OPS-SAT in orbit - a technical rundown of this open experimentation platform

Dominik Marszk[1], David Evans[2], Maximilian Henkel[3], Otto Koudelka[3], Manuel Kubicka[3], Mario Merri[2], Tom Mladenov[2], Kevin Otto[2], Mehran Sarkarati[2]

[1] IMS Space Consultancy GmbH
Mail: dominik.marszk@esa.int

[2] European Space Agency
European Space Operations Centre
Mail: name.lastname@esa.int

[3] Graz University of Technology
Mail: koudelka@tugraz.at
henkel@tugraz.at
manuel.kubicka@tugraz.at

Abstract: OPS-SAT is a flying hardware and software laboratory in a form of a triple unit CubeSat built by the Graz University of Technology (TU Graz) for the European Space Agency (ESA) and launched in December 2019.
To this date, over 100 entities from 18 countries have registered proposals to fly software experiments on OPS-SAT. In fact, ESA allows several different configurations for both the ground and space segments.
In this paper, we introduce the mission profile, hardware features of our flight and ground segments, and the end-to-end software stack provided to the experimenters, aiming to minimise the time and effort needed to develop a working solution.

1. INTRODUCTION

1.1. Mission goal

OPS-SAT is a flying hardware and software laboratory in a form of a triple unit CubeSat, built by the Technical University of Graz for the European Space Agency and launched in December 2019. The mission's primary goal is to break the "has never flown, therefore it will never fly" cycle, preventing many ideas from ever reaching the real world environment, as they are deemed too risky, or too expensive to implement and test. In order to achieve this, OPS-SAT provides a set of generic capabilities and subsystems that are commonly found in space missions, large and small. Not only are these capabilities made available to the experimenters, but also the level, easiness, and lack of restrictions to make use of OPS-SAT's capabilities are brought to the next level through provision of an open-source space and ground software framework. [1]

1.2. Main mission features

To increase the spectrum of possible applications, the mission includes multiple subsystems representative of a typical spacecraft, plus some novel ones. These subsystems consist of a programmable Satellite Experimental Processing Platform (SEPP – based on a dual core ARM CPU + FPGA), coarse and fine Attitude Determination and Control System (ADCS), GNSS receiver, HD camera, UHF software-defined radio receiver, optical receiver, S-band radio transceiver, X-band transmitter, and a retroreflector. [1] [2]

To date, over 100 entities from 18 countries have registered software experiment proposals to fly on OPS-SAT. These include space agencies, multi-national corporations, universities, and one-man shows. To accommodate such a variety of experimenters with very diverse backgrounds, resources, and domains of interest, ESOC offers the experimenters multiple access mechanisms. This ranges from remote experiment execution with basic real-time telemetry and telecommand link forwarding, to higher-level application interfaces, such as file uplink and downlink, or space-to-ground data mirroring. On top of that, a lightweight web-based monitoring and control system is offered in combination with a user-friendly on-board API wrapping the lower level interfaces.

Furthermore, to ease the process of testing and validation required to qualify the experiment for uplink, we provide scheduled remote access slots to the satellite's engineering model on the ground. Experimenters can also use those access slots to test software, simulate their operations and get acquainted with the operational setting.

2. OPS-SAT SPACE SEGMENT SYSTEMS AND INTERFACES

2.1. Telecommand & Telemetry (TM/TC) handling

The main TM/TC subsystems are the X-Band transmitter (Syrlinks EWC27) and the S-Band transceiver (Syrlinks EWC31). Frame synchronisation and coding is done by "CCSDS Engine" (CCSDSE) in adherence with CCSDS TM Space Data Link Protocol and CCSDS TC Synchronisation and Channel Coding standards.

The S-Band transceiver maximum, and default downlink speed is 1 Mbps while the uplink speed is fixed to 256 Kbps. The high uplink speed (four times faster than any ESA spacecraft) is needed to facilitate regular changes of on-board experimental software and FPGA configurations.

The X-Band transmitter maximum downlink speed is 50 Mbps. The CCSDS Engine's maximum downlink rate goes up to 24 Mbps and higher speeds can be supported with a "bypass mode". In this mode, the CCSDS engine is bypassed allowing to connect either TM, TC, or both channels of the currently selected radio directly to the SEPP's FPGA port. This gives the ability to experiment with custom implementations of higher bandwidth or different framing protocols that are not supported by the platform out of the box. For example, prototyping and flight demonstration of the new CCSDS Unified Space Data Link Protocol could be performed using a custom FPGA core and a modified ground station equipment.

The connectivity between SEPP and CCSDSE is realised via CAN-bus interface for up to 500 Kbps effective bandwidth, shared with the core satellite bus On-Board Computer (OBC, GomSpace Nanomind A3200). The CAN bus fragmentation protocol allows transmitting CCSDS Space Packets (SPPs) up to 256 bytes in length. If longer SPPs or higher bandwidth are required, a dedicated SpaceWire interface between the SEPP and the CCSDSE can be used.

A backup TM/TC subsystem is GomSpace NanoCom AX-100, used to control core bus components (OBC, EPS) via UHF band.

The overall composition and routing of the TM/TC subsystems are visible on the Figure 1.

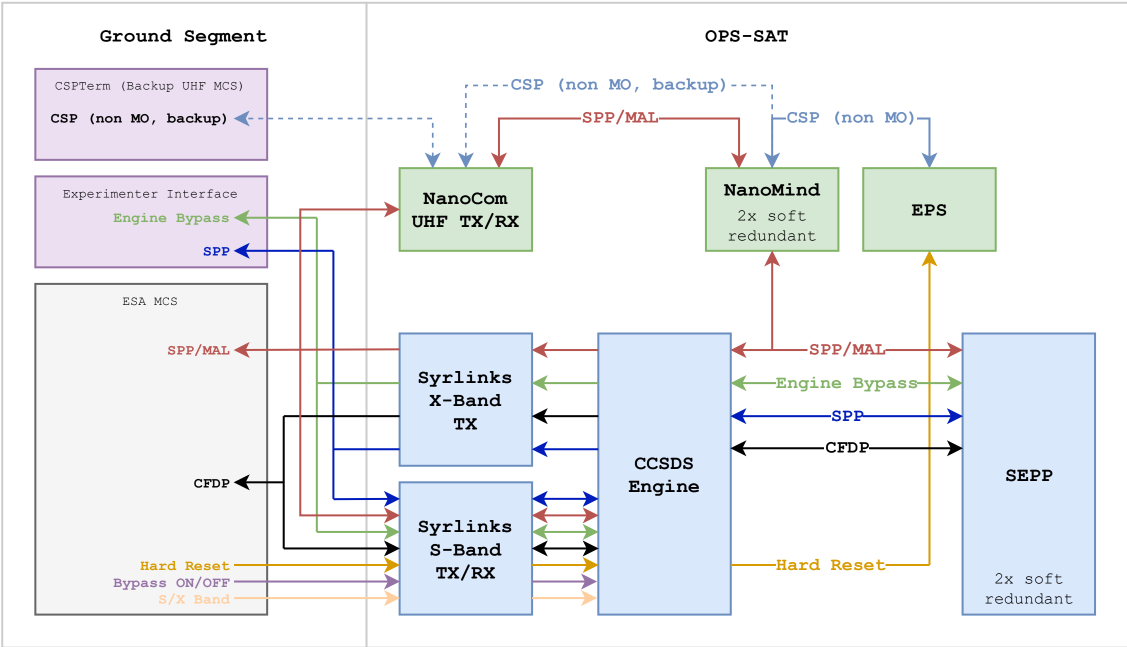

Figure 1 – OPS-SAT TM/TC Synopsis

2.2. SEPP - Satellite Experimental Processing Platform

SEPP is the core of the experimental capabilities provided by the platform. The board was developed by TU Graz. It is built around Altera Cyclone V SoC with 1 GB of ECC-enabled DDR3 memory dedicated to the CPU and 512 MB of DDR3 memory dedicated to the FPGA. This is the processor on to which the experimental software and firmware will be loaded and installed. From here the experiment can interact with the units described later.

The overall system diagram showing both OPS-SAT core and payload buses is shown on the Figure 2. SEPP is visible in the middle-right portion of the picture, providing access to the most of the buses available on-board.

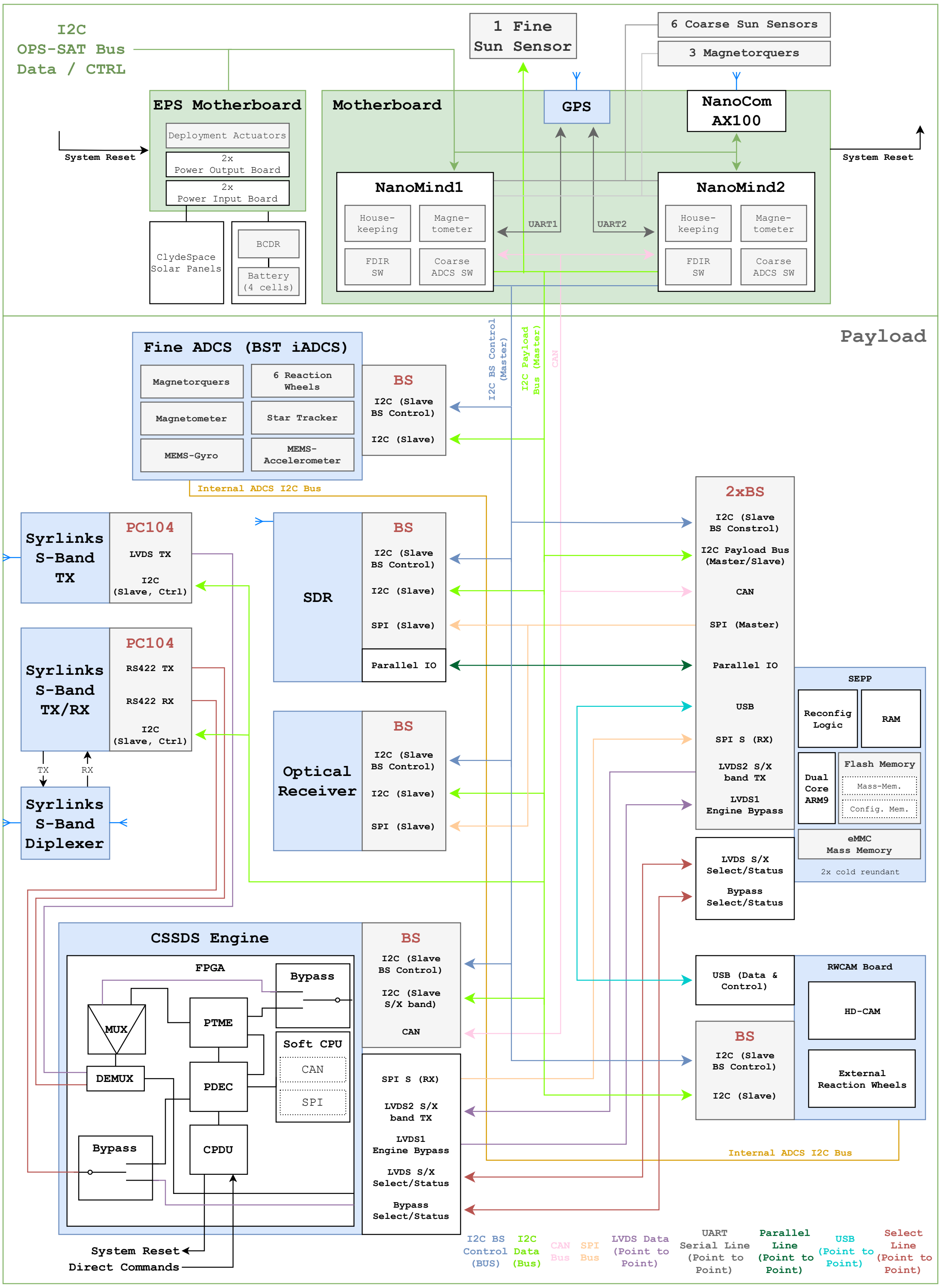

Figure 2 – OPS-SAT System Architecture

2.2.1. Provided software

Developing, testing, deploying, and operating a spacecraft On-Board Software (OBSW) is a difficult task due to the extra requirements derived from the uniqueness of the environment and the hardware design, as well as desired reliability. We aim to minimize any additional, redundant effort spent by the OPS-SAT users in that area. We are going to achieve that through providing the experimenter community with a comprehensive, fully functional reference firmware and software stack running on the SEPP, consisting of:

- a reference Altera Cyclone V SoC design;
- a baseline configuration and FPGA IP cores handling all of the SEPP interfaces like CAN, I2C, SPI, SpaceWire, radio transceivers bypass;
- an embedded Linux system image, built on Angstrom + Yocto 2.4.4 platform;
- Python 3.5 installation;
- Java Runtime Environment 8 installation;
- protocol bridge applications allowing access the CCSDS-compliant data streams on CAN and SpaceWire buses through a TCP socket;
- a set of user space libraries, implementing drivers for all of the payloads connected to the SEPP; and
- NanoSat MO Framework (NMF) – a set of Java libraries implementing CCSDS Mission Operations (MO) Services standards in a form of abstract software components. It allows the users to quickly build experiments as portable applications, already including a simple, service-oriented Monitoring & Control interface; the apps are also executable in a desktop simulation environment. [3][4]

2.3. SDR – Software Defined Radio Receiver

The SEPP is stacked together with an SDR receiver board. The board is built around LMS6002D Analog Front-End IC, controlled via SPI and I^2C interfaces wired to the SEPP.

The receiver presents with the following RF characteristics:

- minimum input frequency: 300 MHz,
- maximum input frequency: 1.575 GHz (due to the low-pass filter),
- RF sensitivity: -98 dBm,
- RF sampling rate 1.5 - 40 MHz, and
- LNA gain for 430-440 MHz: 19.5 – 20 dB (including LPF).

The SDR is optimised to amplify and receive data in the 70 cm band and connected to a dipole UHF antenna. Antenna measurements show centre frequency Fc = 425 MHz, and 10 dB bandwidth BW_{10dB} = 75.9 MHz.

The unit is intended to provide a sort of spectrum analyser in the sky, allowing experiments to configure the SDR, make measurements and process them on the SEPP. From there the results can be stored and downloaded to the ground.

2.4. Camera

The camera used is Berlin Space Technologies IMS100, connected to the SEPP via an USB interface. The sensor has the following specification:

- RGGB Bayer color filter array pattern,
- maximum resolution 2048 x 1944 pixels,
- frame rate up to 5Hz at 2048 x 1944 pixels, and
- ground resolution ~60 m/pixel.

2.5. ADCS – Fine Attitude Determination and Control System

The ADCS used is Berlin Space Technologies iADCS-100, connected to the SEPP via an I^2C interface. This unit will allow experiments to rotate and point the spacecraft. It is also needed to perform ground station tracking when using the laser uplink and ranging, or X band downlink. Also direct access to the low level sensor measurements and actuator commands is provided for advanced experiments. The system has the following specifications:

- a star tracker (BST IMS-200),
- 6 reaction wheels,
- 3 magnetorquers shared with the OBC's Coarse ADCS, and
- a variety of operating modes including: Measurement, Detumbling, Sun Pointing, Target Pointing, Nadir Pointing.

2.6. GNSS Receiver

The receiver on-board is NovAtel OEM615, able to track GPS L1 signals. It is connected to the OBC via UART interface, which then exposes a platform-specific GPS MO Service to the SEPP.

2.7. Optical Receiver

Optical Receiver is based on an Arduino-compatible ATmega328P MCU connected to a Multi Pixel Photon Counting sensor module C13365SA via an analogue comparator. The board controlled via SPI and I^2C interfaces wired to the SEPP.

The MCU comes with a rudimentary firmware and any particular experiment is encouraged to implement their own firmware which can be flashed from SEPP using an AVR SPI download protocol.

TU Graz has access to a laser ground station which will be used to commission the unit.

2.8. Retroreflector

A retroreflector mounted on the bottom side of the spacecraft, next to the camera and the X-Band antenna, allows for satellite ranging experiments.

3. OPS-SAT GROUND SEGMENT SYSTEMS AND INTERFACES

The OPS-SAT ground segment was developed with two major drivers in mind. One was to reuse the ESA Mission Control System infrastructure (MICONYS [5]) – a software suite used by most ESA missions and other major European organisations so that any lessons learned from OPS-SAT could feed back into the main stream of ESA missions. This is not only the first real-life use of this software for a CubeSat mission, but also the first time that such system is used to monitor & control a space segment based on CCSDS MO Services instead of ECSS Packet Utilization Standard (PUS). Moreover, OPS-SAT is the first ESA mission adopting the CCSDS File Delivery Protocol (CFDP) to implement File-based-Operations and the corresponding implementation in MICONYS. Another driver was to meet the already discussed unique requirements of the mission. Keeping that in mind allowed us to create a hybrid ground data system, consisting of:

- a fully representative operational network, with the same MCS core (SCOS-2000) as every other ESA mission uses to the day, including bleeding edge features, like CCSDS File Delivery Protocol (CFDP) or support for CCSDS MO Services.
- a set of applications first time demonstrated in OPS-SAT, like Data Proxy, allowing to dynamically switch between different Ground-Space interfaces, NMF Ground MO Proxy, serving as a protocol-bridge and a ground data archive for NMF experiments, or Lightweight MCS, serving as a GUI for the experiment applications developed on top of the NMF.

3.1. SMILE Ground Infrastructure

The central element from where all OPS-SAT operations take place is the Special Mission Infrastructure Laboratory Environment (SMILE) located on the ESOC premises. The SMILE lab has direct access to the on-site 3.7 meter S- and X-band tracking antenna illustrated in Figure 3 as well as an upcoming UHF station for which preparations are underway. The baseband equipment for all antennas can be configured to be a mix of either standardized components such as CORTEX baseband equipment, but also SDR based solutions. This also allows OPS-SAT experimenters to not only experiment in space, but also perform experiments on the ground-system of OPS-SAT, making the mission a fully end-to-end experimenter platform.

Figure 3: SMILE 3.7 m S/X-band tracking dish radome

Also located in the SMILE environment is the Engineering Model (EM) of OPS-SAT which is an almost fully representative testbed for the real satellite. The EM is used daily for procedure validation, simulation and experimenter tests. A dedicated development network was setup in SMILE to allow procedure development and experimenter tests to be run in parallel with the operations.

3.2. MCS - Mission Control System

OPS-SAT ground segment is built around SCOS-2000 – a Mission Control System core used by most ESA missions and many major European organisations. This allowed us to build a fully-fledged, reliable system without needlessly spending effort on doing it from scratch. Instead, we focused on tailoring the existing software for the unique mission needs and goals, in particular:

- support for CCSDS MO Services MAL/SPP protocol,

- integration with CCSDS File Delivery Protocol,
- connectivity with non-SLE, UHF ground stations,
- support for POCKET+ telemetry compression protocol,
- exposing the live TM/TC link to auxiliary applications and the internet.

All of the above were achieved with minimum modifications to the SCOS core, but rather with configuration changes, and developing a set of small applications seamlessly integrating into the existing MCS infrastructure.

3.3. MATIS – Mission Operations

To facilitate the operations of OPS-SAT, the ESA Mission Automation System (MATIS) is used and serves as the centrepiece of our daily operations. MATIS executes procedures written in the ECSS Procedures Language for Users in Test and Operations (PLUTO). MATIS has various ways to interact with the spacecraft and ground segment to support the daily operations, including:

- opening and closing the links to the various ground stations,
- configuring the MCS for operations in the UHF band or S/X-band,
- releasing telecommands through the MCS,
- monitoring telemetry and sending reports and alerts to the Flight Control Team,
- direct and live access to the SEPP shell command-line,
- triggering CFDP file uplink and downlink transfers between the MCS and SEPP,
- low-level access to the satellite core bus via CSP-term,
- opening and closing the TM/TC links to the experimenters.

OPS-SAT is the first ESA mission to fully integrate MATIS in unmanned operations.

3.4. NMF - NanoSat MO Framework

NanoSat MO Framework, as mentioned before, comes in the form of open-source Java libraries and applications, allowing to quickly create applications interfacing with the OPS-SAT payloads and exposing a unified monitoring & control interface, compliant with the CCSDS Mission Operations Monitor & Control Services standard. [3][4][6]

Not only NMF provides means to quickly develop, test and deploy on-board applications, but also it comes with a set of ground components meaning to abstract away difficulties coming from operating a remote system with very sparse live visibility.

A major such component is Ground MO Proxy. Its function is to mirror the space application data model, configuration and state (persisted as COM – MO Common Object Model) in an opportunistic manner, and expose them further to other ground systems. This approach ensures that the need to worry about the specifics of operating a space mission is almost completely abstracted away from the experiment developer.

3.5. LWMCS - Light-Weight Mission Control System

The ultimate application provided to the OPS-SAT experimenters is called Light-Weight Mission Control System, built by reusing ESA Ground Operations System User Desktop (EUD) - a selected set of well-developed Graphical User Interfaces used to monitor and control SCOS based system. These GUIs were packaged into a web application, and tailored to support integration with the MO Services. Most importantly COM and MC. Summarizing, this allows us to host a web-based MCS, providing the OPS-SAT experimenters with the means to monitor and control their NMF-based applications.

4. DATA SYSTEMS SUMMARY

A high level diagram of the ground and space systems with focus on the experimenter interfaces can be found on the Figure 4. It shows clearly a different levels of access provided, which should satisfy the needs of any experimenters.
A diagram going into more details of the system, interfaces, and protocol used is shown on the Figure 5.

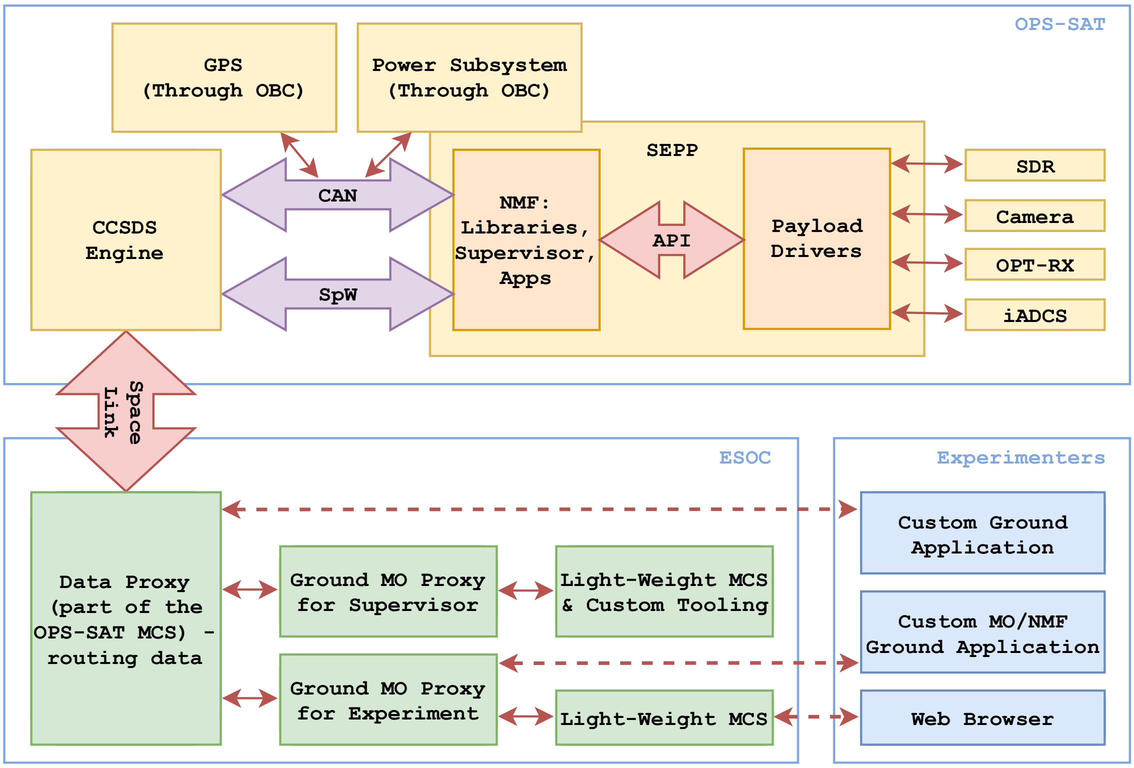

Figure 4 – High level OPS-SAT experimenter data system diagram

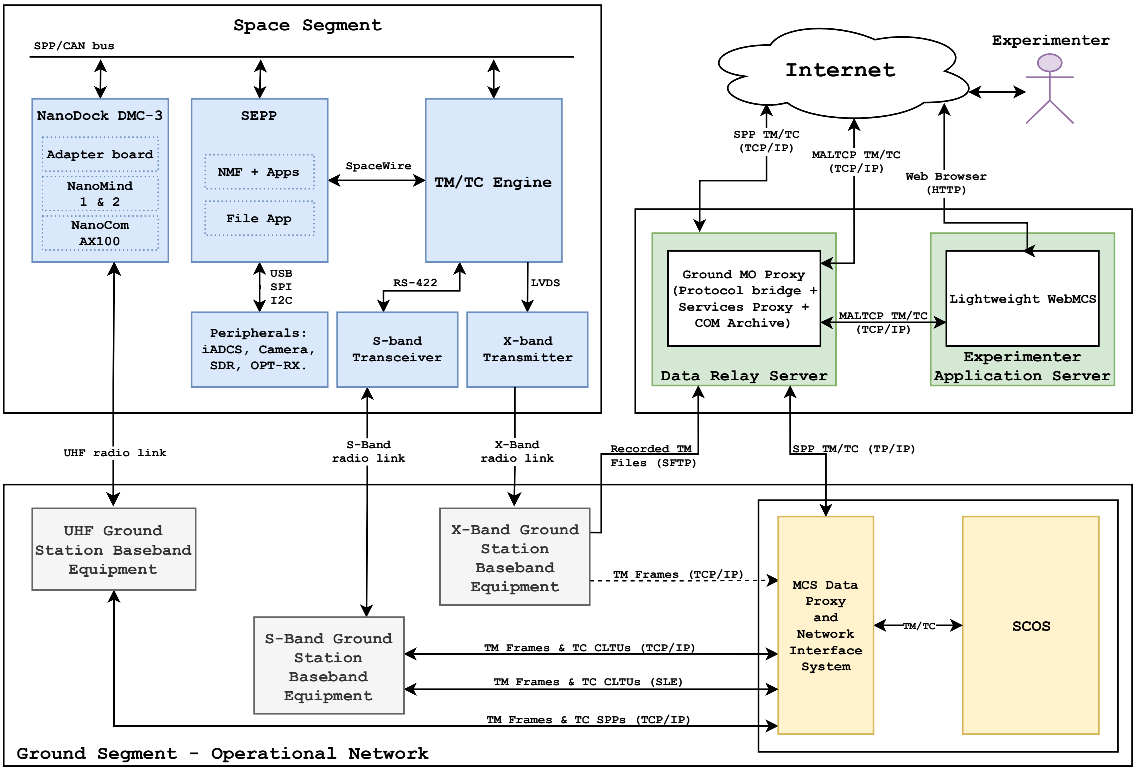

Figure 5 – OPS-SAT operational data system diagram with focus on the interfaces

5. CURRENT MISSION STATUS

The mission launched on 18th December 2019. Launch and Early Orbit Phase (LEOP) finished at the end of the year. It has been running successfully since then and extended twice. Current end of mission date is planned for November 2022.

Since launch, the satellite has hosted tens of experiments on-board and is still delivering valuable data and in-orbit-validation opportunities to its users.

6. REFERENCES

[1] D. Evans and A. Lange, OPS-SAT: Operational Concept for ESA'S First Mission Dedicated to Operational Technology, SpaceOps 2016 conference

[2] OPS-SAT – eoPortal Satellite Missions Directory: https://directory.eoportal.org/web/eoportal/satellite-missions/o/ops-sat (accessed January 2019)

[3] C. Coelho, O. Koudelka, M. Merri, NanoSat MO framework: When OBSW turns into apps, 2017 IEEE Aerospace Conference (2017)

[4] C. Coelho, A Software Framework for Nanosatellites based on CCSDS Mission Operations Services with Reference Implementation for ESA's OPS-SAT Mission (2017)

[5] ESA, MICONYS software suite: https://www.esa.int/Our_Activities/Operations/gse/MICONYS (accessed April 2019)

[6] Consultative Committee for Space Data Systems, Mission Operations Monitor & Control Services, CCSDS 522.1-B-1 (October 2017)