LEONSEGS: Ground Segment as a Service and Cross Mission Planning for SmallSat Operations

Álvaro Núñez Martínez
GMV
Madrid, Spain
anunez.m@gmv.com

***Abstract*— LEONSEGS is a federated ground segment platform that combines Ground Segment as a Service with multi mission planning and semantic archive retrieval to execute end to end satellite operations and Earth Observation (EO) product generation across heterogeneous missions and external EO catalogues. The architecture integrates a mission operations and processing stack that includes flight dynamics, mission planning, mission control, orchestration and payload data processing, together with an automated EO service layer for request handling, multi mission planning, semantic discovery and multi mission EO product generation.**

At system level, operations and EO service delivery are jointly realised by Mission Operations Control and Processing Service (MOCPS) and Automated Multi mission Earth Observation Service (AMEOS) segments. The former turns user intent into executable plans, schedules ground contacts, commands and monitors spacecraft, and brings payload data to ground and into the processing chain. The latter captures and brokers requests, selects missions and data from internal and external catalogues through semantic discovery, and assembles the resulting EO products. The two segments exchange plans, events, telemetry summaries and product metadata through consistent interface contracts and shared data models aligned with community standards, including STAC and OGC APIs for catalogues and product descriptions, and CCSDS recommendations for planning, control and telemetry. This standardised approach enables straightforward integration with third party missions and ground station providers, supports automation across planning and execution, and ensures portability across platforms.

Keywords—Ground Segment as a Service (GSaaS), Multi-mission, Federation, Earth Observation (EO) Services

I. Introduction

Small satellite missions are increasingly expected to deliver responsive operations and timely Earth Observation (EO) products while operating in an ecosystem of heterogeneous spacecraft, ground station networks, processing chains, and data catalogues. In practice, this heterogeneity often results in duplicated integration effort, fragmented operational tooling, and limited automation, especially when scaling from a single mission to multi-mission service models. At the same time, EO users and downstream applications demand shorter turnaround times from request to product delivery, and they frequently require access to data distributed across multiple internal and external archives.

Two trends are therefore converging. On the one hand, Ground Segment as a Service (GSaaS) approaches seek to reduce operational burden by providing mission operations capabilities, such as planning and scheduling, monitoring and control, and processing integration, as reusable services that can be consumed by different missions. On the other hand, EO service delivery increasingly requires higher-level automation: brokering user requests, selecting the best mission or provider under constraints, discovering relevant data across distributed catalogues (including beyond pure metadata matching), and assembling EO products that may combine multi-source inputs. Addressing these needs in isolation is insufficient; an end-to-end solution must bridge service-level intent with operational execution and data exploitation, while preserving interoperability across missions and third-party providers.

This paper presents LEONSEGS as a federated ground segment platform that jointly realises mission operations and EO service delivery across heterogeneous missions and external EO catalogues. The system is structured into two cooperating segments. The Mission Operations Control and Processing Service (MOCPS) provides GSaaS capabilities to translate user intent into executable operational artefacts, including mission geometry and event timelines, constrained planning and scheduling (with ground contact allocation), real-time monitoring and control, orchestration across the planning–execution boundary, and the ingestion of payload data into processing chains. In parallel, the Automated Multi-mission Earth Observation Service (AMEOS) captures and brokers EO requests, performs cross-mission planning and selection (including routing to external providers when applicable), supports discovery across internal and external catalogues, leveraging semantic-based retrieval to enhance relevance and contextual understanding, and assembles EO products for delivery.

The main contributions of this work are:

1) An end-to-end, federated architecture that integrates GSaaS mission operations with an automated EO service layer, enabling consistent workflows from user request to planning, execution, data ingestion, and product delivery.

2) A multi-mission orchestration approach that supports routing and coordination across heterogeneous missions and third-party providers through consistent interface contracts and shared operational artefacts (plans, schedules, events, telemetry summaries, and product metadata).

3) Representative end-to-end workflows that exercise archived retrieval, new tasking, provider-based tasking, and GSaaS monitoring and control, establishing a structured baseline for validation and future scaling.

II. System Overview and Concept of Operations

LEONSEGS is a federated platform that connects mission operations capabilities delivered as a service with EO service workflows that span internal missions, external providers, and distributed catalogues. The system is externally perceived through a small set of stakeholder roles and through end-to-end operational workflows that translate user intent into plans, scheduled contacts, execution outcomes, and EO products. Fig. 1 summarizes the system context and the main external stakeholders.

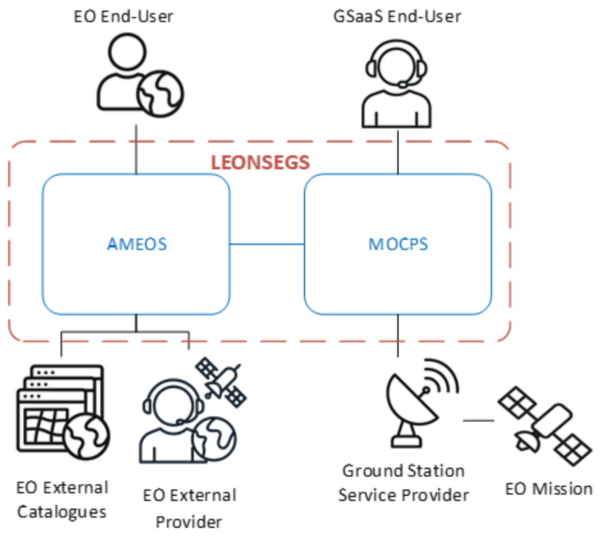

Fig. 1. System context and high-level segmentation of LEONSEGS into AMEOS and MOCPS, interacting with EO end-users, GSaaS users, external catalogues/providers, ground station providers, and EO missions.

A. *Stakeholders and Roles*

EO End-User: An individual or organization consuming EO products generated by EO services for operational, scientific, or commercial purposes. The EO end-user interacts with LEONSEGS to request domain-specific EO services (e.g., imagery acquisition, archived retrieval, value-added products) and receives actionable EO products as output.

GSaaS User: A satellite operator or New Space company using LEONSEGS to externalize ground segment operations, including mission planning, scheduling, monitoring and control, and integration of their space assets. The GSaaS user operates missions through the platform without deploying dedicated ground infrastructure, while preserving mission-specific constraints and operational governance.

LEONSEGS Operator: The technical administrator responsible for configuring, maintaining, and supervising the platform and its subsystems. The operator ensures platform integrity and availability, manages access-control and security policies, and oversees compliance with applicable legal and operational requirements.

Ground Station Provider: A company or organization offering ground station infrastructure and services for satellite communication, including uplink/downlink support and contact execution. The provider exposes contact opportunities and execution feedback to the platform, enabling contact allocation and scheduling within end-to-end workflows.

EO Product Provider: An entity supplying EO products, either from its own EO missions or through data processing services. Product providers contribute to the LEONSEGS ecosystem by offering catalogues, tasking interfaces, and/or value-added processing chains that can be consumed by EO services.

EO External Catalogue: A discoverable repository of EO metadata and/or products exposed through catalogue interfaces. An EO External Catalogue enables search and discovery (e.g., by spatial/temporal constraints, sensor attributes, and other metadata) and may provide access links or retrieval endpoints for the underlying EO products. The catalogue itself does not necessarily own the space assets; it acts as an index and access layer to archived EO holdings.

EO Mission: A space segment asset (satellite or constellation) capable of acquiring EO data and generating mission-level payload products. An EO Mission may be integrated internally into the platform through mission operations services (enabling planning, scheduling, monitoring and control), or accessed indirectly through provider interfaces when the mission is operated by a third party. In both cases, an EO Mission is the source of new acquisitions and the ultimate origin of mission-generated EO data products.

B. *Operational Concept of Operations (ConOps)*

LEONSEGS operates as a federated service environment in which stakeholder intent is translated into operational artefacts and execution outcomes through a consistent lifecycle. From an external viewpoint, the platform provides a unified operational loop that remains stable across heterogeneous missions and providers, while allowing different fulfilment paths (archived retrieval, new acquisition, provider-based tasking) to be selected transparently according to feasibility and constraints.

The ConOps is structured around five recurring phases:

1) *Request lifecycle and governance:* Stakeholders submit service intent (e.g., AoI, time window, priority, and service constraints) and receive a request identifier used for end-to-end traceability. The platform enforces access-control and policy constraints based on stakeholder role (EO End-User, GSaaS User, LEONSEGS Operator) and maintains request state transitions (submitted, assessed, planned, scheduled, executed, delivered) to support operational governance.

2) *Feasibility and routing across a federated ecosystem:* The platform assesses fulfilment options by combining mission/provider capabilities with operational constraints. The outcome of this phase is a routing decision: (i) fulfilment through an internally operated mission supported by mission operations services, (ii) fulfilment through an external EO product provider, and/or (iii) fulfilment through archived products discovered in internal or external catalogues. This routing decision defines the downstream operational path while preserving a consistent upstream service interface.

3) *Operational planning and contact allocation (mission operations backbone):* For requests routed to internally operated missions, the service intent is transformed into mission-level operational artefacts. These artefacts include planning orders, activity timelines, and contact schedules that incorporate ground contact opportunities and allocations across one or more ground station providers. Planning outputs are generated under mission-specific constraints and operational policies, enabling repeatable planning behaviour across missions while retaining mission uniqueness through configuration.

4) *Execution supervision and operational reporting:* Operational artefacts are executed under supervision, resulting in traceable operational evidence. Execution is reported through structured events and status indicators (e.g., contact execution outcomes, operational state changes, and telemetry-derived summaries) that support operational monitoring, anomaly awareness, and post-pass assessment. This phase closes the operational loop between plan/schedule generation and verified execution outcomes.

5) *Data ingestion, processing-chain handover, and product availability:* Payload data and intermediate outputs are ingested into the data handling pipeline and handed over to processing chains as applicable. The platform publishes product availability through product metadata and delivery references, enabling EO product delivery workflows and downstream exploitation. When fulfilment relies on external catalogues or providers, equivalent product references and provenance information are captured to preserve traceability.

Across all phases, LEONSEGS maintains correlation between service-level intent and mission-level execution by linking each request identifier to the associated operational artefacts (e.g., planning orders, schedules), execution evidence (events and telemetry-derived summaries), and resulting EO products and metadata. This artefact-driven lifecycle enables automation and interoperability across heterogeneous missions and third-party providers while preserving operational governance, accountability, and end-to-end traceability.

C. Scope and Assumptions

LEONSEGS is developed as a Proof of Concept (PoC) prototype targeting a TRL6-level demonstration in a relevant environment, rather than as a fully industrialized operational system. Within this context, the scope of the PoC is to demonstrate an integrated end-to-end service lifecycle in a federated setting: request capture and routing across internally operated missions, external providers, and external catalogues; mission-level planning and ground contact allocation for internally operated missions; execution supervision with operational reporting (including structured events and telemetry-derived summaries); and payload data ingestion with processing-chain handover leading to product availability through product metadata and delivery references. The emphasis is placed on interoperability, automation across the planning–execution boundary, and artefact-level traceability, as these are the key enablers to scale towards multi-mission operations.

Multi-mission capability is assumed to be achieved through bounded integration effort enabled by consistent interface and shared operational artefacts, such as planning orders, schedules, execution events, telemetry summaries, and product metadata, so that heterogeneous missions and providers can be incorporated without redesigning the end-to-end workflow. Ground station providers are treated as service endpoints exposing contact opportunities, booking and execution interactions, and pass outcome reporting, allowing scheduling and execution to be performed in a provider-agnostic manner. External catalogues and providers are assumed to support discovery and access mechanisms sufficient to retrieve products or delivery references, so that third-party fulfilment paths remain traceable and comparable to internally operated mission flows.

Given the PoC/TRL6 nature of the work, hardware-specific TT&C implementation details (e.g., RF chain design and antenna control internals), mission-specific algorithmic depth (such as detailed attitude dynamics modelling), and advanced global optimization methods for large constellations are outside the scope of this paper. The focus is instead on system-level decomposition, interfaces, and artefact-driven operational workflows that can be validated in a representative environment and serve as a baseline for subsequent hardening and industrialization.

III. Architecture and Interfaces

LEONSEGS is designed as a federated, service-oriented platform that jointly realises mission operations capabilities (delivered as GSaaS) and EO service workflows spanning internal missions, external providers, and distributed catalogues. The Proof of Concept targets a TRL6-level demonstration of an integrated system, with particular emphasis on multi-mission interoperability as the main technical challenge.

A. Segment Decomposition and Design Rationale

Following the segmentation defined for the LEONSEGS system, the architecture is decomposed into two cooperating segments: (i) the Mission Operations Control and Processing Service (MOCPS), which provides GSaaS capabilities for New Space missions and satellite operators, and (ii) the Automated Multi-mission Earth Observation Service (AMEOS), which enables EO end-users to request and obtain EO services and products. This decomposition separates mission-execution responsibilities (planning, ground contacts, monitoring and control, payload data handling) from service-level brokering and multi-source exploitation (request handling, mission/provider selection, catalogue discovery, and product assembly), while enabling end-to-end automation through harmonised interface contracts and shared operational artefacts.

B. MOCPS: Mission Operations Control and Processing Service (GSaaS Backbone)

MOCPS implements the mission operations backbone of LEONSEGS. It interconnects with the (simulated) New Space operator space segment and provides capabilities for satellite command and telemetry services, ground station contact planning, on-orbit sensor tasking, raw sensor data download, and mission-level payload data processing. MOCPS also supports the delivery of processed outputs and metadata to AMEOS for service-level product assembly.

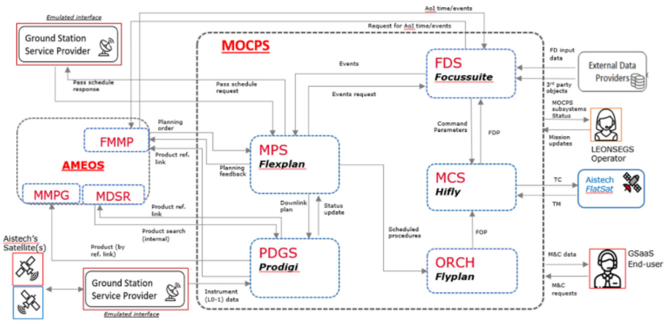

Fig. 2. MOCPS subsystems decomposition, external and internal interfaces.

MOCPS comprises the following main functional elements:

- Flight Dynamics System (FDS): Computes orbit determination/propagation, coverage and access windows, and provides geometry/attitude products to support planning and safe operations; Space Situational Awareness (SSA) is included as part of the FDS.
- Mission Planning System (MPS): Transforms mission/user requests, constraints, and resources into executable schedules; resolves conflicts and optimises alternatives; exports plans for automated execution and monitoring.

- Mission Control System (MCS): Provides real-time spacecraft monitoring and control (TM/TC), out-of-limit checks, command encoding/verification, and TM/TC archiving, exposing operational procedures for execution.
- Operations Orchestrator (ORCH): Coordinates and sequences ground/flight activities across subsystems, enforces dependencies and time windows, triggers automated procedures, and monitors their execution end-to-end.
- Payload Data Ground Segment (PDGS): Processes raw payload data and provides access to the mission-specific catalogue/archive; for systematic processing scenarios, PDGS supports ingestion, processing chains, archiving and cataloguing, and dissemination of outputs, including catalogue interfaces such as STAC where applicable.

In the PoC, MOCPS is largely realised by adapting existing GMV products for flight dynamics, mission planning, monitoring and control, operations automation/orchestration, and payload processing to a multi-mission service model. This approach supports rapid prototyping while focusing innovation on multi-mission interoperability, orchestration, and interface harmonisation.

C. AMEOS: Automated Multi-mission Earth Observation Service (EO Service Layer)

AMEOS is the EO service layer and the primary interface to EO end-users. It manages EO product requests, interconnections with federated EO product providers and external catalogues, and coordinates multi-mission EO product generation. AMEOS is also enhanced by a Data Semantic Retrieval capability that improves the relevance and accuracy of catalogue imagery retrieval and supports contextual interpretation of user queries.

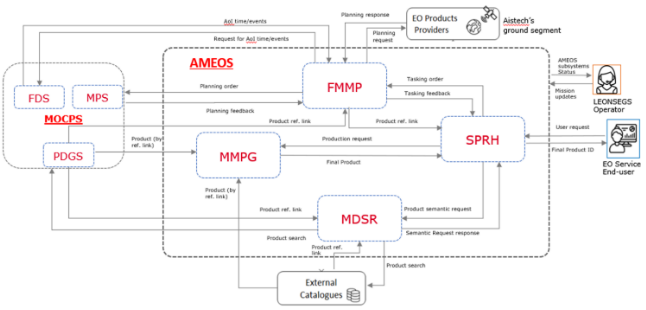

Fig. 3. AMEOS subsystems decomposition, external and internal interfaces.

AMEOS comprises four main functional blocks:

- Smart Product Requests Handling (SPRH): Entry point for EO end-users to request EO services by specifying time frame, Area of Interest (AoI), and the desired EO service; it manages order/request lifecycle and user interaction, and triggers downstream planning/discovery/processing as required.
- Flexible Multi-Mission Planning (FMMP): Selects the most suitable mission to fulfil a request; the selected mission can be an external EO product provider or an internally operated mission integrated via MOCPS. FMMP includes flexible interfaces (e.g., REST APIs, file-based exchange, or message broker integration) and template-based request/response adaptation to accommodate heterogeneous provider interfaces.
- Multi-mission Data Semantic Retrieval (MDSR): Retrieves products from available catalogues (external catalogues and/or mission-specific catalogues exposed by PDGS). Depending on catalogue capabilities, MDSR performs metadata-based or semantic-based retrieval, leveraging STAC-compliant catalogue access and supporting scalable asynchronous request processing.
- Multi-Mission Product Generation (MMPG): Generates the final EO product/service delivered to the end-user. In the PoC, MMPG ingests required inputs from multiple sources (external catalogues and providers, and/or PDGS), triggers on-demand processing chains, and publishes resulting products with associated metadata and delivery references.

D. Interfaces Overview

LEONSEGS relies on a compact set of external and internal interfaces designed to enable automation and end-to-end traceability across heterogeneous missions, providers, and catalogues. At the system boundary, EO end-users submit service requests through a web-based interface over HTTPS and receive product availability and delivery information through user notification and controlled download over HTTPS, with products packaged to support multi-file deliveries. Ground segment operations are integrated with ground station providers through pass scheduling interactions; where available, integration is API-based (commonly REST), while the architecture also accommodates file-based exchanges to interoperate with providers that expose legacy or constrained interfaces. For acquisitions via federated EO product providers, the platform exchanges planning requests and responses using provider-dependent interfaces (typically REST), with the intent to minimise ad-hoc adaptations by converging towards standardised planning request patterns for future provider integrations.

For discovery and archive access, LEONSEGS standardises catalogue interaction around STAC APIs whenever possible. External catalogue search and results use STAC over HTTP(S), and the same STAC-based approach is used for internal catalogue discovery when mission products are exposed by the payload data ground segment. Internally, LEONSEGS follows an artefact-driven integration model that decouples subsystems while preserving traceability. Long-running discovery and production flows use asynchronous patterns (e.g., request artefacts exchanged via object storage and status updates via message queues), enabling scalable orchestration without tight coupling. Between the EO service layer and the mission operations backbone, planning coordination combines file-based exchanges for orders with message-based feedback channels, while time-sensitive feasibility queries use synchronous service calls.

Across all flows, a request identifier is propagated and correlated with operational artefacts (planning orders, schedules, execution outcomes and status, ingestion state) and with published product metadata and delivery references. This enables consistent automation and reporting across heterogeneous missions and third-party providers while maintaining a stable upstream interface for users.

E. *Cross-Cutting Platform Functions and Deployment Considerations*

To sustain the service-oriented model and ensure consistent operation across AMEOS and MOCPS, LEONSEGS incorporates cross-cutting platform functions for identity and access management, usage/accounting support, and observability/monitoring. User and rights management is provided via Keycloak, supporting role/group-based access control and the option to serve multiple subsystems with separated realms. Billing and accounting are implemented within SPRH (with payment-gateway integration as an operationalisation step). System monitoring is supported through centralised log collection and health monitoring components that provide alarms and dashboards.

For the PoC, the prototype is hosted on GMV infrastructure within a virtualised environment and leverages containerised services to enable flexible resource allocation and scalability. Standard enterprise-grade security measures are applied, including role-based access controls and platform hardening practices appropriate to a prototyping environment.

IV. End-to-End Use Cases

This section instantiates the concept of operations described in Section II by presenting the four end-to-end use cases selected for the LEONSEGS Proof of Concept. The use cases collectively exercise archived data exploitation, internal mission tasking through GSaaS, provider-based tasking, and GSaaS mission onboarding and monitoring & control (M&C). The section focuses on UC-LEONSEGS-200 because it covers the complete planning–scheduling–execution–ingestion loop for an internally operated mission and therefore best demonstrates the coupling between the EO service layer and the mission operations backbone.

TABLE I. USE CASES AND PRIMARY CAPABILITIES EXERCISE

Use Case	Primary goal	Capabilities exercised
UC-LEONSEGS-100 (Archived)	Retrieve archived imagery/products for a past time window and AoI.	Catalogue discovery (semantic or metadata), retrieval, processing, dissemination.
UC-LEONSEGS-200 (GSaaS Tasking)	Task an internally operated mission and deliver a processed EO product.	Mission selection, FD feasibility, mission planning & ground contact scheduling, execution supervision (TM/TC), payload downlink/ingestion, mission-level processing, final product generation & dissemination.
UC-LEONSEGS-300 (EO Product Provider Tasking)	Task an external EO product provider and deliver a processed EO product.	Provider selection/tasking, provider planning response tracking, retrieval of provider product, optional post-processing, dissemination.
UC-LEONSEGS-400 (GSaaS M&C)	Onboard a mission and operate it via GSaaS monitoring & control services.	Mission onboarding/config propagation, planning under mission constraints, ground contact scheduling, orchestrated execution, TM/TC operations, operational reporting.

A. *UC-LEONSEGS-200: Satellite Tasking via GSaaS*

UC-LEONSEGS-200 demonstrates how LEONSEGS fulfils an EO service request that requires a new acquisition by a mission operated within the platform. The use case begins with a user request defined by area of interest (AoI), a future time window, and a target spatial resolution, and ends with delivery of a processed EO product. The workflow coordinates service-level request handling and routing with mission-level feasibility, planning, contact scheduling, execution supervision, payload data downlink/ingestion, and final product generation.

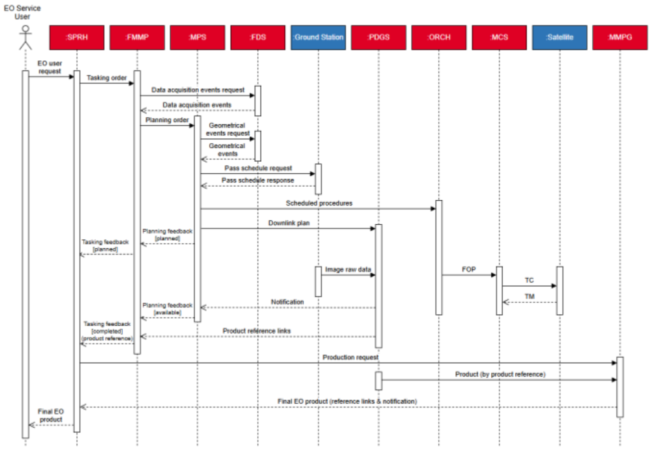

Fig. 4. UC-LEONSEGS-200 UML diagram (satellite tasking via GSaaS and subsequent processing).

Step 1. Request capture and tasking order creation:

The EO end-user submits a service request by selecting an AoI, a future time period, and the desired resolution. The request is captured by the service front-end and translated into a structured tasking order that can be evaluated and routed for fulfilment.

Step 2. Mission selection and feasibility inputs:

The multi-mission planning function selects a suitable mission that can be operated by the platform for the requested acquisition. Feasibility is refined by requesting AoI visibility/access times and acquisition opportunities from flight dynamics, producing candidate events and timing constraints that bound what can be scheduled.

Step 3. Mission planning and schedule refinement:

A mission-level planning order is issued to the mission planning subsystem. Mission planning refines the acquisition timeline and requests additional geometrical events from flight dynamics as needed. Based on the refined timeline and constraints, mission planning coordinates with a ground station provider to generate and confirm a pass schedule that supports both command/telemetry exchange and payload downlink.

Step 4. Execution preparation and coordination:

The confirmed schedule is propagated to the operations orchestrator for execution control, while a downlink plan is delivered to the payload data ground segment to prepare ingestion and expected downlink tasks. In parallel, planning feedback is reported back to the service layer so the end-user can track request progress.

Step 5. Pass execution (TM/TC) and supervision:

At execution time, the orchestrator triggers the planned operational procedures within the allowed time windows. The mission control subsystem performs telecommand and telemetry operations during the pass, while the orchestration layer supervises procedure execution and captures operational evidence (e.g., pass start/stop, completion states, and relevant status indicators).

Step 6. Payload data downlink, ingestion, and mission-level product creation:

During and after the pass, the payload data ground segment receives the payload data stream, ingests it according to the downlink plan, and performs mission-level processing to produce an initial mission product suitable for archiving and catalogue exposure. Once archived and catalogued, the payload segment reports completion and provides product references that can be used for downstream product generation and status tracking.

Step 7. Final product generation and dissemination:

The service layer composes a production request for the final EO service output and submits it to the product generation function. Product generation retrieves the required mission-level products, runs the processing chain required for the requested EO service, and publishes the resulting product with associated metadata and a delivery reference. The service front-end then notifies the EO end-user and provides controlled access to download the product.

Artefact-driven traceability:

UC-200 is intentionally artefact-driven. A single request is transformed into a tasking order, feasibility events, a planning order, a confirmed pass schedule, execution outcomes, ingestion status, mission-level product references, and final EO product metadata/delivery references. This chain of artefacts enables end-to-end traceability, supports automation across planning and execution, and provides a structured basis for validation of both functional behaviour and operational timeliness.

B. UC-LEONSEGS-100: Request for Archived Images

UC-LEONSEGS-100 covers requests over a past time interval. The service layer translates user intent into catalogue search criteria and triggers discovery across available catalogues. Depending on the catalogue capabilities, discovery can be metadata-based or semantic-based. Once candidate products are identified, a production request is created to generate the requested EO service output. The processing component retrieves the selected inputs, executes the processing chain, publishes the resulting EO product with metadata, and notifies the end-user with a delivery reference.

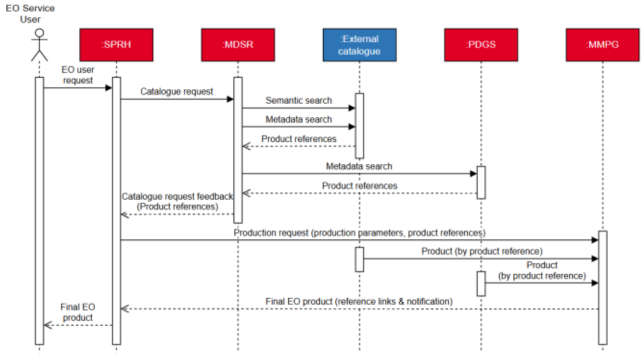

Fig. 5. UC-LEONSEGS-100 UML diagram (Request for Archived Images).

C. UC-LEONSEGS-300: Tasking via an External EO Product Provider

UC-LEONSEGS-300 addresses acquisitions fulfilled by an external EO product provider. After request capture, the multi-mission planning function selects a provider and issues a provider-specific planning request. The provider returns planning status and, once available, product references. The service layer uses these references to retrieve the external product (or a delivery link) and triggers final product generation and dissemination, optionally performing post-processing to deliver a consistent EO service output to the end-user.

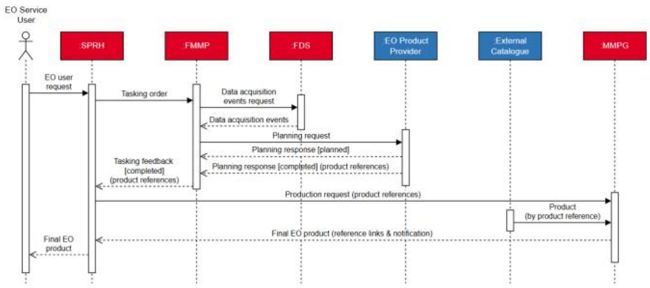

Fig. 6. UC-LEONSEGS-300 UML diagram (Tasking via an External EO Product Provider).

D. UC-LEONSEGS-400: GSaaS Monitoring and Control (M&C) Service

UC-LEONSEGS-400 validates GSaaS mission onboarding and subsequent mission operations through monitoring and control services. The GSaaS user initiates onboarding, which registers mission-specific configuration across the relevant subsystems (metadata, planning rules, orbital data, control procedures, and payload/ingestion configuration). After onboarding, the user can request planning activities that are validated against mission constraints and transformed into an operations plan that includes ground station passes. During execution, the orchestrator triggers scheduled procedures and the mission control subsystem performs uplink/downlink operations and monitoring. Operational reporting (including execution outcomes and telemetry-derived summaries) enables the GSaaS user to supervise the mission through automated and orchestrated platform services.

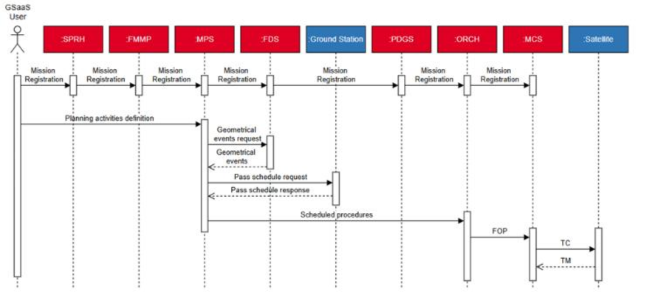

Fig. 7. UC-LEONSEGS-400 UML diagram (GSaaS Monitoring and Control (M&C) Service).

V. Conclusions

This paper presented LEONSEGS, a federated ground segment platform that combines Ground Segment as a Service (GSaaS) with automated EO service delivery to support end-to-end workflows across heterogeneous missions, external providers, and distributed catalogues. The architecture is structured into two cooperating segments: MOCPS, implementing the mission operations backbone (planning, scheduling, monitoring and control, orchestration, and payload data handling), and AMEOS, implementing the EO service layer (request brokering, cross-mission selection, discovery, including semantic retrieval, and multi-mission product generation). This separation of concerns enables automation across the planning–execution boundary while preserving an artefact-driven traceability model linking user intent to operational evidence and delivered EO products.

The end-to-end use case set defined for the Proof of Concept demonstrates how the platform operationalises these principles in representative scenarios. UC-LEONSEGS-200, the primary use case in this paper, validates the full lifecycle for internally operated missions: service request capture, feasibility inputs from flight dynamics, mission planning and ground contact scheduling, supervised execution (TM/TC), payload downlink and ingestion, and final EO product generation and dissemination. Complementary use cases extend coverage to archived retrieval (UC-100), provider-based tasking and retrieval (UC-300), and GSaaS onboarding plus monitoring and control service delivery (UC-400). Together, they provide a structured baseline for validating multi-mission interoperability and for identifying the interface contracts and operational artefacts that are critical to scale beyond single-mission ground segments.

As a TRL6-targeting prototype, LEONSEGS prioritises integration, interoperability, and automation patterns over full operational hardening. The work therefore focuses on architecture and interface design choices that enable repeatable onboarding and provider federation, rather than on mission-specific optimisation or hardware-level TT&C implementation details. The presented approach supports progressive evolution towards operational deployment by keeping the upstream user interaction stable while allowing heterogeneous fulfilment paths (internal missions, external providers, archival catalogues) to be selected according to feasibility and constraints.

Future work will focus on (i) expanding federation coverage to additional missions, catalogues, and ground station providers, (ii) strengthening standardisation and conformance profiles for tasking, scheduling, and catalogue access to further reduce integration effort, (iii) hardening security, observability, and governance functions required for multi-tenant service operation, and (iv) extending validation with quantitative performance indicators (e.g., turnaround times, automation coverage, scheduling efficiency) under representative operational conditions. These steps will support the transition from prototype demonstration to a more operationally robust GSaaS and EO service platform.

Acknowledgment

This work has received funding from the European Union's Horizon Europe programme under Grant Agreement No. 101082493 (LEONSEGS). The author acknowledges the contributions of the LEONSEGS consortium partners (GMV, PLUS, F6S, and AISTECH) to the definition, implementation, and validation activities underpinning the results reported in this paper.