simulated data (data packets containing pixels) only if Digitised parameter is set to 1.	

Note: This corresponds to the simulation of the warm-up feature.

4.8 N-FEE to N-DPU data packet transmission

	Title:	**N-FEE to N-DPU data packet transmission**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-06

NFEESIM-UR-1083	The SimuCam MEB shall manage the parameter DATAPACKLEN defining the length of the content of a packet used for data transmission.

	Title:	**N-FEE to N-DPU data packet transmission**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-06
NFEESIM-UR-1085	The value of DATAPACKLEN shall be retrieved from packet_size sub-register as defined in [AD6].	

	Title:	**N-FEE to N-DPU data packet transmission**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-1089	The default value for DATAPACKLEN shall be 32140 bytes.	

4.9 N-FEE on mode simulation

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-928	In the NFEE_RUNNING_ON state, the N-FEE simulation entities shall simulate the behaviour of a real N-FEE running in ON mode as defined in [AD1], [AD2] and [AD5].	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-929	When entering the NFEE_RUNNING_ON state, the N-FEE simulation entities shall stop the transmission of the data (images or window segments) to the N-DPU.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-930	In the NFEE_RUNNING_ON state, the N-FEE simulation entities shall generate the time-code.

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-20
NFEESIM-UR-931	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-20
NFEESIM-UR-938	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_WINDOWING state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-15
NFEESIM-UR-937	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-936	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_FULLIMAGE state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**

	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-20
NFEESIM-UR-935	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_PARALLEL_TRAP_PUMP_1 state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-20
NFEESIM-UR-934	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_PARALLEL_TRAP_PUMP_2 state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-20
NFEESIM-UR-933	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_SERIAL_TRAP_PUMP_1 state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-20
NFEESIM-UR-932	Upon reception of the dedicated change mode or immediate on RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_SERIAL_TRAP_PUMP_2 state, the N-FEE simulation entity shall enter the NFEE_RUNNING_ON state.	

	Title:	**N-FEE on mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-939	The effective transition to the NFEE_RUNNING_ON state shall be performed depending on the synchronization signal and by fulfilling the timing specified in [AD1], [AD2] and [AD5].

Concerning the mode transition, the N-FEE simulation entity shall mimic the real N-FEE: this is crucial for qualifying the N-DPU ASW.

In the rest of the document, the assumption is made that the effective transition is performed :

- immmediately if the transition is due to an Immediate_on command
- on the release of the next FFS synchronization signal for classical change mode command

4.10 N-FEE standby mode simulation

	Title:	**N-FEE standby mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-06
NFEESIM-UR-451	In the NFEE_RUNNING_STANDBY state, the N-FEE simulation entities shall simulate the behaviour of a real N-FEE running in STANDBY mode as defined in [AD1], [AD2] and [AD5].	

	Title:	**N-FEE standby mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-681	When entering the NFEE_RUNNING_STANDBY state, the N-FEE simulation entities shall stop the transmission of the data (images or window segments) to the N-DPU.	

	Title:	**N-FEE standby mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-830	In the NFEE_RUNNING_STANDBY state, the N-FEE simulation entities shall generate the time-code.	

	Title:	**N-FEE standby mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-550	Upon reception of the dedicated change mode RMAP command from the N-DPU and if the N-FEE simulation entity is in the NFEE_RUNNING_FULLIMAGE state, the N-FEE simulation entity shall enter the NFEE_RUNNING_STANDBY state.

	Title:	**N-FEE standby mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-679	Upon reception of the dedicated change mode command from the N-DPU and if the N-FEE simulation entity is in the NFEE_RUNNING_WINDOWING state, the N-FEE simulation entity shall enter the NFEE_RUNNING_STANDBY state.	

	Title:	**N-FEE standby mode simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-925	The effective transition to the NFEE_RUNNING_STANDBY state shall be performed depending on the synchronization signal and by fulfilling the timing specified in [AD1], [AD2] and [AD5].	

Concerning the mode transition, the N-FEE simulation entity shall mimic the real N-FEE: this is crucial for qualifying the N-DPU ASW.

In the rest of the document, the assumption is made that the effective transition is performed on the release of the next FFS synchronization signal.

4.11 N-FEE fullimage mode simulation

4.11.1 N-FEE fullimage mode management

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-551	In the NFEE_RUNNING_FULLIMAGE state, the N-FEE simulation entities shall simulate the behaviour of a real N-FEE running in FULLIMAGE mode as defined in [AD1], [AD2] and [AD5].	

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-945	In the NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entities shall simulate the behaviour of a real N-FEE running in FULLIMAGE PATTERN mode as defined in [AD1], [AD2] and [AD5].

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-440	Upon reception of the dedicated change mode RMAP command from the N-DPU and if the N-FEE simulation entity is in the NFEE_RUNNING_STANDBY state, the N-FEE simulation entity shall enter the NFEE_RUNNING_FULLIMAGE state.	

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-946	Upon reception of the dedicated change mode RMAP command from the N-DPU and if the N-FEE simulation entity is in the NFEE_RUNNING_ON state, the N-FEE simulation entity shall enter the NFEE_RUNNING_FULLIMAGE_PATTERN state.	

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-655	The effective transition to the NFEE_RUNNING_FULLIMAGE and NFEE_RUNNING_FULLIMAGE_PATTERN state shall be performed depending on the synchronization signal and by fulfilling the timing specified in [AD1], [AD2] and [AD5].	

Concerning the mode transition, the N-FEE simulation entity shall mimic the real N-FEE: this is crucial for qualifying the N-DPU ASW.

In the rest of the document, the assumption is made that the effective transition is performed on the release of the next FFS synchronization signal.

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-444	In the NFEE_RUNNING_FULLIMAGE and NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entity shall select the CCD side (right or left) to transmit according to the content of the ad-hoc RMAP command received from the N-DPU.

Note: with one link between the N-FEE and the N-DPU, only one half-CDD image can be transferred every 6,25 seconds.

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-449	In the NFEE_RUNNING_FULLIMAGE and NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entity shall transmit to the N-DPU, on every FSS, a half-CCD image by fulfilling timing similar to those implemented in the real N-FEE and according to the following CCD cycle (where t0 is the start of the cycle and FN the frame number at t0; t0 corresponds to the occurrence of a FFS synchronization signal): · t0: Transmission of CCD associated to FN1 · t0+6.25: Transmission of CCD associated to FN2 · t0+12.5: Transmission of CCD associated to FN3 · t0+18.75: Transmission of CCD associated to FN4 · t0+EP: Transmission of CCD associated to FN1 · t0+EP+6.25: Transmission of CCD associated to FN2 · t0+EP+12.5: Transmission of CCD associated to FN3 · t0+EP+18.75: Transmission of CCD associated to FN4 · …	

Note: the first CCD sent is not necessarily the CCD1: it is the CCD associated to FN1.

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-902	In the NFEE_RUNNING_FULLIMAGE and NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entity shall transmit for each half-CCD image the following packets: · One HK packet · a set of data packets containing pixels corresponding to ILN lines of the half-CCD image · The Y-position in the CCD image of the first line to be transmitted is FLN · The Y-position in the CCD image of the last line to be transmitted is LLN-OLN-1 · a set of data packets containing OLN overscan lines (if OLN >0) · The Y-position in the CCD image of the first overscan line to be transmitted is given by the parameter 4510 · The Y-position in the CCD image of the last overscan line to be transmitted is given by the parameter 4510 + OLN -1

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-903	Deleted	

Note: Nominally, in the NFEE_RUNNING_FULLIMAGE and NFEE_RUNNING_FULLIMAGE_PATTERN state, a N-FEE simulation entity transmits a half-CDD image in 4 seconds: this corresponds to a data rate of about ((2255+25)*(4510+30+1)*16/1000/1000)/4 = 41 Mbps over 4 seconds (51.3 Mbps taking into account the SpaceWire overhead).

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-650	In the NFEE_RUNNING_FULLIMAGE and NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entity shall start the transmission of a new half-CCD image to the N-DPU at each release of the FSS synchronization signal.	

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-30

NFEESIM-UR-674	In the NFEE_RUNNING_FULLIMAGE and NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entity shall have completed the transmission of the entire half-CCD image RT seconds after the release of the FSS synchronization signal. · RT corresponds to the CCD readout time.S

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-511	In the NFEE_RUNNING_FULLIMAGE state, the N-FEE simulation entity shall be able to manage the two following modes of data generation: · N-FEE fullimage / pattern mode · N-FEE fullimage / SSD image mode	

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-512	At startup, by default, the SimuCam MEB shall use the N-FEE fullimage / pattern mode.	

	Title:	**N-FEE fullimage mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-15
NFEESIM-UR-843	In the NFEE_RUNNING_FULLIMAGE_PATTERN state, the N-FEE simulation entity shall manage only the pattern mode for data generation.	

4.11.2 N-FEE fullimage / pattern mode

	Title:	**N-FEE fullimage / pattern mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-10-30
NFEESIM-UR-441	The SimuCam MEB shall allow the user to configure, via an Ethernet command, a N-FEE simulation entity to work in the N-FEE fullimage / pattern mode.	

	Title:	**N-FEE fullimage / pattern mode**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-443	When the N-FEE fullimage / pattern mode is activated, the N-FEE simulation entity shall build, on the fly, half-CCD images which are filled with patterns according to [AD3].	

4.11.3 N-FEE fullimage / SSD image mode

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-501	The SimuCam MEB shall allow the user to configure, via an Ethernet command, a N-FEE simulation entity to work in the N-FEE fullimage / SSD image mode.	

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-553	When the N-FEE fullimage / SSD image mode is activated, the N-FEE simulation entity shall use fullimages from the SSD as data sent to the N-DPU.	

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-457	For each N-FEE simulation entity, the SimuCam MEB shall allow to manage the following parameters driving the N-FEE fullimage / SSD image mode: · Number of exposures: EN where 1 < EN < 65536 · List of the CCD1 fullimage ID: $CCD1\_ID_1$, $CCD1\_ID_2$, ..., $CCD1\_ID_{EN}$ · List of the CCD2 fullimage ID: $CCD2\_ID_1$, $CCD2\_ID_2$, ..., $CCD2\_ID_{EN}$ · List of the CCD3 fullimage ID: $CCD3\_ID_1$, $CCD3\_ID_2$, ..., $CCD3\_ID_{EN}$ · List of the CCD4 fullimage ID: $CCD4\_ID_1$, $CCD4\_ID_2$, ..., $CCD4\_ID_{EN}$	

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07

NFEESIM-UR-461	For each N-FEE simulation entity, the SimuCam MEB shall manage and store in its non-volatile memory a default set of parameters driving the N-FEE fullimage / SSD image mode (CCD ID list).

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-462	For each N-FEE simulation entity in NFEE_CONFIG state, the SimuCam MEB shall allow the user to change the default set of parameters driving the N-FEE fullimage / SSD image mode (CCD ID list).	

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-463	For each N-FEE simulation entity in NFEE_CONFIG state, the SimuCam MEB shall allow the user to change the active set of parameters driving the N-FEE fullimage / SSD image mode (CCD ID list).	

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-464	For each N-FEE simulation entity, at startup, the SimuCam MEB shall use the default set of parameters driving the N-FEE fullimage / SSD image mode (CCD ID list) as active set of parameters.	

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-458	When the N-FEE fullimage / SSD image mode is activated in a N-FEE simulation entity, the SimuCam MEB shall transmit to the N-DPU the CCD fullimage right side or left side (depending on the configuration received in the SpaceWire command from the N-FEE) according to the active fullimage ID list and according to the following cycle (t0 is the start of the cycle and the assumption is made here that FN = 0 at t0): · t0: CCD1_ID_1 · t0+0.25EP: CCD2_ID_1 · t0+0.5EP: CCD3_ID_1 · t0+0.75EP: CCD4_ID_1 · t0+EP: CCD1_ID_2 · t0+1.25EP: CCD2_ID_2 · t0+1.5EP: CCD3_ID_2 · t0+1.75EP: CCD4_ID_2 · … · t0+(EN-1)EP: CCD1_ID_{EN} · t0+(EN-0.75)EP: CCD2_ID_{EN} · t0+(EN-0.5)EP: CCD3_ID_{EN} · t0+(EN-0.25)EP: CCD4_ID_{EN} · t0+(EN)EP: CCD1_ID_1 · t0+(EN+0.25)EP: CCD2_ID_1 · t0+(EN+0.5)EP: CCD3_ID_1 · t0+(EN+0.75)EP: CCD4_ID_1 · ...

Note: the sequence given above is true if CCD are associated to frame in the following way : Frame1 = CCD1, Frame2 = CCD2, Frame3 = CCD3 and Frame4 = CCD4.

If for example the configuration was Frame1 = CCD3, Frame2 = CCD4, Frame3 = CCD1 and Frame4 = CCD2, then the sequence would be:

- t0: CCD3_ID_1
- t0+0.25EP: CCD4_ID_1
- t0+0.5EP: CCD1_ID_2
- t0+0.75EP: CCD2_ID_2
- t0+EP: CCD3_ID_2
- t0+1.25EP: CCD4_ID_2
- t0+1.5EP: CCD1_ID_3
- t0+1.75EP: CCD2_ID_3
- …
- t0+(EN-1)EP: CCD3_ID_{EN}
- t0+(EN-0.75)EP: CCD4_ID_{EN}
- t0+(EN-0.5)EP: CCD1_ID_1
- t0+(EN-0.25)EP: CCD2_ID_1

- t0+(EN)EP: CCD3_ID$_1$
- t0+(EN+0.25)EP: CCD4_ID$_1$
- t0+(EN+0.5)EP: CCD1_ID$_2$
- t0+(EN+0.75)EP: CCD2_ID$_2$

...

	Title:	**N-FEE fullimage / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-460	When a N-FEE simulation entity has reached the end of the ID list, it shall start again at the beginning of the ID list.	

4.12 N-FEE windowing mode simulation

4.12.1 N-FEE windowing mode management

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-554	In the NFEE_RUNNING_WINDOWING state, a N-FEE simulation entity shall simulate the behaviour of a real N-FEE running in WINDOWING mode as defined in [AD1], [AD2] and [AD5].	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-947	In the NFEE_RUNNING_WINDOWING_PATTERN state, a N-FEE simulation entity shall simulate the behaviour of a real N-FEE running in WINDOWING PATTERN mode as defined in [AD1], [AD2] and [AD5].	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-555	Upon reception of the dedicated change mode RMAP command from the N-DPU and if the N-FEE simulation entity is in the NFEE_RUNNING_STANDBY state, the N-FEE simulation entity shall enter the NFEE_RUNNING_WINDOWING state.	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-948	Upon reception of the dedicated change mode RMAP command from the N-DPU and if the N-FEE simulation entity is in the NFEE_RUNNING_ON state, the N-FEE simulation entity shall enter the NFEE_RUNNING_WINDOWING_PATTERN state.	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-656	The effective transition to the NFEE_RUNNING_WINDOWING sate shall be performed depending on the synchronization signal and by fulfilling the timing specified in [AD1], [AD2] and [AD5].	

Note: In the rest of the document, the assumption is made that the effective transition is performed on the release of a FFS synchronization signal.

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-673	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall transmit to the N-DPU after each FSS:
	· one HK packet
	· a set of data packets containing window segments corresponding to the window position list
	· a set of data packets containing OLN overscan lines
	  · The Y-position in the CCD image of the first overscan line to be transmitted is 4510
	  · The Y-position in the CCD image of the last overscan line to be transmitted is given by the formula 4510 +OLN -1	

See [AD3] for the definition of the CCD image coordinate reference.

	Title:	**Deleted**
	Type:	
	Verif.:	
	LastEdit:	2019-12-06
NFEESIM-UR-833	Deleted	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-831	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall start the transmission to the N-DPU of a new set of window segments and overscan lines at each release of the FSS synchronization signal.	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-675	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall have completed the transmission of all the window segments and overscan lines RT seconds after the release of the FSS synchronization signal.	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-20
NFEESIM-UR-788	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall be able to transmit over RT seconds a maximum amount of data of 29.2 Mb corresponding to: · 44123 36-pixel windows · OLN overscan lines (which contains the smearing data)	

Note: 29.2 Mb transmitted over 4s corresponds to a data rate of 7.6 Mbps.

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-516	In NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall mimic the window extraction functionality implemented by the real N-FEE.	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test

	LastEdit:	2019-11-08
NFEESIM-UR-517	In NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall transmit the data to the N-DPU with timing similar to the real N-FEE one.	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-513	In NFEE_RUNNING_WINDOWING state, the N-FEE simulation entity shall be able to manage the following modes of data generation: · N-FEE windowing / pattern mode · N-FEE windowing / SSD image mode · N-FEE windowing / SSD window mode	

Note: The modes mentioned above correspond to fullimage injection modes: they define how the fullimages which are the input to the windowing operations are built.

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-520	At startup, by default, each N-FEE simulation entity shall use the N-FEE windowing / pattern mode.	

	Title:	**N-FEE windowing mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-844	In NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall manage pattern modes for data generation	

4.12.2 Window list management

	Title:	**Window list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-518	In NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall use window position lists as defined in [AD2] to: · extract the window segments from the fullimage injected as inputs, · assembly them in SpaceWire packets sent to the N-DPU.

	Title:	**Window list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-20
NFEESIM-UR-791	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation entity shall be able to manage, for each CCD, one window position list allowing to define the following amount of windows: · 44123 windows	

Note 1: The figure 44123 is justified in [RD8].

Note 2: The 4510 30-pixel smearing windows are not defined in the window position lists: the data corresponding to these windows are conveyed as overscan lines.

	Title:	**Window list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-574	In any of the NFEE_RUNNING states, the N-FEE simulation entity shall be able to change dynamically the content of its current window position lists by processing the ad-hoc RMAP commands sent by the N-DPU during the idle time (i.e. time between two readout phases).	

Note: This capability is crucial to be able to validate the window position list construction functionality implemented in the N-DPU ASW.

	Title:	**Window list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-569	When entering the NFEE_CONFIG state, the N-FEE simulation entity shall set as current window position lists a set of default window position lists stored in its non-volatile memory.	

	Title:	**Window list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-576	The SimuCam MEB shall allow the user to change, via a specific Ethernet command, the default window position lists of a given N-FEE simulation entity.

	Title:	**Window list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-575	The SimuCam MEB shall allow the user to request, via a specific Ethernet command, the window position lists managed by each N-FEE simulation entity.	

Note: this function is useful for debugging and verification purpose.

4.12.3 Packet order list management

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1101	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE shall manage a packet order list for each CCD.	

A packet order list is a list of n entries, with n<=512, each entry indicating a CCD side (left or right).

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1105	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the packet order lists shall be built from in memory associated data.	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1135	In any of the NFEE_RUNNING states, the N-FEE simulation entity shall be able to change dynamically the content of its current packet order lists by processing the ad-hoc RMAP commands sent by the N-DPU during the idle time (i.e. time between two readout phases).	

	Title:	**Packet order list management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1136	When entering the NFEE_CONFIG state, the N-FEE simulation entity shall set as current packet order lists a set of default packet order list stored in its non-volatile memory.	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1137	The SimuCam MEB shall allow the user to change, via a specific Ethernet command, the default packet order lists of a given N-FEE simulation entity.	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1138	The SimuCam MEB shall allow the user to request, via a specific Ethernet command, the packet order lists managed by each N-FEE simulation entity.	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1106	The address where to retrieve the 512 bits of the packet order list for a given CCD shall be retrieved from ccd<x>_pktorder_list_ptr register, with <x> in [1,2,3,4] (see [AD6]).	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1107	The 512 bits of the in memory packet order list for a given CCD shall be interpreted as · bit = 1 : Left CCD side · bit = 0 : Right CCD side	

	Title:	**Packet order list management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1109	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation shall maintain two buffers (DPBLEFT and DPBRIGHT) to bufferize data packet buffers for each half CCD.	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1110	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation shall manage two parameters DPBLENLEFT and DPBLENRIGHT for data packet buffer length.	

Default values for DPBLENLEFT and DPBLENRIGHT are TBD bytes.

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1112	The SimuCam MEB shall allow the user to modify default values for DPBLENLEFT and DPBLENRIGHT.	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1113	The SimuCam MEB shall allow the user to modify, via an ethernet command, values for DPBLENLEFT and DPBLENRIGHT.	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1115	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation shall transmit data to N-DPU with respect to packet order list management algorithm.	

We descrive hereafter packet order algorithm.

When building Data packets for windows mode, data from each side of the CCD shall be separated in different packets.

Each time a packet is full (i.e. packet size is equal to DATAPACKLEN),

- a buffer is allocated for both side of the CCD to store unsent data packet
- if the current entry of the packet order list match the packet asscociated CCD side
  - the packet is sent,
  - the system point to the next entry of the packet order list
  - while there is store delayed sending packet and current entry of the packet order list match the delayed sending packet side
    - sent stored delayed sending packet
    - point to the next entry of the packet order list
- otherwise it is stored for delayed sending

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1103	In the NFEE_RUNNING_WINDOWING and NFEE_RUNNING_WINDOWING_PATTERN state, the N-FEE simulation shall set dedicated registers entries in case of errors (see error_flag register in [AD6]).	

	Title:	**Packet order list management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-1116	The SimuCam MEB shall allow the user to request, via a specific Ethernet command, the dump of the packet order list manage for each CCD.	

4.12.4 N-FEE windowing / pattern mode

	Title:	**N-FEE windowing / pattern mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-556	The SimuCam MEB shall allow the user to configure, via an Ethernet command, a N-FEE simulation entity to work in the N-FEE windowing / pattern mode.	

	Title:	**N-FEE windowing / pattern mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-515	When the N-FEE windowing / pattern mode is activated, the N-FEE simulation entity shall build on the fly, using patterns as defined in [AD3], the fullimages which are used as input of the windowing operation.

4.12.5 N-FEE windowing / SSD image mode

	Title:	**N-FEE windowing / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-521	The SimuCam MEB shall allow the user to configure, via an Ethernet command, a N-FEE simulation entity to work in the N-FEE windowing / SSD image mode.	

	Title:	**N-FEE windowing / SSD image mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-522	When the N-FEE windowing / SSD image mode is activated, the N-FEE simulation entity shall use, as input of the windowing operation, fullimages from the SSD according to the active fullimage ID list and according to the cycle described in NDPU-GSE-UR-458.	

4.12.6 N-FEE windowing / SSD window mode

The SimuCam MEB will be intensively used for validating the scientific algorithms implemented in the N-DPU ASW. This validation process will be performed using test data provided by the DPA team: the test data will be produced by the PLATO Image Simulator (PIS) developed by the DPA team. The PIS tool generates batches of windows and not fullimages. To optimize the overall test process, the SimuCam MEB shall offer a working mode in which the data input are windows and not fullimages. Without this working mode, if we want for example test the photometry algorithm on only one star but for 1000 exposures, we would be forced to generate 1000 fullimages and to load in the SimuCam MEB an amount of data corresponding to 1000 x 40 MBytes = 40 GBytes. With this working mode, we could download in the SimuCam a window stack whose the size would be 10*10*2*1000 = 200 kBytes (where the size of window is 10*10 pixels): the gain in term of data to be transferred is huge.

The requirements below describe more in detail the needs related to this working mode.

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-581	The SimuCam MEB shall allow the user to configure, via an Ethernet command, a N-FEE simulation entity to work in the N-FEE windowing / SSD window mode.	

	Title:	**N-FEE windowing / SSD window mode**

	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-614	When the N-FEE fullimage / SSD window mode is activated, the N-FEE simulation entity shall use fullimages patched with windows from the SSD as data sent to the N-DPU.	

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-615	For each N-FEE simulation entity, the SimuCam MEB shall allow to manage the following parameters driving the N-FEE windowing / SSD window mode: · Number of exposures: EN where 1 < EN < 65536 · Maximal number of windows to be patched by CCD: PN_MAX = 50000 · ID of the CCD fullimage which will be patched: CCD1_ID, CCD2_ID, CCD3_ID, CCD4_ID · Number of windows to be patched in CCD1: PN1 where 0 < PN1 < PN_MAX · Number of windows to be patched in CCD2: PN2 where 0 < PN2 < PN_MAX · Number of windows to be patched in CCD3: PN3 where 0 < PN3 < PN_MAX · Number of windows to be patched in CCD4: PN4 where 0 < PN4 < PN_MAX · List of the window stack ID and window positions for the CCD1: · $CCD1\_WS\_ID_1$, $CCD1\_WS\_X_1$, $CCD1\_WS\_Y_1$, $CCD1\_WS\_SIDE_1$ · $CCD1\_WS\_ID_2$, $CCD1\_WS\_X_2$, $CCD1\_WS\_Y_2$, $CCD1\_WS\_SIDE_2$ · ... · $CCD1\_WS\_ID_{PN1}$, $CCD1\_WS\_X_{PN1}$, $CCD1\_WS\_Y_{PN1}$, $CCD1\_WS\_SIDE_{PN1}$ · List of the window stack ID and window positions for the CCD2: · $CCD2\_WS\_ID_1$, $CCD2\_WS\_X_1$, $CCD2\_WS\_Y_1$, $CCD2\_WS\_SIDE_1$ · $CCD2\_WS\_ID_2$, $CCD2\_WS\_X_2$, $CCD2\_WS\_Y_2$, $CCD2\_WS\_SIDE_2$ · ... · $CCD2\_WS\_ID_{PN2}$, $CCD2\_WS\_X_{PN2}$, $CCD2\_WS\_Y_{PN2}$, $CCD2\_WS\_SIDE_{PN2}$ · List of the window stack ID and window positions for the CCD3: · $CCD3\_WS\_ID_1$, $CCD3\_WS\_X_1$, $CCD3\_WS\_Y_1$, $CCD3\_WS\_SIDE_1$ · $CCD3\_WS\_ID_2$, $CCD3\_WS\_X_2$, $CCD3\_WS\_Y_2$, $CCD3\_WS\_SIDE_2$ · ... · $CCD3\_WS\_ID_{PN3}$, $CCD3\_WS\_X_{PN3}$, $CCD3\_WS\_Y_{PN3}$, $CCD3\_WS\_SIDE_{PN3}$ · List of the window stack ID and window positions for the CCD4: · $CCD4\_WS\_ID_1$, $CCD4\_WS\_X_1$, $CCD4\_WS\_Y_1$, $CCD4\_WS\_Y_1$, $CCD3\_WS\_SIDE_1$ · $CCD4\_WS\_ID_2$, $CCD4\_WS\_X_2$, $CCD4\_WS\_Y_2$, $CCD4\_WS\_Y_1$, $CCD3\_WS\_SIDE_2$

	· ... · CCD4_WS_ID$_{PN4}$, CCD4_WS_X$_{PN4}$, CCD4_WS_Y$_{PN4}$, CCD4_WS_Y$_1$, CCD3_WS_SIDE$_{PN4}$

Note: The figures given in the requirement above, in particular PN_MAX, could be adjusted depending on the limitations of the hardware resources. It could be also acceptable, that the N-FEE windowing mode / SSD window mode could be limited to a reduced number of N-FEE simulation entities. This feeding mode will be mainly used in a test configuration dedicated to the N-DPU ASW test where only one N-DPU board is used and then only 2 N-FEE simulation entities are used.

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-617	For each N-FEE simulation entity, the SimuCam MEB shall manage and store in its non-volatile memory a default set of parameters driving the N-FEE windowing / SSD window mode (window stack lists).	

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-618	For each N-FEE simulation entity in NFEE_CONFIG state, the SimuCam MEB shall allow the user to change the default set of parameters driving the N-FEE windowing / SSD window mode (window stack lists).	

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-619	For each N-FEE simulation entity in NFEE_CONFIG state, the SimuCam MEB shall allow the user to change the active set of parameters driving the N-FEE windowing / SSD window mode (window stack lists).	

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-625	For each N-FEE simulation entity in NFEE_CONFIG state, the SimuCam MEB shall check each new programmed active set of parameters driving the N-FEE windowing / SSD window mode and reject it if: · the number of entries of each window stack is not the same, · EN is greater than the number of entries of each window stack.

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-620	For each N-FEE simulation entity, at startup, the SimuCam MEB shall use the default set of parameters driving the N-FEE windowing / SSD window mode (CCD ID list) as active set of parameters.	

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08

NFEESIM-UR-621	When the N-FEE windowing / SSD window mode is activated, the N-FEE simulation entity shall use, as input of the windowing process, fullimages built according to the following cycle and operations (t0 is the start of the cycle and the assumption is made here that FN = 0 at t0): · t0: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_1$, CCD1_WS_ID$_2$_E$_1$, ..., CCD1_WS_ID$_{PN1}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · t0+0.25EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_1$, CCD2_WS_ID$_2$_E$_1$, ..., CCD2_WS_ID$_{PN2}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · t0+0.5EP: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_1$, CCD3_WS_ID$_2$_E$_1$, ..., CCD3_WS_ID$_{PN3}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · t0+0.75EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_1$, CCD4_WS_ID$_2$_E$_1$, ..., CCD4_WS_ID$_{PN4}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · t0+EP: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_2$, CCD1_WS_ID$_2$_E$_2$, ..., CCD1_WS_ID$_{PN1}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack. · t0+1.25EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_2$, CCD2_WS_ID$_2$_E$_2$, ..., CCD2_WS_ID$_{PN2}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack. · t0+1.5EP: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_2$, CCD3_WS_ID$_2$_E$_2$, ..., CCD3_WS_ID$_{PN2}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack. · t0+1.75EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_2$, CCD4_WS_ID$_2$_E$_2$, ..., CCD4_WS_ID$_{PN4}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack. · ... · t0+(EN-1)EP: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_{EN}$, CCD1_WS_ID$_2$_E$_{EN}$, ..., CCD1_WS_ID$_{PN1}$_E$_{EN}$ where E$_{EN}$ correspond to the entry #EN in each stack. · t0+(EN-0.75)EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_{EN}$, CCD2_WS_ID$_2$_E$_{EN}$, ..., CCD2_WS_ID$_{PN2}$_E$_{EN}$ where E$_{EN}$ correspond to the entry #EN in each stack. · t0+(EN-0.5)EP: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_{EN}$, CCD3_WS_ID$_2$_E$_{EN}$, ..., CCD3_WS_ID$_{PN3}$_E$_{EN}$ where E$_{EN}$ correspond to the entry #EN in each stack.

	· t0+(EN-0.25)EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_{EN}$, CCD4_WS_ID$_2$_E$_{EN}$, ..., CCD4_WS_ID$_{PN4}$_E$_{EN}$ where E$_{EN}$ correspond to the entry #EN in each stack. · t0+(EN)EP: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_1$, CCD1_WS_ID$_2$_E$_1$, ..., CCD1_WS_ID$_{PN1}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · t0+(EN+0.25)EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_1$, CCD2_WS_ID$_2$_E$_1$, ..., CCD2_WS_ID$_{PN2}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · t0+(EN+0.5)EP: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_1$, CCD3_WS_ID$_2$_E$_1$, ..., CCD3_WS_ID$_{PN3}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · t0+(EN+0.75)EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_1$, CCD4_WS_ID$_2$_E$_1$, ..., CCD4_WS_ID$_{PN4}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack. · ...

Note: the sequence given above is true if CCD are associated to frame in the following way : Frame1 = CCD1, Frame2 = CCD2, Frame3 = CCD3 and Frame4 = CCD4.

If for example the configuration was Frame1 = CCD3, Frame2 = CCD4, Frame3 = CCD1 and Frame4 = CCD2, then the sequence would be:

- t0: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_1$, CCD3_WS_ID$_2$_E$_1$, ..., CCD3_WS_ID$_{PN1}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack.
- t0+0.25EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_1$, CCD4_WS_ID$_2$_E$_1$, ..., CCD4_WS_ID$_{PN2}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack.
- t0+0.5EP: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_2$, CCD1_WS_ID$_2$_E$_2$, ..., CCD1_WS_ID$_{PN3}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack.
- t0+0.75EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_2$, CCD2_WS_ID$_2$_E$_1$, ..., CCD2_WS_ID$_{PN4}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack.
- t0+EP: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_2$, CCD3_WS_ID$_2$_E$_2$, ..., CCD3_WS_ID$_{PN1}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack.
- t0+1.25EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_2$, CCD4_WS_ID$_2$_E$_2$, ..., CCD4_WS_ID$_{PN2}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack.
- t0+1.5EP: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_3$, CCD1_WS_ID$_2$_E$_3$, ..., CCD1_WS_ID$_{PN2}$_E$_3$ where E$_3$ correspond to the entry #3 in each stack.

- t0+1.75EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_3$, CCD2_WS_ID$_2$_E$_3$, ..., CCD2_WS_ID$_{PN4}$_E$_3$ where E3 correspond to the entry #3 in each stack.
- ...
- t0+(EN-1)EP: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_{EN}$, CCD3_WS_ID$_2$_E$_{EN}$, ..., CCD3_WS_ID$_{PN1}$_E$_{EN}$ where E$_{EN}$ correspond to the entry #EN in each stack.
- t0+(EN-0.75)EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_{EN}$, CCD4_WS_ID$_2$_E$_{EN}$, ..., CCD4_WS_ID$_{PN2}$_E$_{EN}$ where E$_{EN}$ correspond to the entry #EN in each stack.
- t0+(EN-0.5)EP: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_1$, CCD1_WS_ID$_2$_E$_1$, ..., CCD1_WS_ID$_{PN3}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack.
- t0+(EN-0.25)EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_1$, CCD2_WS_ID$_2$_E$_1$, ..., CCD2_WS_ID$_{PN4}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack.
- t0+(EN)EP: fullimage CCD3_ID patched with windows from the window stacks CCD3_WS_ID$_1$_E$_1$, CCD3_WS_ID$_2$_E$_1$, ..., CCD3_WS_ID$_{PN1}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack.
- t0+(EN+0.25)EP: fullimage CCD4_ID patched with windows from the window stacks CCD4_WS_ID$_1$_E$_1$, CCD4_WS_ID$_2$_E$_1$, ..., CCD4_WS_ID$_{PN2}$_E$_1$ where E$_1$ correspond to the entry #1 in each stack.
- t0+(EN+0.5)EP: fullimage CCD1_ID patched with windows from the window stacks CCD1_WS_ID$_1$_E$_2$, CCD1_WS_ID$_2$_E$_2$, ..., CCD1_WS_ID$_{PN3}$_E$_2$ where E$_2$ correspond to the entry #2 in each stack.
- t0+(EN+0.75)EP: fullimage CCD2_ID patched with windows from the window stacks CCD2_WS_ID$_1$_E$_2$, CCD2_WS_ID$_2$_E$_2$, ..., CCD2_WS_ID$_{PN4}$_E2 where E$_2$ correspond to the entry #2 in each stack.
- ...

	Title:	**N-FEE windowing / SSD window mode**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-622	When a N-FEE simulation entity has reached the end of the entry #EN, it shall start again at the entry #1.	

4.13 N-FEE Trap pumping mode simulation

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21

NFEESIM-UR-951	In the NFEE_RUNNING_PARALLELE_TRAP_PUMP_1, NFEE_RUNNING_PARALLELE_TRAP_PUMP_2, NFEE_RUNNING_SERIAL_TRAP_PUMP_1 and NFEE_RUNNING_SERIAL_TRAP_PUMP_2 state, the N-FEE simulation entities shall simulate the behaviour of a real N-FEE running respectiveley in Parallele trap pumping mode 1, Parallele Trap pumping mode 2, Serial trap pumping mode 1 and Serial trap pumping mode 2 as defined in [AD1], [AD2] and [AD5].

When NFEE_RUNNING_PARALLELE_TRAP_PUMP_1, NFEE_RUNNING_PARALLELE_TRAP_PUMP_2, NFEE_RUNNING_SERIAL_TRAP_PUMP_1 and NFEE_RUNNING_SERIAL_TRAP_PUMP_2 state are concerned we use hereafter the notation NFEE_RUNNING_xxx_TRAP_PUMP_y

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-953	In each NFEE_RUNNING_xxx_TRAP_PUMP_y state, the N-FEE simulation entities shall generate the time-code.	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-954	Upon reception of the dedicated change mode RMAP command from the N-DPU and if the N-FEE simulation entity is in NFEE_RUNNING_STANDBY state, the N-FEE simulation entity shall enter the required NFEE_RUNNING_xxx_TRAP_PUMP_y state.	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-962	The effective transition to the NFEE_RUNNING_xxx_TRAP_PUMP_y state shall be performed depending on the synchronization signal and by fulfilling the timing specified in [AD1], [AD2] and [AD5].	

Concerning the mode transition, the N-FEE simulation entity shall mimic the real N-FEE: this is crucial for qualifying the N-DPU ASW.

In the rest of the document, the assumption is made that the effective transition is performed :

- immmediately if the transition in due to an Immediate_on command
- on the release of the next FFS synchronization signal for classical change mode command

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-971	Immediately after entering NFEE_RUNNING_xxx_TRAP_PUMP_y state the N-FEE simulation entity shall execute a pumping simulation process.	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-972	While still in the NFEE_RUNNING_xx_TRAP_PUMP_y state the N-FEE simulation shall start each new pumping simulation process on the next FFS following the end of the previous pumping simulation process	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-973	The pumping simulation process shall be divided into two parts: · the pumping · the data emitting	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-974	During the pumping part of the pumping simulation process no data shall be emitted (except periodical HK and timecode).	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-975	During the data emitting part of the pumping simulation process simulated data shall be emitted by fulfilling timing similar to those implemented in the real N-FEE	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test

	LastEdit:	2019-11-21
NFEESIM-UR-976	The duration D of the pumping part of the pumping simulation process is computed through the following formula D =CI+((900+DT)xSH-6.5)x10^{-6} sec. (TBC) where · CI : Charge injection = 0.5 sec. · DT : Dwell time = TBD · SH : Shuffles = TBD	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-977	The data emitting process shall start on the first PFS occuring after pumping part ending.	

The graphic hereafter describes the Trap pumping simulation in parallele mode:

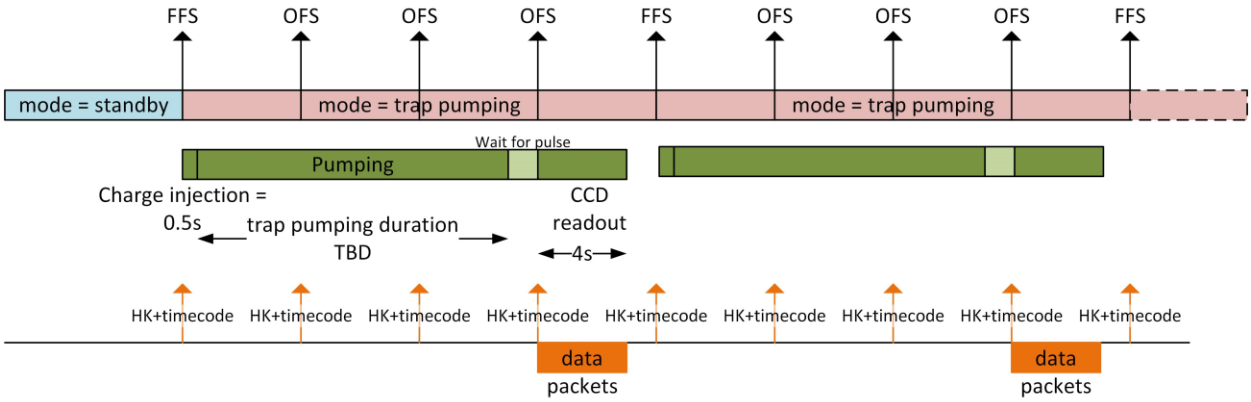

Figure 10 - N-FEE trap pumping mode timeline

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-978	In the NFEE_RUNNING_PARALLELE_TRAP_PUMP_1 and NFEE_RUNNING_PARALLELE_TRAP_PUMP_2 state, the data emitted shall corresponds to a half CCD.	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-982	In NFEE_RUNNING_PARALLELE_TRAP_PUMP_1 and NFEE_RUNNING_PARALLELE_TRAP_PUMP_2 state, the N-FEE simulation entity shall build, on the fly, half-CCD images which are filled with patterns according to [AD3].	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-979	In the NFEE_RUNNING_SERIAL_TRAP_PUMP_1 and NFEE_RUNNING_SERIAL_TRAP_PUMP_2 state, the data emitted shall corresponds to a single row of a half CCD.	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-983	In NFEE_RUNNING_SERIAL_TRAP_PUMP_1 and NFEE_RUNNING_SERIAL_TRAP_PUMP_2 state the N-FEE simulation entity shall build, on the fly, a single row of a half-CCD images which are filled with patterns according to [AD3].	

	Title:	**N-FEE trap pumping mode management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1094	In each NFEE_RUNNING_xx_TRAP_PUMP_y, the half CCD to be transmitted is the one configured to the first frame at pumping start (TBC).	

4.14 N-FEE Immediate ON command simulation

	Title:	**Immediate ON command simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07
NFEESIM-UR-943	On reception of Immediate On any data packet that are still to be emitted are to be canceled.	

	Title:	**Immediate ON command simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-944	On reception of immediate On command any data packet ongoing transmission shall be stopped and the transmission terminated by a RMAP EEP character (TBC).	

4.15 N-FEE housekeeping simulation

	Title:	**N-FEE housekeeping simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-579	In any of the NFEE_RUNNING states, the N-FEE simulation entity shall transmit to the N-DPU after each FSS a SpaceWire packet containing a simulated N-FEE HK block (the expected behaviour is defined in [AD2]).	

	Title:	**N-FEE housekeeping simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-644	In any of the NFEE_RUNNING states, the N-FEE simulation entity shall be able to transmit to the N-DPU, on reception of a RMAP request, a RMAP response containing one specific HK parameter (if only one parameter has been requested) or a simulated N-FEE HK block containing several HK parameters which are located contiguously in memory (if a block of parameter has been requested).	

	Title:	**N-FEE housekeeping simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-643	The N-FEE simulation entity shall manage autonomously, depending on the simulation state, the values of the N-FEE HK parameters belonging to the following categories: · N-FEE mode · SpaceWire status · Frame number register · CCD status · See [AD5] for the detailed list.	

Note: The user cannot change by command the values of the parameters listed above.

	Title:	**N-FEE housekeeping simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-645	The SimuCam MEB shall allow, for each N-FEE simulation entity, whatever its state, to set by an Ethernet command the values of the N-FEE HK parameters belonging to the following categories: · Temperatures · Voltages · Currents · See [AD5] for the detailed list.	

	Title:	**N-FEE housekeeping simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-647	At startup, the SimuCam MEB shall configure the values of the N-FEE HK parameters with default values stored in its non-volatile memory.	

	Title:	**N-FEE housekeeping simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-648	The SimuCam MEB shall allow the user to change the values of the default N-FEE HK parameters stored in its non-volatile memory.	

4.16 N-FEE configuration simulation

	Title:	**N-FEE configuration simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-907	The SimuCam MEB shall react, for each N-FEE simulation entity, to all the configuration commands transmitted by RMAP which are relevant for the simulation.	

	Title:	**N-FEE configuration simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-07

NFEESIM-UR-908	The SimuCam MEB shall manage, for each N-FEE simulation entity, the following commands (see [AD5]): · mode selection control · SpaceWire packet configuration #1 which includes: · Digitise control · CCD port data transmission selection control · Packet size control · TBD

	Title:	**N-FEE configuration simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-08
NFEESIM-UR-909	The SimuCam MEB shall manage, for each N-FEE simulation entity, the following commands which are specific to the windowing features (see [AD5]): · SpaceWire packet configuration #2 which includes: · CCD windowing configuration · Window width · Window height · TBD	

4.17 RMAP register simulation

	Title:	**RMAP register simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-15
NFEESIM-UR-1070	The SimuCam MEB shall simulate all the registers define in [AD6].	

	Title:	**RMAP register simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-15
NFEESIM-UR-1073	The SimuCam MEB shall set default value to register as defined in [AD6].	

	Title:	**RMAP register simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13

NFEESIM-UR-1074	The SimuCam MEB shall manage Read and Write access to all configuration simulated registers.

Configuration registers are all those whose register name is postfixed "_config" (from 0x00000000 to 0x0000005C).

	Title:	**RMAP register simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-13
NFEESIM-UR-1076	The SimuCam MEB shall manage Read RMAP access to all HK simulated registers.	

HK registers are all those whose register name is postfixed "_hk" (from 0x00000700 to 0x000007BC).

	Title:	**RMAP register simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-20

NFEESIM-UR-1078	The SimuCam MEB shall fully simulate the following config registers (sub-register name extracted from [AD6]): · v_start · v_end · ccd_readout_order · int_sync_period · Trap_Pumping_Dwell_counter · sensor_sel · digitise_en · ccd1_win_list_ptr · ccd1_pktorder_list_ptr · ccd1_win_list_length · ccd1_win_size_x · ccd1_win_size_y · ccd2_win_list_ptr · ccd2_pktorder_list_ptr · ccd2_win_list_length · ccd2_win_size_x · ccd2_win_size_y · ccd3_win_list_ptr · ccd3_pktorder_list_ptr · ccd3_win_list_length · ccd3_win_size_x · ccd3_win_size_y · ccd4_win_list_ptr · ccd4_pktorder_list_ptr · ccd4_win_list_length · ccd4_win_size_x · ccd4_win_size_y · ccd_mode_config · ccd1_last_Epacket · ccd1_last_Fpacket · ccd2_last_Epacket · ccd2_last_Fpacket · ccd3_last_Epacket · ccd3_last_Fpacket · ccd4_last_Epacket · ccd4_last_Fpacket · clear_error_flag · Readout_pause_counter · Trap_Pumping_Shuffle_counter

By fully simulate we means that register content modification shall be reflected on the simulation behavior

	Title:	**RMAP register simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-20
NFEESIM-UR-1079	The SimuCam MEB shall fully simulate the following HK registers (sub-register name extracted from [AD6]): · spw_status · frame_counter · op_mode · frame_number · error_flags	

The content of those registers shall reflect the actual status of the simulation (e.g. last spacewire errors flag shall be set if any spacewire error occurs).

	Title:	**RMAP register simulation**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-12-20

NFEESIM-UR-1080	The SimuCam MEB shall provide ethernet command to change the content of the following HK registers (sub-register name extracted from [AD6]): · TOU_SENSE_1 · TOU_SENSE_2 · TOU_SENSE_3 · TOU_SENSE_4 · TOU_SENSE_5 · TOU_SENSE_6 · CCD1_TS · CCD2_TS · CCD3_TS · CCD4_TS · PRT1 · PRT2 · PRT3 · PRT4 · PRT5 · ZERO_DIFF_AMP · CCD1_VOD_MON · CCD1_VOG_MON · CCD1_VRD_MON_E · CCD2_VOD_MON · CCD2_VOG_MON · CCD2_VRD_MON_E · CCD3_VOD_MON · CCD3_VOG_MON · CCD3_VRD_MON_E · CCD4_VOD_MON · CCD4_VOG_MON · CCD4_VRD_MON_E · VCCD · VRCLK_MON · VICLK · VRCLK_LOW · 5VB_POS_MON · 5VB_NEG_MON · 3V3B_MON · 2V5A_MON · 3V3D_MON · 2V5D_MON · 1V5D_MON

	· 5VREF_MON · VCCD_POS_RAW · VCLK_POS_RAW · VAN1_POS_RAW · VAN3_NEG_MON · VAN2_POS_RAW · VDIG_RAW · VDIG_RAW_2 · VICLK_LOW · CCD1_VRD_MON_F · CCD1_VDD_MON · CCD1_VGD_MON · CCD2_VRD_MON_F · CCD2_VDD_MON · CCD2_VGD_MON · CCD3_VRD_MON_F · CCD3_VDD_MON · CCD3_VGD_MON · CCD4_VRD_MON_F · CCD4_VDD_MON · CCD4_VGD_MON · IG_HI_MON · IG_LO_MON · TSENSE_A · TSENSE_B

4.18 Error injection

4.18.1 Error injection management

The error injection capability will be used for testing all the various failure cases. This capability is crucial to cover by test all the software functions related to the failure management.

	Title:	**Error injection management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-659	The SimuCam MEB shall allow the user to generate errors independently on each N-FEE simulation link.	

	Title:	**Error injection management**
	Type:	Functional

	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-419	The SimuCam MEB shall allow the user to trigger the errors (SpaceWire, RMAP, etc.) at any moment.	

	Title:	**Error injection management**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-723	The SimuCam MEB shall allow the user to set, in each error injection command, how many consecutive times the error has to be repeated: · one shot (the error is injected only one time) · n times	

4.18.2 SpaceWire errors

	Title:	**SpaceWire errors**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-418	The SimuCam MEB shall allow the user to generate on demand the following SpaceWire errors: · Errors at exchange level: parity, disconnect, escape sequence, character sequence, credit. · Errors at network level: EEP received, invalid destination address.	

Note: The SpaceWire exchange level errors are tricky to generate without a specific support of the SpW IP core.

4.18.3 RMAP errors

	Title:	**RMAP errors**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-420	The SimuCam MEB shall allow the user to generate on demand the following RMAP errors: · missing responses · injection of any failure code in the RMAP reply	

4.18.4 Time-code errors

	Title:	**Time-code errors**

	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-481	The SimuCam MEB shall allow the user to generate on demand the following time-code errors: · time-code missing · wrong time-code value (a time code is received at the right time but its value is wrong) · unexpected time-code (a time-code is received at an unexpected moment) · Jitter on time-code	

4.18.5 Fullimage transmission errors

	Title:	**Fullimage transmission errors**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-421	The SimuCam MEB shall allow the user to generate on demand the following fullimage transmission errors: · missing packets: the packet is identified by its position number inside a train of packets · missing data inside packets: the packet is identified by its position number inside a train of packets; the data to be removed in the packet is identified by an offset and a size · transmission disabled: no more packets are transmitted	

4.18.6 Window transmission errors

	Title:	**Window transmission errors**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-480	The SimuCam MEB shall allow the user to generate on demand the following window transmission errors: · missing packets: the packet is identified by its position number inside a train of packets · missing data inside packets: the packet is identified by its position number inside a train of packets; the data to be removed in the packet is identified by an offset and a size · transmission disabled: no more packets are transmitted	

	Title:	**Window transmission errors**

	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-571	In MEB_RUNNING mode, the SimuCam MEB shall allow the user to enable / disable, via a specific Ethernet command, the current window position list of a given N-FEE simulation entity.	

Note: this function is useful for testing some failure cases (windowing not programmed at N-FEE level or windowing erased due to an N-FEE malfunction). When a window position list has been disabled, the N-FEE simulation entity is no more able to generate data packets containing the window segments. However, the HK packet and the overscan data packets continue to be produced.

4.18.7 Image and Window content errors

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-989	The SimuCam MEB shall allow the user to inject an error on a given pixel.	

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-990	An injected error is identified though four parameters · X and Y: position of the pixel on an half CCD · SIDE: half-CCD side · P : count of frames to wait after process starting before the error is activated · N : count of frames to maintain the error activated	

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-992	The SimuCam MEB shall manage up to 100 pixel errors.	

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test

	LastEdit:	2019-11-21
NFEESIM-UR-993	The SimuCam shall provide a command to upload a single error.	

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-994	The SimuCam MEB shall provide a command to start pixel error injection.	

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-999	After reception of the start pixel error injection, error applying process shall be started at the next FFS.	

When this command is received the whole set of upload errors is applied.

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-997	The SimuCam MEB shall manage more than one execution of the start pixel error injection command.	

The Uploaded pixel errors shall not be cleared after execution.

If an error injection process is already running, it will be stopepd at next FFS as a new one is started, resetting frame counter P and already injected errors N to 0 for each error.

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1002	When no more error is to be applied, the pixel error applying process shall be stopped.	

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21

NFEESIM-UR-1001	The SimuCam MEB shall reject any command for error injection when an error injection process is running.

	Title:	**Image and window content error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-996	The SimuCam MEB shall provide a command to clear the uploaded pixel errors.	

4.18.8 Data packet header error

	Title:	**Data packet header error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1117	The SimuCam MEB shall allow the user to inject error on the header of any transmitted data packet.	

	Title:	**Data packet header error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1118	A data packet header error is identified through · FC : the frame which contains the packet the error to be applied on · PN : the packet number on the selected frame on which the error shall be applied · FIELD : the field to be modified in [MODE, LAST_PACKET, CCD_SIDE,CCD_NUMBER,FRAME_NUMBER,DATA_PACKET_TYPE,FRAME_COUNTER,SEQUENCE_COUNTER,LENGTH] as defined in [AD2] · VALUE : the value to apply to the field (w.r.t associated field type and range)	

	Title:	**Data packet header error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1119	The SimuCam MEB shall manage up to 10 data packet header errors.	

	Title:	**Data packet header error**
	Type:	Functional
	Verif.:	Test

	LastEdit:	2019-11-21
NFEESIM-UR-1120	The SimuCam MEB shall provide an ethernet command to upload a single data packet header error.	

	Title:	**Data packet header error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1121	The SimuCam MEB shall provide a command to start data packet error injection.	

	Title:	**Data packet header error**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-1122	After reception of the start data packet error injection, error applying process shall start at the next FFS.	

When this command is received the whole set of uploaded errors is applied.

4.19 Status / event / log reporting

4.19.1 Status reporting

	Title:	**Status reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-425	The SimuCam MEB shall report periodically, as an Ethernet packet, its global status and the status of each N-FEE simulation entity.	

	Title:	**Status reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21

NFEESIM-UR-546	The MEB global status shall contain at least the following information: · active mode · EP · RT · synchronization signal source (internal / external) · information related to the SSD storage (see SSD storage status) · last command · time of the last command · last error / warning ID or label · time of the last error / warning

	Title:	**Status reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-547	The N-FEE simulation entity status shall contain at least the following information: · active mode · SpaceWire link state · packet statistics: count of incoming packets, count of outgoing packets · failure statistics · ~~logical address of the last RMAP command~~ · ~~last RMAP command (full packet content reported as an hexa string)~~ · ~~time of the last RMAP command~~ · last error / warning ID or label · time of the last error / warning	

	Title:	**Status reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-427	The SimuCam MEB shall allow the user to configure, via an Ethernet command, the generation period of its status reports.	

	Title:	**Status reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12

NFEESIM-UR-466	The SimuCam MEB shall allow the user to enable / disable, via an Ethernet command, the periodic reporting of its status reports.

	Title:	**Status reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-12
NFEESIM-UR-465	The SimuCam MEB shall allow the user to request on demand the status reports.	

4.19.2 Event reporting

	Title:	**Event reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-467	The SimuCam MEB shall generate, as an Ethernet packet, a sporadic report to notify the user of the relevant progress or failure events.	

	Title:	**Event reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-468	The SimuCam MEB shall identify each progress or failure events thanks to an identifier.	

	Title:	**Event reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-469	The SimuCam MEB shall allow the user to enable / disable, via an Ethernet command, the generation of a sporadic report using its identifier.	

	Title:	**Event reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21

NFEESIM-UR-470	At startup, the SimuCam MEB shall enable or disable the generation of the sporadic reports according to a default configuration stored in its non-volatile memory.

	Title:	**Event reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-724	The SimuCam MEB shall allow the user to modify the default event generation status used at startup.	

4.19.3 Log reporting

	Title:	**Log reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-726	The SimuCam MEB shall record in a log all the progress / failure events which are enabled.	

	Title:	**Log reporting**
	Type:	Functional
	Verif.:	Test
	LastEdit:	2019-11-21
NFEESIM-UR-727	The SimuCam MEB shall allow the user to download, via an Ethernet command, the progress / failure event logs.	

4.20 Quality requirement

	Title:	**Quality requirement**
	Type:	Quality
	Verif.:	Test
	LastEdit:	2019-11-05
NFEESIM-UR-985	Each SimuCam shall be uniquely identified.	

	Title:	**Quality requirement**
	Type:	Quality
	Verif.:	Test
	LastEdit:	2019-11-05

NFEESIM-UR-986	A naming convention shall enable to uniquely identify SimuCam embedded hardware and software version

5. Deliveries and schedule

5.1 SimuCam MEB firmware needed dates

Four main releases of the SimuCam MEB firmware are foreseen. The table below gives for each SimuCam MEB firmware release: the needed date, an overview of the main expected features and the use cases (purpose of the release):

SimuCam MEB firmware releases	Needed dates	Features	Use cases
R0	April 2019	- Fullimage mode with patterns (no support for SSD) - 4 N-FEE simulations supported - Periodic status reporting - Ethernet TC/TM partially implemented (for commanding the functions implemented)	- N-DPU ASW V0.3 development (fullimage acquisition) - MEU EM tests (including 2 x N-DPU)
R1	October 2019	- Fullimage mode (patterns and SSD fullimages) - SSD management for fullimage only - 6 N-FEE simulation supported - On mode - Exposure duration management - Digitize flag - Non-volatile memory management - Time management - RMAP logging - CCD order management including constant CCD readout - Immediate on mode command	- N-DPU ASW V0.4 development (fullimage acquisition) - MEU EM tests
R1+	December 2019	- Error injection (Partial) - Fullimage error injection - External Sync - Synchronization Signal - NFEE Timing Configuration - N-FEE configuration simulation - SSD fullimage storage management	- N-DPU ASW V0.4 validation - MEU EM tests
R2	January 2020	- N-FEE Fullimage mode (patterns and SSD fullimages) - N-FEE Windowing mode (patterns and SSD fullimages, no support of SSD window stacks)	- N-DPU ASW V0.8 development (window acquisition) - DPS EM tests

		- N-FEE window list management: partially implemented (static LUT pre-loaded in the SDD) - N-FEE packet order list management: partially implemented (static list pre-loaded in the SDD) - N-FEE timing configuration - N-FEE configuration simulation - RMAP register simulation - Digitalization area management (vstart / vend registers) (Partial readout).	
R2+	March 2020	- Same as R2 but with full N-FEE window list management (loading of the window position list by RMAP) and full N-FEE packet order list management (loading of the packet order list by RMAP) - RMAP configuration areas - Status reporting (NFEE Simulation Entity) - N-FEE trap pumping mode management	- N-DPU ASW V0.8 development (window acquisition) - DPS EM tests
R3	May 2020	- N-FEE windowing mode management (Full) - N-FEE windowing / SSD window mode - RMAP Dump - SSD window stack storage management	- N-DPU ASW V0.8 development (window acquisition) - DPS EM tests
R4	June 2020	- Synchronization signal provided on the SpaceWire link (Req 1125/1126) - RTC - Status reporting - Event reporting - Log reporting - Mode transition implementation - Immediate ON - Error injection management (Full) - SpaceWire errors - RMAP errors - Time-code errors - Window transmission errors - Image and window content error - Data packet header error	- N-DPU ASW V0.8 validation - N-DPU ASW V1.x development and validation - DPS EQM tests - DPS FM tests

Table 1 - SimuCam firmware delivery schedule

An overview of the functional scope of each SimuCam MEB firmware release is given in the chapter 6 'Functional scope versus release'.

The detailed requirement allocation against the releases can be found in the chapter 7.

5.2 SimuCam MEB hardware needed dates

LESIA needs 5 SimuCam MEB units. Two of them will be delivered to IAA as part of the N-DPU ASW GSE (for testing the MEU with the flight S/W). The table below gives the needed dates.

SimuCam MEB unit	Needed dates	Comment
SimuCam #1	April 2019	
SimuCam #2	April 2019	Will be delivered to IAA as part of the N-DPU ASW GSE
SimuCam #3	December 2019	
SimuCam #4	December 2019	
SimuCam #5	December 2019	Will be delivered to IAA as part of the N-DPU ASW GSE

Table 2 - SimuCam unit delivery schedule

6. Functional scope versus release

Function	R0	R1	R2	R2+	R3	R4
SimuCam general requirements	X	X	X	X	X	X
SimuCam SSW requirements	X	X	X	X	X	X
SimuCam MEB general requirements	X	X	X	X	X	X
MEB communication interface	X	X	X	X	X	X
MEB Modes	X	X	X	X	X	X
N-FEE simulation entities	X	X	X	X	X	X
Reset and startup	X	X	X	X	X	X
Non-volatile memory		X	X	X	X	X
Time management						X
Endianness	X	X	X	X	X	X
MEB qualification					X	X
N-FEE / N-DPU interface	X	X	X	X	X	X
N-FEE / N-DPU SpaceWire physical interface	X	X	X	X	X	X
N-FEE / N-DPU communication protocol	X	X	X	X	X	X
SpaceWire / RMAP interface configuration	X	X	X	X	X	X
RMAP protocol management	X	X	X	X	X	X
RMAP configuration areas	X	X	X	X	X	X
RMAP echoing mode			X	X	X	X
Deleted		X	X	X	X	X
Timing and synchronization management		X	X	X	X	X
N-FEE timing configuration		X	X	X	X	X
Synchronization signal	X	X	X	X	X	X
Time-code management	X	X	X	X	X	X
Synchronization reset		X	X	X	X	X
SSD management		X	X	X	X	X
SSD storage unit		X	X	X	X	X
Fullimage storage management		X	X	X	X	X
Window stack storage management					X	X

Function	R0	R1	R2	R2+	R3	R4
N-FEE Digitalization simulation		X	X	X	X	X
Digitalization area definition		X	X	X	X	X
Read-out order		X	X	X	X	X
Warm-up simulation		X	X	X	X	X
N-FEE to N-DPU data packet transmission		X	X	X	X	X
N-FEE on mode simulation		X	X	X	X	X
N-FEE standby mode simulation	X	X	X	X	X	X
N-FEE fullimage mode simulation	X	X	X	X	X	X
N-FEE fullimage mode management	X	X	X	X	X	X
N-FEE fullimage / pattern mode	X	X	X	X	X	X
N-FEE fullimage / SSD image mode		X	X	X	X	X
N-FEE windowing mode simulation			X	X	X	X
N-FEE windowing mode management			X	X	X	X
Window list management			X	X	X	X
Packet order list management			X	X	X	X
N-FEE windowing / pattern mode			X	X	X	X
N-FEE windowing / SSD image mode			X	X	X	X
N-FEE windowing / SSD window mode					X	X
N-FEE Trap pumping mode simulation			X	X	X	X
N-FEE Immediate ON command simulation		X	X	X	X	X
N-FEE housekeeping simulation	X	X	X	X	X	X
N-FEE configuration simulation	X	X	X	X	X	X
RMAP register simulation		X	X	X	X	X
Error injection		X	X	X	X	X
Error injection management		X	X	X	X	X
SpaceWire errors						X
RMAP errors						X
Time-code errors						X
Fullimage transmission errors		X	X	X	X	X
Window transmission errors						X
Image and Window content errors						X

Function	R0	R1	R2	R2+	R3	R4
Data packet header error					X	X
Status / event / log reporting			X	X	X	X
Status reporting			X	X	X	X
Event reporting						X
Log reporting			X	X	X	X
Quality requirement		X	X	X	X	X

7. Detailed requirement allocation

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-403	SimuCam general requirements	X	X	X	X	X	X	X
NFEESIM-UR-407	SimuCam SSW requirements	X	X	X	X	X	X	X
NFEESIM-UR-558	SimuCam MEB communication interface	X	X	X	X	X	X	X
NFEESIM-UR-1164	SimuCam MEB communication interface						X	X
NFEESIM-UR-559	SimuCam MEB communication interface	X	X	X	X	X	X	X
NFEESIM-UR-690	SimuCam MEB communication interface	X	X	X	X	X	X	X
NFEESIM-UR-531	SimuCam MEB modes	X	X	X	X	X	X	X
NFEESIM-UR-537	SimuCam MEB modes	X	X	X	X	X	X	X
NFEESIM-UR-535	SimuCam MEB modes	X	X	X	X	X	X	X
NFEESIM-UR-536	SimuCam MEB modes	X	X	X	X	X	X	X
NFEESIM-UR-543	SimuCam MEB modes	X	X	X	X	X	X	X
NFEESIM-UR-405	N-FEE simulation entity		X	X	X	X	X	X
NFEESIM-UR-526	N-FEE simulation entity	X	X	X	X	X	X	X
NFEESIM-UR-672	N-FEE simulation entity	X	X	X	X	X	X	X
NFEESIM-UR-534	N-FEE simulation entity	X	X	X	X	X	X	X
NFEESIM-UR-549	N-FEE simulation entity	X	X	X	X	X	X	X
NFEESIM-UR-529	N-FEE simulation entity	X	X	X	X	X	X	X
NFEESIM-UR-530	N-FEE simulation entity	X	X	X	X	X	X	X
NFEESIM-UR-532	N-FEE simulation entity	X	X	X	X	X	X	X
NFEESIM-UR-567	SimuCam MEB reset and startup	X	X	X	X	X	X	X
NFEESIM-UR-562	SimuCam MEB reset and startup	X	X	X	X	X	X	X
NFEESIM-UR-568	SimuCam MEB reset and startup							X
NFEESIM-UR-496	Non-volatile memory							X
NFEESIM-UR-591	Time management							X
NFEESIM-UR-660	Time management							X
NFEESIM-UR-695	Time management							X
NFEESIM-UR-696	Time management							X
NFEESIM-UR-693	Time management							X
NFEESIM-UR-658	Endianness	X	X	X	X	X	X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-708	Endianness	X	X	X	X	X	X	X
NFEESIM-UR-858	Coordinate system	X	X	X	X	X	X	X
NFEESIM-UR-822	SimuCam MEB tests and qualification						X	X
NFEESIM-UR-823	SimuCam MEB tests and qualification						X	X
NFEESIM-UR-826	N-FEE / N-DPU SpaceWire physical interface	X	X	X	X	X	X	X
NFEESIM-UR-827	N-FEE / N-DPU SpaceWire physical interface	X	X	X	X	X	X	X
NFEESIM-UR-448	N-FEE / N-DPU communication protocol	X	X	X	X	X	X	X
NFEESIM-UR-423	SpaceWire / RMAP interface configuration	X	X	X	X	X	X	X
NFEESIM-UR-475	SpaceWire / RMAP interface configuration							X
NFEESIM-UR-476	SpaceWire / RMAP interface configuration							X
NFEESIM-UR-477	SpaceWire / RMAP interface configuration							X
NFEESIM-UR-478	SpaceWire / RMAP interface configuration							X
NFEESIM-UR-474	SpaceWire / RMAP interface configuration		X	X	X	X	X	X
NFEESIM-UR-473	SpaceWire / RMAP interface configuration	X	X	X	X	X	X	X
NFEESIM-UR-638	RMAP protocol management	X	X	X	X	X	X	X
NFEESIM-UR-640	RMAP protocol management	X	X	X	X	X	X	X
NFEESIM-UR-641	RMAP protocol management	X	X	X	X	X	X	X
NFEESIM-UR-642	RMAP protocol management	X	X	X	X	X	X	X
NFEESIM-UR-583	RMAP configuration areas	X	X	X	X	X	X	X
NFEESIM-UR-584	RMAP configuration areas	X	X	X	X	X	X	X
NFEESIM-UR-585	RMAP configuration areas	X	X	X	X	X	X	X
NFEESIM-UR-699	RMAP configuration areas					X	X	X
NFEESIM-UR-588	RMAP echoing mode					X	X	X
NFEESIM-UR-589	RMAP echoing mode					X	X	X
NFEESIM-UR-590	RMAP echoing mode					X	X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-1166	RMAP echoing mode					X	X	X
NFEESIM-UR-1167	RMAP echoing mode					X	X	X
NFEESIM-UR-702	Deleted							
NFEESIM-UR-703	Deleted							
NFEESIM-UR-704	Deleted							
NFEESIM-UR-706	Deleted							
NFEESIM-UR-447	N-FEE timing configuration		X	X	X	X	X	X
NFEESIM-UR-898	N-FEE timing configuration	X	X	X	X	X	X	X
NFEESIM-UR-899	N-FEE timing configuration							X
NFEESIM-UR-900	N-FEE timing configuration							X
NFEESIM-UR-829	N-FEE timing configuration		X	X	X	X	X	X
NFEESIM-UR-485	Synchronization signal	X	X	X	X	X	X	X
NFEESIM-UR-636	Synchronization signal		X	X	X	X	X	X
NFEESIM-UR-631	Synchronization signal		X	X	X	X	X	X
NFEESIM-UR-632	Synchronization signal		X	X	X	X	X	X
NFEESIM-UR-633	Synchronization signal		X	X	X	X	X	X
NFEESIM-UR-634	Synchronization signal							X
NFEESIM-UR-635	Synchronization signal							X
NFEESIM-UR-630	Synchronization signal	X	X	X	X	X	X	X
NFEESIM-UR-835	Synchronization signal	X	X	X	X	X	X	X
NFEESIM-UR-629	Synchronization signal	X	X	X	X	X	X	X
NFEESIM-UR-910	Synchronization signal	X	X	X	X	X	X	X
NFEESIM-UR-1064	Synchronization signal		X	X	X	X	X	X
NFEESIM-UR-1066	Synchronization signal							X
NFEESIM-UR-1067	Synchronization signal							X
NFEESIM-UR-1065	Synchronization signal		X	X	X	X	X	X
NFEESIM-UR-1125	Synchronization signal				X	X	X	X
NFEESIM-UR-1126	Synchronization signal				X	X	X	X
NFEESIM-UR-486	Time-code management	X	X	X	X	X	X	X
NFEESIM-UR-488	Time-code management	X	X	X	X	X	X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-489	Time-code management	X	X	X	X	X	X	X
NFEESIM-UR-838	Deleted							
NFEESIM-UR-709	Synchronization reset		X	X	X	X	X	X
NFEESIM-UR-728	Deleted							
NFEESIM-UR-729	Deleted							
NFEESIM-UR-710	Synchronization reset		X	X	X	X	X	X
NFEESIM-UR-412	SSD storage unit		X	X	X	X	X	X
NFEESIM-UR-430	SSD storage unit		X	X	X	X	X	X
NFEESIM-UR-597	SSD storage unit		X	X	X	X	X	X
NFEESIM-UR-431	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-714	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-439	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-676	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-432	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-454	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-455	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-456	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-548	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-435	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-433	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-600	SSD fullimage storage management		X	X	X	X	X	X
NFEESIM-UR-601	SSD window stack storage management						X	X
NFEESIM-UR-678	SSD window stack storage management						X	X
NFEESIM-UR-717	SSD window stack storage management						X	X
NFEESIM-UR-602	SSD window stack storage management						X	X
NFEESIM-UR-603	SSD window stack storage management						X	X
NFEESIM-UR-604	SSD window stack storage management						X	X
NFEESIM-UR-605	SSD window stack storage management						X	X
NFEESIM-UR-606	SSD window stack storage management						X	X
NFEESIM-UR-607	SSD window stack storage management						X	X
NFEESIM-UR-608	SSD window stack storage management						X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-609	SSD window stack storage management						X	X
NFEESIM-UR-610	SSD window stack storage management						X	X
NFEESIM-UR-1005	Digitalization area definition				X	X	X	X
NFEESIM-UR-1006	Digitalization area definition				X	X	X	X
NFEESIM-UR-1008	Digitalization area definition				X	X	X	X
NFEESIM-UR-839	Deleted							
NFEESIM-UR-1050	Read-out order		X	X	X	X	X	X
NFEESIM-UR-1051	Read-out order		X	X	X	X	X	X
NFEESIM-UR-1052	Read-out order		X	X	X	X	X	X
NFEESIM-UR-842	Read-out order				X	X	X	X
NFEESIM-UR-1055	Warm-up simulation		X	X	X	X	X	X
NFEESIM-UR-1056	Warm-up simulation		X	X	X	X	X	X
NFEESIM-UR-1057	Warm-up simulation		X	X	X	X	X	X
NFEESIM-UR-1083	N-FEE to N-DPU data packet transmission		X	X	X	X	X	X
NFEESIM-UR-1085	N-FEE to N-DPU data packet transmission		X	X	X	X	X	X
NFEESIM-UR-1089	N-FEE to N-DPU data packet transmission		X	X	X	X	X	X
NFEESIM-UR-928	N-FEE on mode simulation		X	X	X	X	X	X
NFEESIM-UR-929	N-FEE on mode simulation		X	X	X	X	X	X
NFEESIM-UR-930	N-FEE on mode simulation		X	X	X	X	X	X
NFEESIM-UR-931	N-FEE on mode simulation				X	X	X	X
NFEESIM-UR-938	N-FEE on mode simulation				X	X	X	X
NFEESIM-UR-937	N-FEE on mode simulation		X	X	X	X	X	X
NFEESIM-UR-936	N-FEE on mode simulation		X	X	X	X	X	X
NFEESIM-UR-935	N-FEE on mode simulation					X	X	X
NFEESIM-UR-934	N-FEE on mode simulation					X	X	X
NFEESIM-UR-933	N-FEE on mode simulation					X	X	X
NFEESIM-UR-932	N-FEE on mode simulation					X	X	X
NFEESIM-UR-939	N-FEE on mode simulation		X	X	X	X	X	X
NFEESIM-UR-451	N-FEE standby mode simulation	X	X	X	X	X	X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-681	N-FEE standby mode simulation	X	X	X	X	X	X	X
NFEESIM-UR-830	N-FEE standby mode simulation	X	X	X	X	X	X	X
NFEESIM-UR-550	N-FEE standby mode simulation	X	X	X	X	X	X	X
NFEESIM-UR-679	N-FEE standby mode simulation				X	X	X	X
NFEESIM-UR-925	N-FEE standby mode simulation		X	X	X	X	X	X
NFEESIM-UR-551	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-945	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-440	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-946	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-655	N-FEE fullimage mode management		X	X	X	X	X	X
NFEESIM-UR-444	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-449	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-902	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-903	Deleted							
NFEESIM-UR-650	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-674	N-FEE fullimage mode management	X	X	X	X	X	X	X
NFEESIM-UR-511	N-FEE fullimage mode management		X	X	X	X	X	X
NFEESIM-UR-512	N-FEE fullimage mode management		X	X	X	X	X	X
NFEESIM-UR-843	N-FEE fullimage mode management		X	X	X	X	X	X
NFEESIM-UR-441	N-FEE fullimage / pattern mode	X	X	X	X	X	X	X
NFEESIM-UR-443	N-FEE fullimage / pattern mode	X	X	X	X	X	X	X
NFEESIM-UR-501	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-553	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-457	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-461	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-462	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-463	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-464	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-458	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-460	N-FEE fullimage / SSD image mode		X	X	X	X	X	X
NFEESIM-UR-554	N-FEE windowing mode management				X	X	X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-947	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-555	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-948	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-656	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-673	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-833	Deleted							
NFEESIM-UR-831	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-675	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-788	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-516	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-517	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-513	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-520	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-844	N-FEE windowing mode management				X	X	X	X
NFEESIM-UR-518	Window list management				X	X	X	X
NFEESIM-UR-791	Window list management				X	X	X	X
NFEESIM-UR-574	Window list management						X	X
NFEESIM-UR-569	Window list management				X	X	X	X
NFEESIM-UR-576	Window list management				X	X	X	X
NFEESIM-UR-575	Window list management				X	X	X	X
NFEESIM-UR-1101	Packet order list management				X	X	X	X
NFEESIM-UR-1105	Packet order list management				X	X	X	X
NFEESIM-UR-1135	Packet order list management					X	X	X
NFEESIM-UR-1136	Packet order list management				X	X	X	X
NFEESIM-UR-1137	Packet order list management				X	X	X	X
NFEESIM-UR-1138	Packet order list management				X	X	X	X
NFEESIM-UR-1106	Packet order list management				X	X	X	X
NFEESIM-UR-1107	Packet order list management				X	X	X	X
NFEESIM-UR-1109	Packet order list management				X	X	X	X
NFEESIM-UR-1110	Packet order list management				X	X	X	X
NFEESIM-UR-1112	Packet order list management				X	X	X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-1113	Packet order list management				X	X	X	X
NFEESIM-UR-1115	Packet order list management				X	X	X	X
NFEESIM-UR-1103	Packet order list management				X	X	X	X
NFEESIM-UR-1116	Packet order list management				X	X	X	X
NFEESIM-UR-556	N-FEE windowing / pattern mode				X	X	X	X
NFEESIM-UR-515	N-FEE windowing / pattern mode				X	X	X	X
NFEESIM-UR-521	N-FEE windowing / SSD image mode				X	X	X	X
NFEESIM-UR-522	N-FEE windowing / SSD image mode				X	X	X	X
NFEESIM-UR-581	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-614	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-615	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-617	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-618	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-619	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-625	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-620	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-621	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-622	N-FEE windowing / SSD window mode						X	X
NFEESIM-UR-951	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-953	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-954	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-962	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-971	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-972	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-973	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-974	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-975	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-976	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-977	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-978	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-982	N-FEE trap pumping mode management					X	X	X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-979	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-983	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-1094	N-FEE trap pumping mode management					X	X	X
NFEESIM-UR-943	Immediate ON command simulation		X	X	X	X	X	X
NFEESIM-UR-944	Immediate ON command simulation							X
NFEESIM-UR-579	N-FEE housekeeping simulation		X	X	X	X	X	X
NFEESIM-UR-644	N-FEE housekeeping simulation	X	X	X	X	X	X	X
NFEESIM-UR-643	N-FEE housekeeping simulation	X	X	X	X	X	X	X
NFEESIM-UR-645	N-FEE housekeeping simulation		X	X	X	X	X	X
NFEESIM-UR-647	N-FEE housekeeping simulation							X
NFEESIM-UR-648	N-FEE housekeeping simulation							X
NFEESIM-UR-907	N-FEE configuration simulation	X	X	X	X	X	X	X
NFEESIM-UR-908	N-FEE configuration simulation	X	X	X	X	X	X	X
NFEESIM-UR-909	N-FEE configuration simulation				X	X	X	X
NFEESIM-UR-1070	RMAP register simulation		X	X	X	X	X	X
NFEESIM-UR-1073	RMAP register simulation		X	X	X	X	X	X
NFEESIM-UR-1074	RMAP register simulation		X	X	X	X	X	X
NFEESIM-UR-1076	RMAP register simulation		X	X	X	X	X	X
NFEESIM-UR-1078	RMAP register simulation		X	X	X	X	X	X
NFEESIM-UR-1079	RMAP register simulation		X	X	X	X	X	X
NFEESIM-UR-1080	RMAP register simulation		X	X	X	X	X	X
NFEESIM-UR-659	Error injection management		X	X	X	X	X	X
NFEESIM-UR-419	Error injection management		X	X	X	X	X	X
NFEESIM-UR-723	Error injection management		X	X	X	X	X	X
NFEESIM-UR-418	SpaceWire errors							X
NFEESIM-UR-420	RMAP errors							X
NFEESIM-UR-481	Time-code errors							X
NFEESIM-UR-421	Fullimage transmission errors		X	X	X	X	X	X
NFEESIM-UR-480	Window transmission errors							X
NFEESIM-UR-571	Window transmission errors							X
NFEESIM-UR-989	Image and window content error							X

Requirement ID	Requirement Title	R0	R1	R1+	R2	R2+	R3	R4
NFEESIM-UR-990	Image and window content error							X
NFEESIM-UR-992	Image and window content error							X
NFEESIM-UR-993	Image and window content error							X
NFEESIM-UR-994	Image and window content error							X
NFEESIM-UR-999	Image and window content error							X
NFEESIM-UR-997	Image and window content error							X
NFEESIM-UR-1002	Image and window content error							X
NFEESIM-UR-1001	Image and window content error							X
NFEESIM-UR-996	Image and window content error							X
NFEESIM-UR-1117	Data packet header error							X
NFEESIM-UR-1118	Data packet header error							X
NFEESIM-UR-1119	Data packet header error							X
NFEESIM-UR-1120	Data packet header error							X
NFEESIM-UR-1121	Data packet header error							X
NFEESIM-UR-1122	Data packet header error							X
NFEESIM-UR-425	Status reporting					X	X	X
NFEESIM-UR-546	Status reporting					X	X	X
NFEESIM-UR-547	Status reporting					X	X	X
NFEESIM-UR-427	Status reporting							X
NFEESIM-UR-466	Status reporting							X
NFEESIM-UR-465	Status reporting							X
NFEESIM-UR-467	Event reporting							X
NFEESIM-UR-468	Event reporting							X
NFEESIM-UR-469	Event reporting							X
NFEESIM-UR-470	Event reporting							X
NFEESIM-UR-724	Event reporting							X
NFEESIM-UR-726	Log reporting							X
NFEESIM-UR-727	Log reporting							X
NFEESIM-UR-985	Quality requirement		X	X	X	X	X	X
NFEESIM-UR-986	Quality requirement		X	X	X	X	X	X

8. Distribution list

Name	Email	Organization	Distribution
Rainer Berlin	Rainer.Berlin@dlr.de	DLR, Berlin	X
Rosario Consentino	consentino@tng.iac.es	INAF-FGG	X
Florent Ducellier	florent.ducellier@obspm.fr	LESIA	X
Mauro Focardi	mauro@arcetri.astro.it	INAF, Florence	X
Rodrigo França	rodmarfran@gmail.com	IMT, São Paulo	X
Nicolas Gauthier	nicolas.gauthier@obspm.fr	LESIA, Meudon	X
Loïc Gueguen	loic.gueguen@obspm.fr	LESIA, Meudon	X
Lee-Roy Malac-Allain	Lee-Roy.Malac-Allain@obspm.fr	LESIA, Meudon	X
Vanderlei Parro	vparro@ieee.org	IMT, São Paulo	X
Gisbert Peter	gisbert.peter@dlr.de	DLR, Berlin	X
Stefano Pezzuto	pezzuto@iaps.inaf.it	INAF, Rome	X
Philippe Plasson	philippe.plasson@obspm.fr	LESIA, Meudon	X
William Recart	william.recart@nexeya.com	Nexeya, Toulouse	X
Julio Rodríguez	julio@iaa.es	IAA-CSIC, Granada	X
Tiago Sanches da Silva	tiago.eem@gmail.com	IMT, São Paulo	X
Rosario Sanz	sanz@iaa.es	IAA-CSIC, Granada	X
Bernd Ulmer	bernd.ulmer@ib-ulmer.de	DLR/IBU, Berlin	X
Karsten Westerdorff	Karsten.Westerdorff@dlr.de	DLR, Berlin	X
Claas Ziemke	Claas.Ziemke@dlr.de	DLR, Berlin	X