IMAGIN-e

Mission Overview

Written by	***Responsibility*** + handwritten signature if no electronic workflow tool
	Thales Alenia Space team Microsoft Team
Verified by	
Approved by	

Approval evidence is kept within the document management system

PROPRIETARY INFORMATION

THALES ALENIA SPACE LIMITED DISTRIBUTION

Change Records

ISSUE	DATE	§ CHANGE RECORDS	AUTHOR

Table of contents

1 Introduction

1.1 Scope and purpose

Edge Computing is understood as the processing and valorization of data in the edge of the cloud, as close as possible to the data source. In space this means processing and analysis of the data directly on the satellite. This novel concept is currently not well established in the space sector, where all data is downloaded and analyzed on the ground. However, it has a great potential to change the paradigm of space resource use, which could result in improved response time, optimization of frequency usage and greater control of the data and its value, improving the privacy and security of the system, to mention just a few of the advantages that are foreseen.

To validate the potential of this concept, a demonstrator is being developed and will be put it in orbit to test different use cases. The space segment of this demonstrator will be embarked on board the International Space Station and will be operational for one year. From the point of view of the ground segment, it will mean that for the first time, users will have a direct connection to a payload that is acting as a node of the Microsoft Azure cloud. This document describes the characteristics of the whole mission and the possibilities that it offers in the context of #orbitalAI challenge. This should allow the participants of the challenge to better understand which could be the most interesting thematic applications to propose in relation to the different constraints related to the design of the satellite and its concept of operations.

1.2 Terms, Definitions and Acronyms

The following abbreviations are used throughout this document:

AI	Artificial Intelligence
ADCS	Attitude Determination and Control System
AOCS	Attitude and Orbit Control Sub-system
CONOPs	Concept of Operations
EO	Earth Observation
FOV	Field Of View
FTP	File Transfer Protocol
Gb	Gigabytes
GSD	Ground Sampling Distance
ISS	International Space Station
LEO	Low Earth Orbit
LEOP	Launch and Early Orbit Phase
Mbps	Megabits per second
MC	Mission Computer
MSFT	Microsoft
MSI	Multi-Spectral Imager
OBC	On Board Computer
PL	Payload
TMTC	Telemetry and Telecommand

2 IMAGIN-e mission

Thales Alenia Space in collaboration with Microsoft are developing an experiment to gather unmatched Earth observation insights onboard the International Space Station.

The aim of this mission, called IMAGIN-e (**I**SS **M**ounted **A**ccessible **G**lobal **I**maging **N**od-e), is to demonstrate capabilities and operating modes of an architecture oriented to Edge computing in space.

Thales Alenia Space will deploy a powerful on-orbit computer, an on-orbit application framework, and high-performance Earth Observation sensors to unlock new on-orbit climate data processing applications for the benefits of our planet sustainability. Thales Alenia Space and Microsoft are working together in remote sensing, computer vision and climate science to demonstrate the potential of next-generation on-orbit compute for Earth observation. This space edge computing capacity will allow gathering faster, to-the-point Earth observation insights immediately applicable for our planet's surveillance, understanding and protection.

The on-orbit testbed will include on-board computing hardware as well as data collection sensor systems. This demonstration will enable application developers to easily create and deploy Earth Observation data processing applications on a real functional environment, at the crossing of space and cloud worlds.

Microsoft is providing an on-orbit software framework and space software development kit (SDK).

Among various use cases of Edge Computing in Space, the one finally selected lies on the Earth Observation field. This way several images will be generated on-board and processed by a dedicated processing computer. This processing computing will be flexible and accessible from ground during the mission duration in order to upload and deploy new solutions with processing algorithms implemented as well as to retrieve the processed (value added) data and the raw original images when relevant.

The Mission Computer (MC) will host different applications that will be updated and changed during the duration of the mission, including the two solutions awarded in the #ORBITALAI Challenge.

The top-level performances of this mission are as given in the next table:

Capability	Performance	Note
Payload Orbit	ISS Orbit Altitude = 370-460 km Near-circular orbit Inclination: 51,6º Eccentricity: 0,007	- Maximum accessible latitude: ±51° - Orbit period: 90 - 93 min - ISS pass over same zone/different time: 3 days (aprox.) - ISS pass over same zone/same time: 60 days (aprox.)
Platform	Located outside the Columbus module	
Mission Design Life	1 year in operation	2 years as an option
Orientation	Sensors pointed NADIR	
Field of View	20º	Minimum value
Spacecraft Agility	NA	The Payload has no control over attitude.
Pointing Accuracy and stability	Attitude rate non micro-gravity mode : ±0.05 deg/sec/axis Attitude knowledge on Bartolomeo: <1.0 deg/axis (3σ)	The Payload has no control over attitude. The ISS undergoes a constant jitter motion affecting payload attitude stability. Arrival of cargo/crew may disturb stability.

Payload Power	120 W	Maximum operational power provided by ArgUs adapter
D/L peak rate	10 Mbps	Through ArgUS connection
U/L peak rate	1 Mbps	Through ArgUS connection
Platform On Board Data Storage Requirement	Up to 500 GB	Temporary storage, shared between ArgUS users
Link Availability	70%	Estimated
Payload Thermal	Stable thermal environment must be maintained	Payload has active thermal control (heaters) to maintain equipment within safe ranges.

Table 1 – Top-level performances of the IMAGIN-e mission

2.1 Mission Architecture

The functional experiment will consist of two cameras capturing images of the Earth surface that will be stored for further processing, a Mission Computer (MC) devoted to the on-board processing of such images and an On-Board computer (OBC) devoted to manage the operation of the experiment and the connectivity towards the ISS.

This payload will be integrated as a node inside the Azure Orbital Space cloud computing platform where the MC will act as a peer of the rest of processors of the network. In this way, the end users shall be able to access the payload to perform the following tasks on the ISS node from its local nodes:

- Planning and execution of image capture tasks
- Planning and execution of image processing tasks
- Access to processed images
- Update of image processing algorithms
- Download processed results

To this end, IMAGIN-e space segment is composed of a payload (PL) that will be hosted on board the ISS, on the outside of the Columbus module; and the ground segment will be formed by the ground station of the service provider, to which two different interfaces will be connected: the one destined to operate the payload receiving the telemetries and sending the necessary commands, and the one that will connect with the end users, which will be operating through the Microsoft Azure cloud.

2.2 Payload Description

The mission consists of a space hardened multi-sensor payload connected to ground control and to a terrestrial cloud (Azure) thanks to a dedicated ground segment. This payload consists of:

- A hyperspectral camera, which acquires images at the time required by the user, performs radiometric correction and first-order processing of the acquired images. This is the camera proposed for usage in the context of #orbitalAI challenge.
- A processing module (MC), which interacts with the OBC. This computer will host the SDK developed by Microsoft and executes the image processing algorithms developed by the users.
- An On-Board Computer (OBC), which combines the functionalities of the payload and platform computer, being in charge of both TMTC channel management, power and camera control, and management of the MC (e.g., attending camera requests, MC health and SW updates).

- An additional RGB camera to provide general images of the scene. If interesting for the application, its usage can be also proposed as complement of the hyperspectral one.
- Additional modules for power and temperature management and for power supply distribution.
- A mechanical structure.

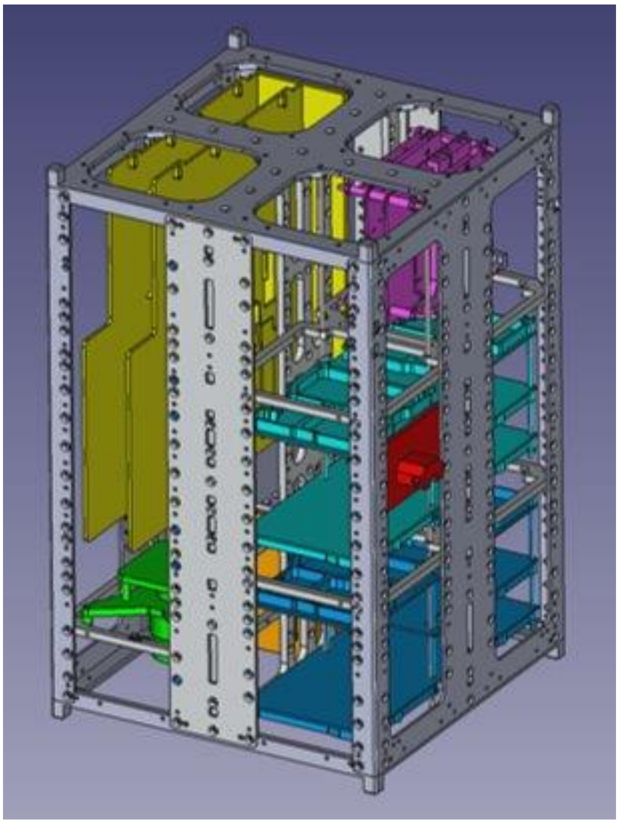

Figure 1 – Internal view of the PL

These last two elements are usually part of the satellite platform. However, in this mission, since the payload receives platform services from a service provider hosted outside the ISS, the payload itself will be in charge of power and thermal management for its operation, and communications with the host platform. The presence of an Attitude and Orbit Determination and Control Subsystem will therefore not be necessary either.

The MC will host the MSFT SW infrastructure (Azure Orbital Space SDK) to transparently run the customer applications developed by the Payload Applications Developer community. These applications will be granted to request, collect, process and download camera products, together with other platform generated data.

The figure shows a simplified architecture of the experiment payload. The OBC is the sole interface with the ISS infrastructure and controls the two cameras and the MC.

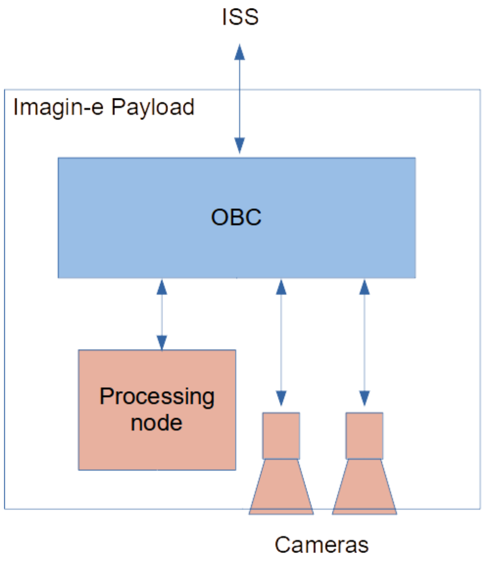

Figure 2 – Payload functional diagram

2.2.1 Logical connections

There are two types of users:

- Payload operator
- End user

The payload operator can access the OBC, supervise the telemetries and issue telecommands to the Imagin-e payload. The payload operator has privileged access to the payload resources.

The end user is capable of deploying applications that can use the processing resources (including AI capabilities), request images from the cameras, process them, and deliver back a set of processed results. The end user has no knowledge of the information flowing through the TMTC channel.

2.2.2 Hyperspectral camera

The hyperspectral camera in the visible and near infrared (VNIR) range is the HyperScout M model manufactured by COSINE REMOTE SENSING BV.

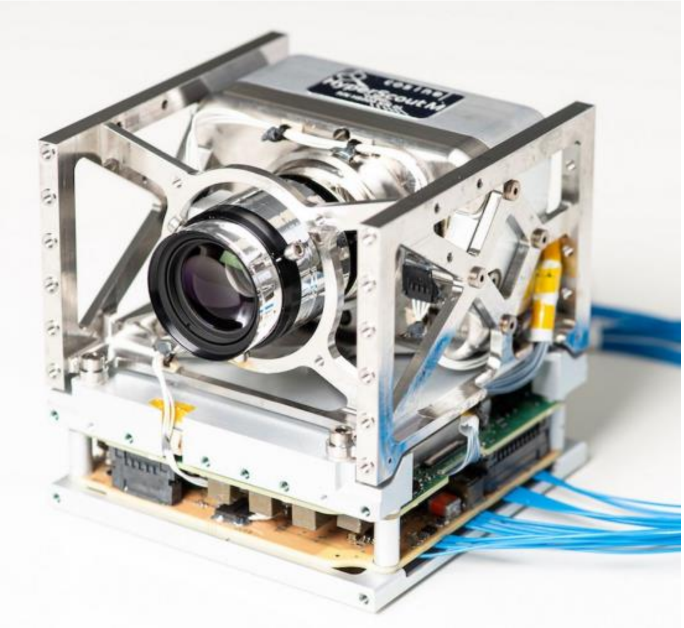

Figure 3 – IMAGIN-e hyperspectral camera

It allows a spatial resolution on the ground of tens of meters with a spatial coverage of tens of km. Its main features are:

- Volume: 1U (10 cm x 10 cm x 10 cm).
- Mass: 1200 g
- FOV: 25.3º x 11.6º.
- Spectral range: 450 nm - 950 nm (VNIR)
- Spectral resolution 20 nm
- Number of spectral bands: 50 (30 rows/band)
- GSD ACT from 450 km (ISS orbit): 50 m
- Swath from 450 km (ISS orbit): 200 km x 90 km (ACT x ALT)

Image acquisition is performed by sweeping the scene on Earth taking advantage of the motion of the ISS in its orbit ("pushbroom" acquisition without TDI). The frame period at 450 km (ISS) is 208 ms, which means 5 frames per second. The frame period leads to a shift between frames of the length corresponding to each spectral band (some overlapping exists). An image of an area of the Earth is acquired in each spectral band and the image set is stacked in a hyperspectral cube.

The camera is equipped with on-board image processing HW that allows to obtain a hyperspectral cube of the top of the atmosphere (TOA) with a georeferencing relative to the center of the cube, which allows to know the position on Earth of that center. In this sense, the available products on-board for the AI applications are:

- L0: raw data
- L1R: Top of Atmosphere DN in sensor geometry, central pixel geo-referencement, fine band-to-band alignment

The image processing from L0 to L1R is proprietary of COSINE REMOTE SENSING BV.

As of today, production of more advanced products, L1B or higher is not planned on board at launch. However, a SW upgrade could be available during mission operation, which will enable applications to have L1C products available.

2.2.2.1 Configurable parameters

This hyperspectral camera has several parameters that can be configured before capturing, in order to tailor the capture to the user's needs.

In order to simplify the experiment and to ensure the integrity of the camera throughout its duration, not all of these parameters will be available for configuration by the applications (the parameters that can be configured are TBC as of today).

2.2.3 RGB camera

The RGB camera in the visible (VIS) range is the NanoCam C1U model manufactured by GOMSPACE.

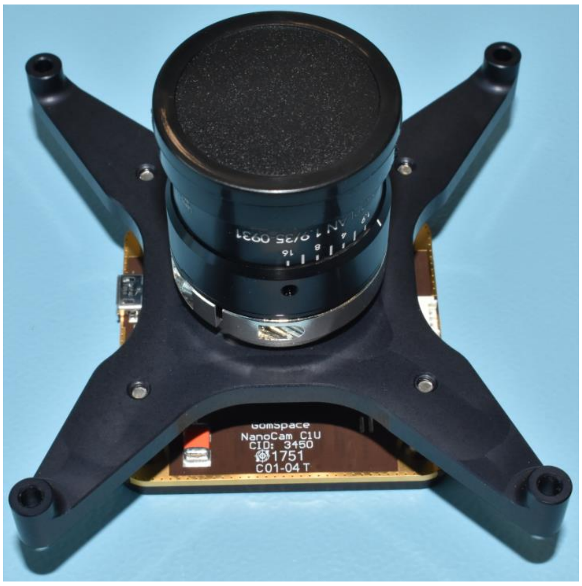

Figure 4 – IMAGIN-e RGB camera

It allows a spatial resolution on the ground of tens of meters with a spatial coverage of tens of km. Its main features are:

- Volume: 1U (86.0 x 91.7 x 57.9 mm).
- Mass: 169 g
- FOV: 10º x 4º.
- Spectral range: 400 nm - 750 nm (VIR)
- Number of spectral bands: 3 (RGGB Bayer matrix)
- GSD ACT from 450 km (ISS orbit): 40 m
- Swath from 450 km (ISS orbit): 82 km x 61 km (ACT x ALT)

Image acquisition is performed taking “snapsots” of the scene on Earth taking advantage of the motion of the ISS in its orbit.

The camera is equipped with on-board image processing HW that allows to obtain BMP and JPG images from RAW imaging data.

2.2.4 Mission Computer

This module is the heart of the payload as it will allow the transformation of the products acquired by the camera into relevant information for each of the users running their applications on it. Therefore, it will be responsible for the following activities:

- Loading user applications.
- Accessing the status and configuration of the cameras.
- Requesting captures from an experiment running autonomously.
- Access to the products processed in the chambers.
- Extract useful data from the processed products.
- Download data from user applications.
- Provide service to be monitored by the OBC.

It will also be able to update the in-flight SW of both the AI applications and the lower layers of the SW stack. The specifications of this processor module are:

- System memory: 32 GB of DDR4 SDRAM.
- Processing performance: >7 DMIPS/MHz/core and >4 Coremark/MHz/core. 16 Cortex-A72 cores, CPU clock 0.9 - 2 GHz, each core can independently be switched off, and its frequency can be adjusted.
- eMMC: 32 GB eMMC 5.1 4-bit eMMC 5.1 flash.
- F-RAM: 1 Mbit of non-volatile ferroelectric RAM.
- EEPROM: 1 x 256 Kbit EEPROM (for VPD data) and 1 x 256 Kbit EEPROM (system data).

2.3 AI Applications

The processing capabilities, and the imagery generated by the hyperspectral and RGB cameras will be available to a series of pre-selected applications (hereafter called Apps) that will run on the MC. Apps from multiple partners will be operating on the mission computer throughout the mission lifetime. In addition to the two AI Apps selected through the AI4EO competition, IMAGIN-e will support Apps written by partner academic institutions, as well as Apps developed by Thales Alenia Space and Microsoft. For example, Deeper Vision, an Earth Observation data analytics software by Thales Alenia Space, is one of the software intended to be run onboard the ISS using Space Edge Computing capacities. Scheduling of operational time and resources between Apps will be managed by the IMAGIN-e team with application developers input and participation.

2.3.1 Minimum Application Resource Allocation

Apps operating on the IMAGIN-e MC can general expect to operate with the following minimum compute resources.

- Memory: 8GB

- Processor: 4 Cortex-A72 cores at up to 2 GHz
- Scratch Storage: 2GB

With this minimal footprint, the IMAGIN-e can exercise the option to potentially serve multiple operational Apps at a time. Apps that require additional resources from the Mission Computer are not explicitly prohibited, but will require additional deconfliction with other resident applications.

3 Mission Timeline

The IMAGIN-e payload is designed for a 12-month lifetime from launch with potential extension up to 1 more year. Next figure shows the nominal top-level timeline of the mission, and Table 2 provides a description of the activities undertaken during each phase.

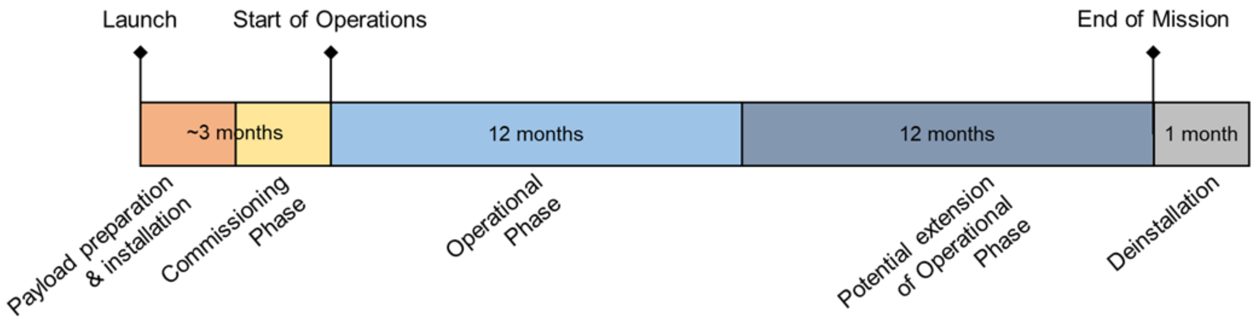

Figure 5 – Mission Timeline

Phase	Duration	Activities
Launch, Payload Preparation & Installation	3 months	• Launch IVA payload preparation • Installation outside the ISS • Initial health checks of the platform
Commissioning Phase		• Platform commissioning • Payload commissioning and calibration • Ground segment chain commissioning
Operational Phase	12 months	• Nominal payload operations • Platform maintenance • Nominal ground segment operations
Potential Extension of the Operational Phase	12 months	• Depending on the payload health status nominal operations can be extended for another 12 months
Deinstallation	1 month (TBC)	• Disassemble the payload from the ISS • Payload decommissioning

Table 2 – Summary of Mission Phases

4 Mission operations

4.1 Payload calibration

Cameras calibration will be performed during commissioning phase. It will consist of the following:

- Hyperspectral camera: update of the calibration parameters set at on-ground calibration campaign by the camera manufacturer and optimization of the working point of the camera to maximize SNR and MTF in different scenarios.
- RGB camera: optimization of the working point of the camera to maximize SNR in different scenarios.

4.2 Capturing strategy

User applications will be able to make requests to the payload sensors, by choosing a specific sensor, selecting a synchronized capture from both cameras, or asking for a specific region of interest from an image. These requests may be scheduled and will be queued and executed on a first-come, first-served basis. Some platform data, as instantaneous ISS position and attitude, is also made available for the users.

In order to guarantee the correct camera usability and safety, optimizing the image processing SW pipeline, users will only have access to part of the camera parameters (from the full set, configured by the payload operator) to configure capture tasks according to the specific needs of his application.

5 Applications development

As part of the open competition, there is no requirement for AI model integration within the On-Board application environment. Additional requirements will be provided during the Incubation phase, and in any case supported by Thales Alenia Space and Microsoft.

5.1 IMAGIN-e Basic Simulator

As part of the OrbitalAI challenge, a IMAGIN-e basic simulator is provided in order to help the challenge participants to create realistic App. To create this simulator, the main requirements were to provide the user with an easy-to-use tool, allowing to simulate in a relatively realistic way the different products generated on board, without geographical or temporal coverage constraints. Even if the payload of the IMAGIN-e mission has been the subject of specific studies on its intrinsic performances, the actual performances of the mission once in orbit cannot be fully simulated. Moreover, the possibility of simulating any AOI, without constraint of cost or commercial license, imposes this simulator to use as input the PRISMA data. Indeed, the spectral and spatial characteristics of PRISMA represent a very good starting point for IMAGIN-e. The possibility to scale the simulation for any time and/or location is enabled by the possibility to use Sentinel-2 as input, with all the known limitations to emulate a hyperspectral data cube from a multispectral imager. This simulator is therefore provided with all its limitations.

The ATBD of the IMAGIN-e Basic Simulator is provided in the document "Data Simulation Design for the HyperSpectral Imager".

5.2 On-Board Application Environment

5.2.1 Azure Space SDK Framework

The Azure Orbital Space SDK is a k3s Kubernetes -based microservices platform that abstracts spacecraft telemetry and sensor data to provide a common interface for satellite operators and payload application developers to use data collected in-orbit. Using the Azure Orbital Space SDK APIs, developers will be able to reuse applications across different families or generations of satellite hardware, telemetry, and sensor data sources. This framework runs on both ARM64 and AMD64 processor architecture, and standard Linux distributions, e.g., Ubuntu. A high-level architecture diagram of the Space SDK framework is provided below:

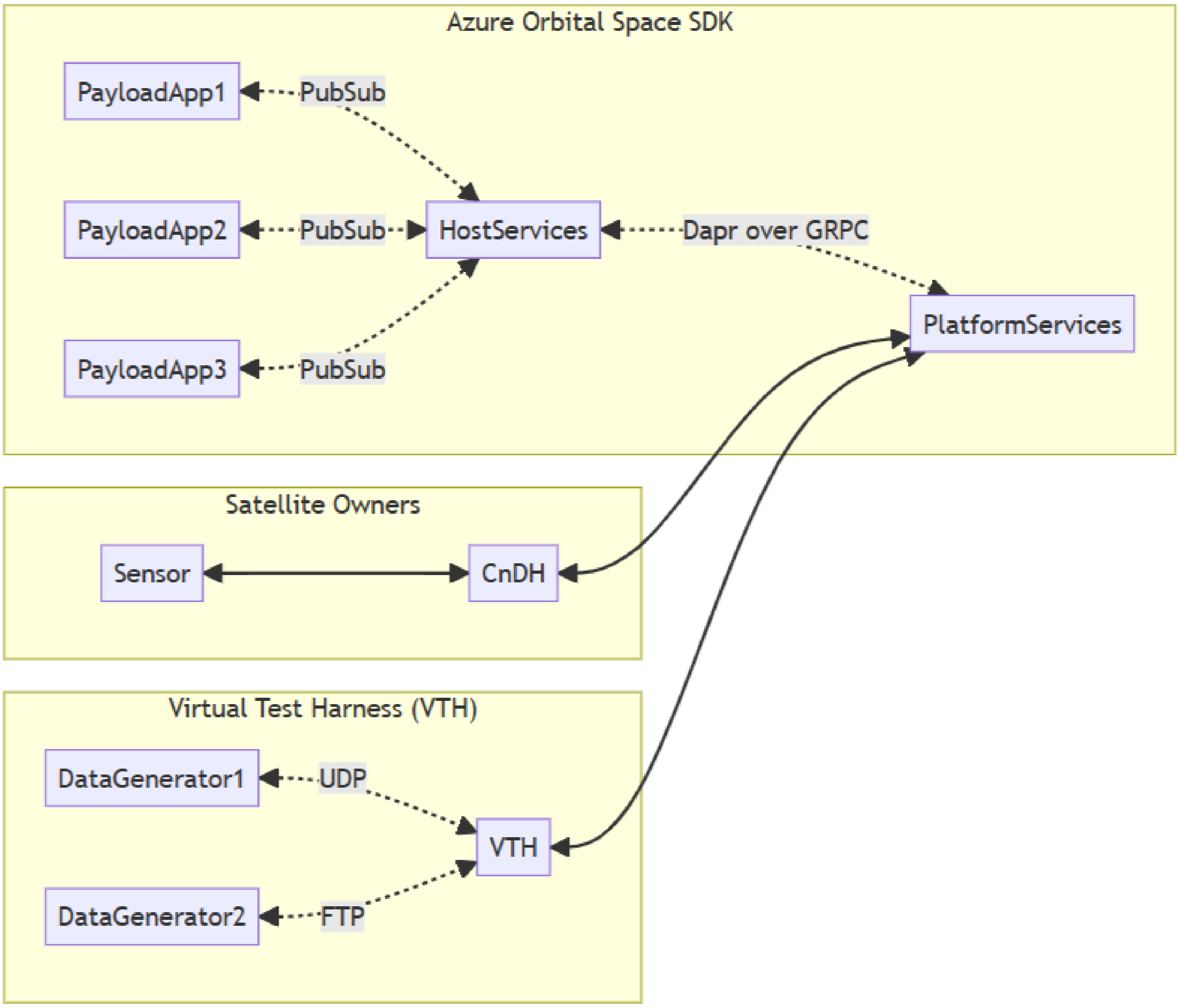

Figure 6 high-level architecture diagram of the Space SDK framework

Internal to the framework, software services are divided into three categories: Platform Services, Host Services, and Payload Applications. Platform and Host services are built-in service offerings that implement the differentiated compute framework of the Space SDK. Payload applications are written by Payload Application Developers to execute a specific on-orbit workload.

Platform Services abstract on-board spacecraft systems giving payload applications (via the Host Services) and Satellite Owner Operators a common interface to interact with the spacecraft. This layer of abstraction is nearer to the spacecraft and its hardware than the Host Services. For those reasons, Platform Services are managed and interacted with exclusively by the Satellite Owner Operator (SOO) and are logically isolated from the payload applications and Host Services by Kubernetes namespaces.

The Azure Orbital Space SDK Platform Services today provide Satellite Owner Operators with two capabilities:

- The **Message Translation Service** to translate telemetry and sensor data from the spacecraft to a common format
- The **Deployment Service** to deploy, update, and stop update payload applications

Host Services provide payload applications with the ability to interact with spacecraft telemetry, sensor data, and coordinate communication with the ground by abstracting implementation details about the spacecraft and its hardware.

The Azure Orbital Space SDK exposes the following Host Services to payload applications via client libraries for .NET and python:

- The **Sensor Service** to subscribe to spacecraft telemetry for sensor data
- The **Position Service** to subscribe to spacecraft telemetry for positioning
- The **Link Service** to send and receive messages to and from the ground
- The **Logging Service** to write to logs

5.2.2 Ground-Based Development Environment

Writing apps that run predictably in space can be challenging for a myriad of reasons. To assist developers in overcoming those challenges from the ground, the Azure Orbital Space SDK provides a Virtual Test Harness (VTH). The VTH is used with spacecraft or sensor specific Data Generators to feed applications under development with information as it would be available on a spacecraft. With the VTH:

- Developers can develop and test in a predictable and repeatable fashion
- Developers do not have to wait for a spacecraft to be available
- Developers can replicate scenarios they would see on-orbit without code changes to their applications

The Azure Orbital Space SDK uses the term "data generator" as a generic term for any software that produces data that can be used to test an application. A Data Generator can be an HTTP service, an FTP server, a blob container, a file on disk, or any other source of data.

5.3 Application Lifecycle

5.3.1 Development & Test

Payload Application Developers begin their development experience by setting up a dedicated development system. This can be done on a local Linux machine running Ubuntu, or with an Azure cloud-hosted Virtual Machine. Using Microsoft Visual Studio Code remote development tools, developers can build applications in a Kubernetes native DevTest environment with live debugging and interact with data generators using VTH.

Users of the AO Space SDK are provided access to Microsoft developed application templates, sample applications, and sample data generators to aid in the development of new Payload applications. Included with the SDK is the ability to live run and debug software with data generators connected to the framework via the VTH.

Once an application developer has completed a release of their app, the developer builds the app into a application package, including container images, deployment specifications, and any ancillary data files required for the application to run. The developer can then test the deployment of their app via an Azure DevOps, or equivalent, build pipeline.

Once the Payload App Developer is satisfied with the development and testing of their app, they will hand over their application to the IMAGIN-e team for Application Verification (see below).

5.3.2 Application Verification

Applications going to orbit on the IMAGIN-e payload will be required to go through a verification process prior to uplink and operation on-orbit. Developers will follow the develop-build-test procedure outlined above in the Application Lifecycle. Once an application has been tested and built by the developer, it is provided to the IMAGIN-e team for verification.

Developers will provide the IMAGIN-e team with a development specification and context files (e.g. configuration information, ancillary data) required to verify the applications functionality and interactivity with the Space SDK framework. This set of verification tests will exercise the functionality of the application on the ground using the VTH and IMAGIN-e data generators.

An application is verified once the IMAGIN-e team has been able to successfully reproduce the deployment tests provided by the developer, and the developer has confirmed that the application has operated as expected through an inspection of the logs and test results.

5.3.3 Operations

Verified applications go through two operational stages, 1) Uplink 2) Scheduled Deployment 3) Results Delivery

5.3.3.1 Uplink

Verified applications need to be moved from the IMAGIN-e ground system onto the on-orbit payload in advance of scheduling operations. This includes providing the required new / updated application containers. Subject to the availability of storage on-orbit, application developers will work with the IMAGIN-e team to schedule an operational period days or weeks in length.

In advance of that agreed upon period of operation, The IMAGIN-e team will upload the application containers to the IMAGIN-e payload, and stage them in the on-orbit registry. In order to verify that an application has been successfully uploaded, the IMAGIN-e team will conduct a basic test of the Application, confirming that the application containers can be deployed, operated, and shut-down. Payload Application Developers will receive notification from the IMAGIN-e team once the application uplink has been completed and verified.

At the completion of this stage, the application is ready for normal scheduled deployment.

5.3.3.2 Scheduled Deployment

Scheduled deployment is the primary means of operating application on the IMAGIN-e payload. During the operational period for an application, the app developer and Microsoft team will coordinate, at a minimum, Bi-weekly to identify and schedule opportunities to operate the app. For each scheduled application runtime, application developers will deliver deployment specifications and contextual data. The IMAGIN-e team will maintain a master plan of schedules for applications.

Application deployments are scheduled with the uplink of schedule files developed and maintained by IMAGIN-e team according to the agreed upon scheduled opportunities established with developers. The IMAGIN-e team will provide application developers with feedback at the conclusion of each application runtime opportunity to indicate if the application executed successfully and, if not, the specific conditions observed by the Space SDK Framework.

5.3.3.3 Results Delivery

Each application runtime can result in two types of output that will be delivered to application developers, 1) collected logs 2) downlinked data products. Application developers and the IMAGIN-e team will have arranged in advance where these results will be delivered or accessible. Logs and data products will be made available as soon as possible, dictated primarily by the downlink bandwidth provided by the Airbus Bartolomeo platform and the ISS communications infrastructure.

5.3.4 Application Update

Updating applications will follow roughly the same procedure as initial development, test, and uplink. Application developers will provide the updated container images and deployment specifications describing their app to the IMAGIN-e team, tagging these artifacts with unique versioning.

The IMAGIN-e team will be responsible for managing the state of the on-orbit registry to hold the container image versions required by the Application, consistent with each operational period and application runtime. Subject to the availability of storage space on-orbit, multiple versions of an application may not be available at the same time. Therefore, at the opening of the operational period, application developers will specify to the IMAGIN-e team which version(s) of an app are required or requested.

END OF DOCUMENT