Autonomous Federation of Earth Observation Constellations: An End-to-End Demonstrator based on the DOMINO Architecture

Cyrille De Lussy
Airbus Defence and Space
Toulouse, France
cyrille.de-lussy@airbus.com

Stéphane Derrien
Capgemini
Toulouse, France
stephane.a.derrien@capgemini.com

Marie Devant
Capgemini
Toulouse, France
marie.devant@capgemini.com

Jean-Loup Farges
DTIS, ONERA, Université de Toulouse
Toulouse, France
jean-loup.farges@onera.fr

Jonathan Guerra
Airbus Defence and Space
Toulouse, France
jonathan.guerra@airbus.com

Philippe Pavero
Airbus Defence and Space
Toulouse, France
philippe.pavero@airbus.com

Gauthier Picard
DTIS, ONERA, Université de Toulouse
Toulouse, France
gauthier.picard@onera.fr

Cédric Pralet
DTIS, ONERA, Université de Toulouse
Toulouse, France
cedric.pralet@onera.fr

Jakub Rezler
ITTI
Poznan, Poland
jakub.rezler@itti.com.pl

Raivis Skadiņš
Tilde
Riga, Latvia
Raivis.Skadins@Tilde.lv

ABSTRACT

As Earth Observation (EO) constellations proliferate, interoperability and resource optimization become critical challenges. This paper introduces the integration and end-to-end validation of the DOMINO-E federation layer agents, demonstrating how their orchestration enables autonomous Earth Observation. We detail the design and validation of its three core functional units: the Virtual Assistant Service (VAS) for natural language request definition; the Coverage Service (CS) for autonomous multi-mission task dispatching and dynamic re-dispatching; and the Satellite Communication and Resource Management Service (SCRMS) for optimized ground segment allocation. Validated via an End-to-End cloud-based testbed and a Sichuan flooding scenario involving the CO3D and Pléiades Neo constellations, the framework demonstrates a significant increase in operational responsiveness. Results confirm that the DOMINO-E Federative architecture successfully automates cross-constellation optimization, reducing area waste and ensuring seamless communication planning in complex EO missions.

KEYWORDS

Satellite Constellations, Federated Systems, Earth Observation, Autonomous Scheduling, DOMINO-E, Mission Management

ACM Reference Format:
Cyrille De Lussy, Stéphane Derrien, Marie Devant, Jean-Loup Farges, Jonathan Guerra, Philippe Pavero, Gauthier Picard, Cédric Pralet, Jakub Rezler, and Raivis Skadiņš. 2026. Autonomous Federation of Earth Observation Constellations: An End-to-End Demonstrator based on the DOMINO Architecture. In *Appears at the International Workshop on Autonomous Agents and Multi-Agent Systems for Space Applications (MASSpace-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., Paphos, Cyprus, May 2026*, IFAAMAS, 9 pages.

1 INTRODUCTION

Advancements in satellite performance-specifically the blurring of resolution boundaries between commercial and defense satellites-combined with New Space initiatives from privately funded actors, have transformed Earth Observation (EO) markets. This has led to worldwide competition impacting service providers, satellite manufacturers, and governmental users [4]. A shift toward smaller satellites and larger constellations has created new business opportunities, while the enhanced EO capabilities of these new constellations enable viable and sustainable ways to fulfill societal expectations for the global population [8]. However, despite the sophistication of present systems, significant room for innovation remains, particularly in the domain of Artificial Intelligence (AI).

The EO value chain is generally segmented into three categories [17]: (1) Upstream: Space and ground segment production and launches; (2) Midstream: Satellite operations, data processing, and archiving; (3) Downstream: Transforming data into value-added products and distributing them to end-users. Within the midstream segment, offline planning and scheduling-which consists in finding methods to schedule observation and upload/download tasks over a constellation-remain identified AI challenges [13]. An attractive

---

Appears at the International Workshop on Autonomous Agents and Multi-Agent Systems for Space Applications (MASSpace-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., S. Chien, G. Picard, I. Zilberstein (Chairs), May 2026, Paphos, Cyprus.

framework for addressing this challenge is the DOMINO architecture, a modular, service-oriented ground segment architecture designed to include multiple constellations [9].

Using the DOMINO architecture, a multi-agent federation layer, called DOMINO-E, has been developed to coordinate systems composed of independent EO missions. The goal of this federation is to allow clients requesting acquisitions of large areas to seamlessly access several satellite constellations and communication sites. This enables the composition and download of acquisitions in significantly reduced time compared to conventional, uncoordinated requests [5]. While previous research has studied the individual agents composing this federation layer in isolation [2, 3, 10–12, 14, 16, 18, 19], the specific contribution of this work is the design and validation of the *integration* of these agents within the federation layer. While pioneering works such as NASA's EO-1 SensorWeb [1] established the feasibility of autonomous science-event triggers (e.g., wildfire detection), they primarily focused on vertical integration between specific assets. In contrast, DOMINO-E addresses the challenge of horizontal federation across heterogeneous constellations. Furthermore, whereas the NASA New Observing Strategies Testbed (NOS-T) [6] provides a framework for evaluating such dynamic responses, DOMINO-E contributes a specific end-to-end architecture that combines natural-language intent (VAS) with cross-mission ground resource optimization (SCRMS), moving beyond event-triggers toward a user-centric service model.

Section 2 of this paper presents the design of the federation layer interacting with external agents, such as the mission centers of individual missions and Ground Station as a Service (GSaaS) provider systems. Sections 3, 4, and 5 detail the specific agents of the federation layer: respectively, the Virtual Assistant Service (VAS), the Coverage Service (CS), and the Satellite Communication and Resource Management Service (SCRMS). The integration of these agents into a testbed is presented in Section 6. Section 6.3 describes the demonstration scenario and its outcomes. Finally, concluding remarks are provided in Section 8.

2 ARCHITECTURAL APPROACH

This section presents the design and functional logic of the three primary services composing the federation layer. We describe the Virtual Assistant Service (VAS) for request management, the Coverage Service (CS) for task orchestration, and the Satellite Communication and Resource Management Service (SCRMS) for ground segment optimization.

The DOMINO architecture stems from the objective of making main interfaces within EO ground systems public and standardized [9]. The building blocks of the architecture, termed "dominoes", are standalone agents relying on their own infrastructure to deliver services, potentially across multiple missions. Each domino and its interfaces are natively designed for virtualized environments, specifically cloud-based infrastructures. Dominoes are categorized according to their functional roles: (i) Reactivity and monitoring; (ii) Mission and satellite command and control; (iii) Data management and processing; (iv) Multi-mission user and resource management. These modular units are combined to form a complete, flexible EO ground segment.

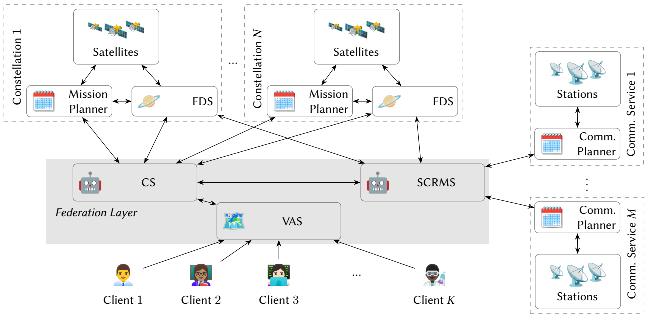

Figure 1: The DOMINO-E Federation Layer and its interactions within the ground segment architecture.

The federation layer, called DOMINO-E, addresses a complex multi-mission decision problem: how to partition a large area and assign these subdivisions to diverse missions to minimize acquisition time, ensure sufficient communication capacity, and maximize image quality-all while operating under uncertainty regarding future mission workloads. This decision process is triggered by new user requests or updates in acquisition statuses from the missions. The ecosystem involves several stakeholders: *clients* who request large-area imagery; *independent missions* (constellations) with their own planning and Flight Dynamic Service (FDS); and *communication services* (mission-owned stations or GSaaS). The global objective is to orchestrate these resources to fulfill client requests faster than uncoordinated, siloed operations [5].

Figure 1 illustrates the proposed federation layer. Fully compliant with the DOMINO architecture, this layer integrates three specific dominoes:

- **VAS (Virtual Assistant Service):** Dedicated to the management and interpretation of user requests.
- **CS (Coverage Service):** Dedicated to the orchestration and dispatching of observation tasks.
- **SCRMS (Satellite Communication and Resource Management Service):** Dedicated to the dispatching and booking of communication resources.

This federation layer acts as a seamless interface between clients and the technical complexities of multiple satellite constellations. By dividing large areas into sub-tasks that can be executed in parallel by distinct missions, the federation optimizes resource utilization and reduces the global load through the merging of overlapping requests.

The standard operational workflow of the federation layer is as follows:

(1) **Orbital Awareness:** The CS and SCRMS dominoes regularly poll FDS dominoes for up-to-date orbital parameters.
(2) **Request Interpretation:** Upon receiving a client request, the VAS interprets the natural language or structured input, derives technical observation requirements, and forwards them to the CS.
(3) **Observation Dispatching:** The CS coordinates with the individual Mission Planner dominoes to establish an agreed-upon global observation plan. This may involve iterative

negotiations or direct dispatching based on mission availability.

(4) **Communication Needs:** Once the observation plan is stabilized, the CS updates the operational needs for each satellite and requests corresponding contact slots from the SCRMS.
(5) **Contact Planning:** On a daily basis (or upon a significant update), the SCRMS computes an optimized contact plan.
(6) **Resource Booking:** The SCRMS submits booking requests to the relevant communication services (GSaaS or mission-owned).
(7) **Closing the Loop:** Upon successful booking, the SCRMS confirms the availability to the CS and, where applicable, the relevant Mission Planners, allowing for the finalization of the satellite's activity sequences.

3 VIRTUAL ASSISTANT SERVICE (VAS)

The Virtual Assistant Service (VAS) serves as the primary entry point for users within the DOMINO-E federation, providing a natural language interface to manage complex Earth Observation (EO) tasks. Unlike traditional forms or rigid command-line interfaces, the VAS allows users to define geographical areas, search product catalogs, and build multi-mission programming requests through intuitive dialogue. The VAS is justified even for expert users by its ability to synthesize cross-mission constraints into a single natural language query, reducing the need to manually navigate multiple service interfaces.

3.1 System Architecture and Base VA

The VAS is built upon a modular platform where all bots are derived from a **Base VA** framework. This framework provides standardized services for message exchange, dialogue management, and user authentication. The architecture is designed to be provider-agnostic, supporting integration with various communication channels such as the DOMINO-E User Access Service (UAS) web interface.

The core of the system is the **Dialogue Manager**, which operates according to a predefined *conversation scenario*. This scenario is structured as a directed graph of dialogue states, where transitions are triggered by detected user intents or system events. This state-machine approach ensures that the assistant maintains context-such as keeping track of a specified "Area of Interest" (AOI) while the user selects a sensor type.

3.2 Natural Language Understanding (NLU)

The NLU subsystem is responsible for converting raw text into structured data through two primary machine learning tasks:

- **Intent Detection:** The system classifies user utterances into functional categories (e.g., *SearchCatalogue*, *RequestAcquisition*). The classification is performed by a Convolutional Neural Network (CNN) architecture. User input is first transformed into 300-dimensional word embeddings using a *fastText* model. To improve model robustness, training data is synthetically augmented using LLMs like GPT-4 to generate semantically diverse examples, which are then manually curated by domain experts.
- **Named Entity Recognition (NER):** The system identifies domain-specific entities such as coordinates, dates, mission names, and sensor constraints. The VAS utilizes a hybrid NER approach: *regex-based models* are used for structured data like dates and coordinates, while *trainable models* are employed for semantic entities identified during testing and domain-specific annotations.

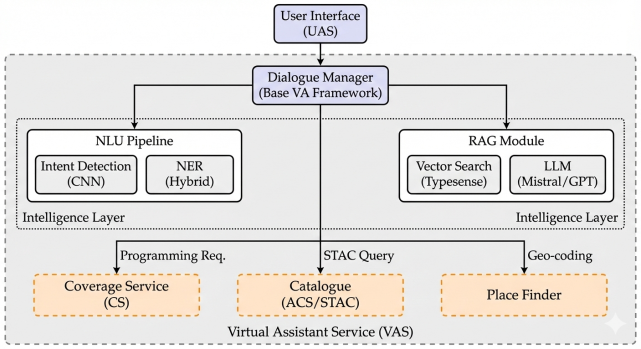

Figure 2: Functional components and integration architecture of the Virtual Assistant Service.

3.3 Hybrid Dialogue and RAG

To handle queries that fall outside the structured dialogue scenarios (e.g., "What is the resolution of Pléiades Neo?"), the VAS implements a **Retrieval-Augmented Generation (RAG)** module. This allows the assistant to answer open-ended questions using unstructured documentation stored in a local knowledge base.

The RAG process involves three stages:

(1) **Vectorization:** Knowledge base documents and user queries are converted into high-dimensional vectors using models such as Mistral or Multilingual Sentence BERT.
(2) **Retrieval:** The system retrieves the most relevant document chunks from a *Typesense* vector database based on cosine similarity.
(3) **Generation:** An LLM synthesizes a response by combining the retrieved technical context with the user's original query, ensuring the answer remains grounded in project-specific data.

3.4 Integration and External Services

The VAS acts as a technical bridge between the user and the other "dominoes" of the federation. It integrates with external systems via standardized APIs:

- **Archive/Catalogue Service (ACS):** The VAS translates user search parameters into *SpatioTemporal Asset Catalog (STAC)* API queries, allowing users to browse historical data from resources like the Copernicus Data Space Ecosystem.
- **Coverage Service (CS):** Once a user request is finalized (e.g., "Monitor the Sichuan region for 48 hours"), the VAS submits a technical *Programming Request* to the CS, which then handles the orchestration of satellite assets.
- **Toponym Search:** A Place Finder component is integrated to allow users to define geographical areas using natural names rather than just coordinates.

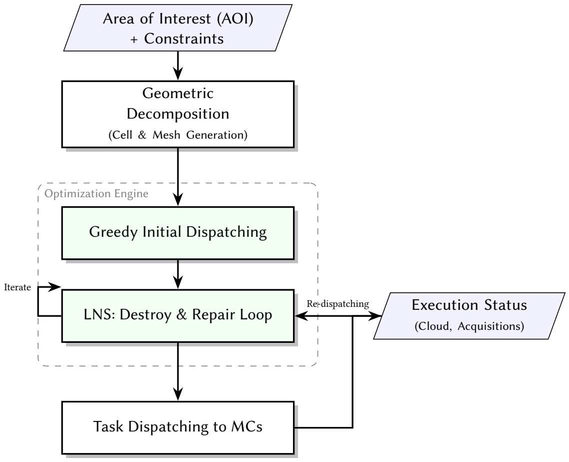

Figure 3: Coverage Service Workflow: from AOI input to autonomous re-dispatching based on mission feedback.

4 COVERAGE SERVICE (CS)

The Coverage Service (CS) is the orchestration core of the DOMINO-E federation layer. It is responsible for partitioning large geographical requests into manageable tasks and distributing them across available satellite missions. Several dispatch strategies are available in the CS. One of them addresses the "Mesh Dispatching" problem, optimizing the use of heterogeneous constellations to minimize acquisition time while accounting for system-specific constraints and execution uncertainties, following the workflow illustrated in Figure 3.

4.1 Spatial Decomposition: Meshes and Cells

To handle heterogeneous systems, the CS employs a dual-layer spatial decomposition approach:

- **Meshes:** Each mission s defines its own set of meshes M_s, aligned with a "world layer split" (typically North-South/East-West). Mesh sizes are dictated by the sensor's swath width and resolution.
- **Cells:** To ensure full coverage without gaps across varying mesh grids, the CS computes an atomic decomposition of the Area of Interest (AOI) into *cells*. A cell is defined as a maximum polygon where all internal points belong to exactly the same mesh for every available mission.

This cell-based approach provides a mathematical guarantee that covering all assigned cells leads to a complete observation of the requested area, regardless of the differing mesh geometries used by the federated missions.

4.2 Optimization Objectives

The dispatching problem is modeled using a coarse-grain representation of satellite capacities. Satellite passes are divided into temporal *slots*, each capable of observing a maximum number of meshes based on the satellite's agility. The CS optimizes a lexicographic objective function:

1. **Minimize Completion Time (T_{max}):** Ensuring the last required image is acquired as early as possible.

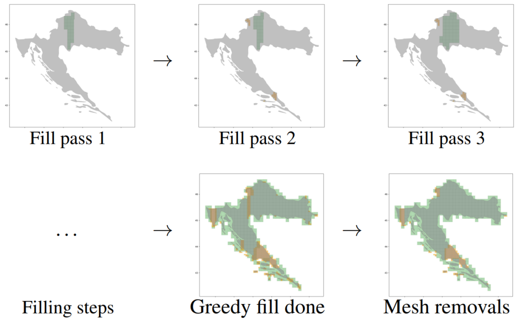

Figure 4: Iterations of the greedy search procedure selecting meshes step-by-step to cover the area of interest, for a scenario involving two systems (system 1 using the green meshes and system 2 using the yellow meshes).

2. **Maximize Mesh Grouping:** Reducing satellite maneuver overhead by preferring contiguous "strips" of meshes within a single pass.
3. **Minimize Area Wastage:** Avoiding overlaps between different missions and redundant observations outside the AOI.

4.3 Large Neighborhood Search (LNS) Algorithm

Given that the dispatching problem is NP-hard [14], the CS utilizes a **Large Neighborhood Search (LNS)** metaheuristic to explore the solution space efficiently.

Initial Heuristic Search. The process begins with a *greedy heuristic* that builds an initial plan. It iterates through available satellite passes chronologically, selecting meshes based on a scoring system that balances geographical utility (cells covered) and the potential for grouping with already selected meshes in the same orbit.

Destroy and Repair Cycles. The LNS then refines this initial plan through iterative cycles:

- **Destroy:** A significant portion of the current assignment (e.g., 15–30% of meshes) is removed. The algorithm employs targeted removal of meshes at the "frontiers" of mission assignments to allow for better cross-mission optimization.
- **Repair:** The removed meshes are re-inserted using the greedy heuristic, allowing the system to escape local optima and find more efficient task distributions.

The steps are illustrated in Figure 4.

4.4 Dynamic Re-dispatching and Uncertainty

A key innovation of the CS is its **autonomous re-dispatching mechanism**. Because the federation's model is a coarse approximation and factors like cloud cover or mission-level planning changes can invalidate acquisitions, the CS regularly re-optimizes the plan. During each re-dispatching cycle (e.g., once every 24 hours), the CS integrates feedback from the Mission Centers regarding successful acquisitions. It removes successfully covered cells from the problem and re-runs the LNS for the remaining AOI, potentially reassigning tasks from a lagging mission to one with better upcoming availability or weather conditions.

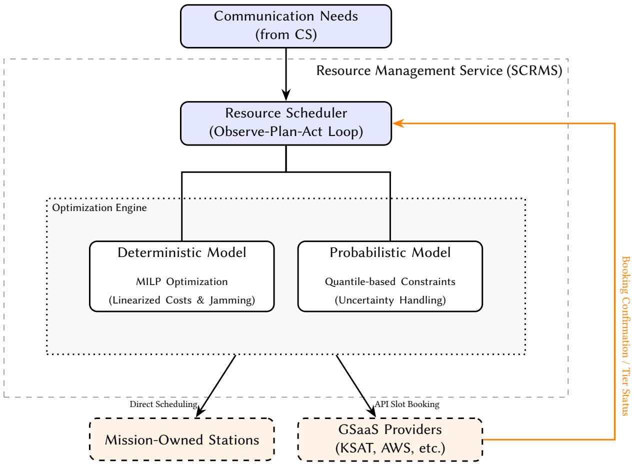

Figure 5: SCRMS Architecture: Integrating deterministic MILP and probabilistic quantile-based models for hybrid ground station scheduling.

5 SATELLITE COMMUNICATION AND RESOURCE MANAGEMENT SERVICE (SCRMS)

The SCRMS optimizes the ground segment resources for the federation, managing the scheduling and booking of satellite-to-ground contacts across both mission-owned and third-party Ground Station as a Service (GSaaS) providers.

5.1 Operational Context

The service's objective is to satisfy the communication needs of each satellite-defined by frequency bands and contact durations-while minimizing costs and avoiding interference.

Needs are categorized as *general* or *localized* (area-specific), specifying the required visibility windows for S-band (uplink/downlink) and X-band (high-speed data downlink). The optimization must account for: (i) **Cost Models:** agreements include *pay-per-pass*, *pay-per-duration*, and *commitment-based* tiers where a volume of contacts is prepaid and extra bookings incur fixed costs; (ii) **Conflict and Jamming:** Conflicts arise when two satellites attempt to access the same station/band simultaneously. Jamming occurs between nearby stations on the same site based on the angular separation of the station-satellite vectors.

The SCRMS operates on an *Observe-Plan-Act* loop [15], triggered daily or by emergency updates:

1. **Observe:** Gathers communication needs and computes potential visibility windows using the Orekit space dynamics library [7].
2. **Plan:** Solves 10 independent daily multi-criteria optimization problems over a 10-day horizon, ensuring fulfillment of needs while minimizing costs and interference.
3. **Act:** Manages the asynchronous booking of contacts through GSaaS APIs, respecting tier-based booking windows.

5.2 Optimization Modeling

The problem involves N satellites, each with L_i potential contacts and K_i needs. Interference between contact l of satellite i and contact m of satellite j is represented by parameter $b_{i,l,j,m}$. The full description of these models are available in [18].

Deterministic Approach. The selection is modeled using binary variables $x_{i,l} \in \{0, 1\}$. While the cost for commitment-based agreements and the jamming criterion J_{in} are naturally non-linear, the problem is linearized to employ Mixed-Integer Linear Programming (MILP) via the Google OR-Tools solver. The non-linear interference term $\sum b_{i,l,j,m} x_{i,l} x_{j,m}$ is linearized using auxiliary variables $y_{i,l,j,m}$ and standard constraints ($y \leq x_{i,l}, y \leq x_{j,m}, y \geq x_{i,l} + x_{j,m} - 1$). Similarly, commitment costs are linearized by introducing variables to represent the volume of contacts above the prepaid threshold Y_s. Experiments show that the MILP approach significantly outperforms round-robin or greedy heuristics in both solution quality and computation time [19].

Probabilistic Approach. To account for the uncertainty of GSaaS contact acceptance, we introduce a probability of success $p_{i,l}$. Simply optimizing expected values would result in failing to satisfy approximately half of the requests. Consequently, we incorporate the standard deviation of need fulfillment into the constraints:

$$\sum_{l} d_{i,k,l} p_{i,l} x_{i,l} - q\Big(\sum_{l} d_{i,k,l}^2 p_{i,l} x_{i,l} (1 - p_{i,l})\Big)^{\frac{1}{2}} \geq D_{i,k}$$

where $D_{i,k}$ is the required time for need k and q controls the fulfillment quantile. By linearizing this standard deviation term, the problem remains solvable via MILP. Numerical results with $q = 2$ demonstrate that this probabilistic approach reduces the communication time deficit from 10% in the deterministic model to 6% [18].

6 TEST-BED INTEGRATION AND DEMONSTRATION

The operational capabilities of the DOMINO-E architecture were validated through a comprehensive End-to-End demonstrator. The demonstration utilizes a shared cloud testbed where the Virtual Assistant Service (VAS) operates in a dedicated namespace, while the Coverage Service (CS) and the Satellite Communication and Resource Management Service (SCRMS) are deployed together to ensure low-latency orchestration.

6.1 Test-bed Architecture

As part of the DOMINO-E initiative, all the developed dominoes have been integrated together. A common platform, called the Testbed, has been created for this purpose (see Figure 6). Hosted on a Google Cloud Platform located in Europe, the biggest challenge of this platform building was to manage high security standards and multinational accesses.

Once provisioned, this platform has been used by all the dominoes, for their own validation, and for a common validation within the framework of an End-to-End test, involving all the dominoes.

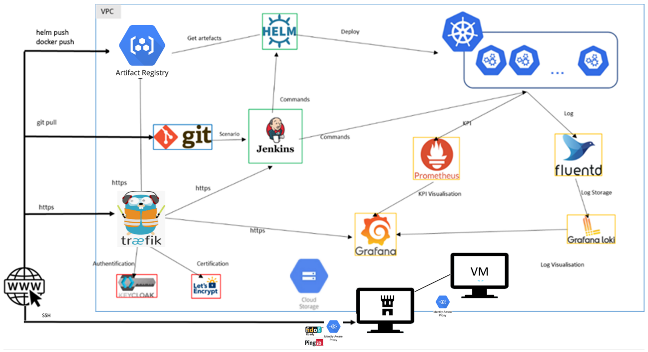

Figure 6: Test-bed architecture and technologies

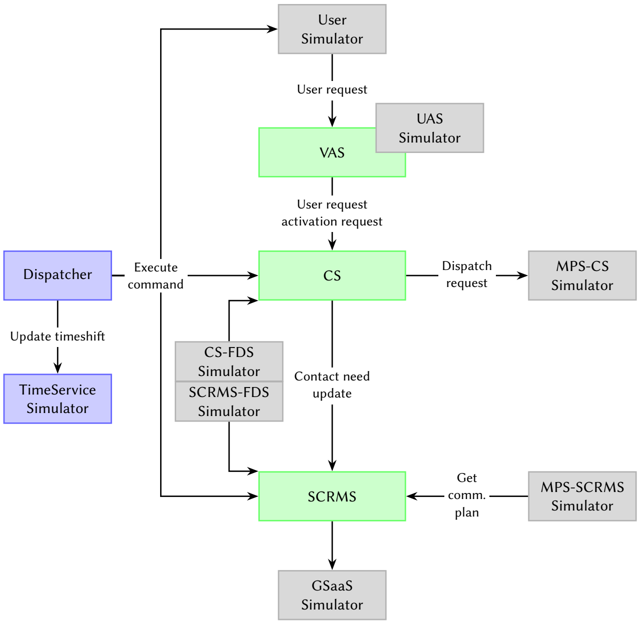

Figure 7: Interactions within the test-bed: dominoes in green, Simulators in grey, and Dispatcher/Scheduler in blue.

6.2 End-to-End Scenario

The underlying story behind the End-to-End scenario is to react to a large scale catastrophic event such as a flood or a hurricane (in the region of Sichuan in our example), by managing the deposit and the planing of imagery request of this area. The End-to-End scenario has been build to trigger interactions between the Virtual Assistant, the Coverage Service and the SCRMS: The User request deposit goes through User Access Service using the Virtual Assistant Service, is dispatched to the constellations by the Coverage Service. This implies a change in download needs, which triggers the SCRMS to change the communication plan. All these steps are displayed in Figure 7, where the inter-component interaction involves a closed-loop feedback from VAS to CS.

The End-to-End validation of the DOMINO-E federation layer was conducted using a high-fidelity simulation environment designed to mimic real-world space operations. The simulation environment provides an integrated testbed where the various "Domino" services (VAS, CS, and SCRMS) are deployed within shared cloud namespaces. To represent the space segment, the simulation incorporates digital twins of the CO3D and Pléiades Neo constellations, allowing for the modeling of realistic orbital dynamics and instrument constraints (FDS), and of the mission planning centers (MPS). The ground segment is simulated through a federated network model that includes both mission-owned stations and virtualized GSaaS (Ground Station as a Service) providers, such as KSAT. Furthermore, a User Simulator was utilized to perform stress tests and rapid-fire validation runs, bypassing the manual chat interface to evaluate the system's algorithmic performance under high-frequency request loads.

6.3 Demonstration

The End-to-End demonstration is available as a video, here: https://www.youtube.com/watch?v=I2biwBIydZ0.

User Interaction and Request Definition. The workflow begins at the **User Access Service (UAS)** interface, where the user initiates a request via the natural language chat interface. In the demonstration scenario, the user defines an observation area over **Sichuan, China**, a region frequently prone to flooding. The VAS guides the user through defining the period of interest and fine-tuning mission parameters for two federated constellations: **CO3D** and **Pléiades Neo**.

Autonomous Coverage Planning and Dispatching. Once the user validates the request, the VAS transfers the technical parameters to the CS. The system's autonomous planning involves several key steps:

- **Decomposition:** The CS divides the target area into multiple independent programming requests and displays them on a global map for user follow-up.
- **Initial Dispatching:** Tasks are distributed between the available constellations, and a progress graph provides an estimated completion timeline.
- **Dynamic Re-dispatching:** The demonstrator showcases a scenario where the system automatically triggers a re-dispatch. For instance, tasks originally assigned to Pléiades Neo are canceled and resubmitted to CO3D to optimize the total completion time based on real-time mission progress.

Communication and Resource Optimization. Following the re-dispatch of observation tasks, the CS communicates updated contact needs to the **SCRMS**. The SCRMS automatically aligns the communication plan with the new coverage requirements, adjusting the number of booked ground station slots to ensure efficient data downlink.

Validation Results. The final state of the demonstration confirms the stability of the federation layer. The results show that: (i) The system achieved approximately 20% total coverage within the simulated window, successfully handling task cancelations and re-assignments. (ii) The SCRMS maintained a close alignment between "required" and "booked" contact durations, confirming that the communication plan remained optimized even as the observation plan evolved. All operations, from initial request to final downlink booking, are executed automatically, demonstrating the high level of autonomy achieved by the DOMINO-E federation layer.

7 OUTCOMES

This section details the performance assessment of the individual services within the DOMINO-E federation. The evaluation focuses on operational efficiency, algorithmic accuracy, and resource consumption. We consider two operational scenarios. The *Occitania* scenario covers a 72,000 km² region in southern France over a 48-hour horizon, while the *France* scenario involves a nationwide coverage request to test scalability. All experiments were conducted on a Linux-based server with a 3.6 GHz Intel Core i7 processor and 32GB RAM.

7.1 Virtual Assistant Service (VAS) Outcomes

The performance of the VAS was evaluated based on its ability to streamline user interactions compared to traditional Graphical User Interfaces (GUIs). Assessment metrics focused on task completion time, dialogue accuracy, and system robustness.

Operational Efficiency. The VAS demonstrated a significant reduction in the time required to perform core Earth Observation (EO) tasks. Comparative tests against the standard User Access Service (UAS) manual interface revealed the following improvements:

- **Catalogue Search:** Utilizing the VAS for area-of-interest selection reduced search times from 131 seconds to 83 seconds on average.
- **Programming Request (PR) Creation:** For expert users, the creation of PRs was reduced from a baseline of 247–340 seconds (UAS manual) to 102–211 seconds when assisted by the VAS.
- **High-Speed Processing:** Automated user simulations achieved task completion in as little as 28 seconds, highlighting the efficiency of the NLU-driven backend.

Dialogue Quality and LLM Reasoning. The hybrid NLU architecture, combining structured dialogue management with LLM-based reasoning, was assessed for response relevance and accuracy.

- **Response Accuracy:** 64.7% of responses were classified as fully correct and context-aware.
- **Advanced Reasoning:** 20.2% of interactions were rated as "superior," where the assistant successfully managed complex follow-up questions, relative date handling (e.g., "two weeks after"), and parameter disambiguation.
- **Failure Modes:** Approximately 12.1% of responses were inadequate, primarily occurring in excessively long dialogue chains where context retention reached its limit.

The 64.7% accuracy primarily stems from NLU intent classification errors in complex multi-parameter queries; however, the 'Human-in-the-loop' design allows users to correct these parameters before submission.

Table 1: Key Performance Indicators (KPIs) for the VAS

Metric	Manual Interface	VAS Assisted
Avg. Image Search Time	131 s	83 s
Avg. PR Creation Time	293 s	157 s
System Accuracy	N/A	84.9%
Memory Footprint (per instance)	N/A	~165 MB

Resource Consumption and Stability. Monitoring of the VAS during peak loads confirmed the service's stability. In concurrent automated testing scenarios, memory consumption for the core processing node remained stable at approximately 165 MB. The system showed high resilience to non-ideal inputs, including spelling errors and incomplete coordinate formats.

The evaluation confirms that the VAS effectively lowers the barrier to entry for complex multi-mission programming, enabling intuitive access to the federation's underlying optimization services.

7.2 Coverage Service (CS) Outcomes

The evaluation of the CS centers on "Area Waste" (the ratio of redundant or unnecessary acquisition area to the target area) and "Completion Time" across different algorithmic use cases: manual dispatch (UC1a), static dispatch (UC1b), dynamic re-dispatch with planning estimation (UC1c), and Large Neighborhood Search (UC2).

Operational Efficiency and Completion. The CS demonstrated the capability to coordinate multiple constellations to achieve high completion rates for large-scale requests.

- **Scenario Performance:** In the medium-scale *Occitania* scenario, the dynamic re-dispatching (UC1c) achieved a 54.63% completion after 10 days of simulated operations.
- **Scalability:** The service successfully handled the *France* scenario (hundreds of thousands of km²), managing the increase in dispatched programming requests proportionally to the area size.
- **Redispatching Efficiency:** Dynamic redispatching was shown to adapt to real planning progression, reducing the estimated time to full completion by several days compared to static methods (e.g., reaching full coverage estimates by 04/07/2034 instead of 08/07/2034 in the Occitania test starting on 01/06/2034).

Algorithmic Accuracy and Area Waste. A key performance indicator for the CS is the minimization of "Area Waste" during the mesh decomposition and dispatch phase.

- **Dispatch Precision:** The Large Neighborhood Search (UC2) significantly outperformed other methods in terms of precision, taking the actual splitting of mission chains as input. This resulted in an **Area Waste of 20.08%** for the loaded Occitania scenario, compared to over 50% for standard decomposition methods (UC1b/c).
- **Mission Optimization:** When taking mission chain self-optimization into account, the waste for UC1c dropped from 50.78% to a more manageable 22.31%, validating the benefit of federated feedback.

Computational Performance. The CS maintains efficient computation times, even when managing complex redispatching logic.

Table 2: Key Performance Indicators (KPIs) for the CS

Metric	Static Dispatch	LNS Dispatch
Coverage Completion (10 days)	32.6%	26.46%
Raw Area Waste	50.16%	20.08%
Waste after Mission Chain Opt.	21.99%	15.43%
Redispatch Time (s)	N/A	51.85

- **Computation Speed:** Initial dispatch for medium areas averaged approximately 70.65 seconds.
- **Dynamic Updates:** Subsequent redispatches were faster (37.38 seconds average), as they leveraged the previous state as an input and dealt with smaller remaining areas.
- **Resource Usage:** The number of external calls remained balanced, with dynamic use cases successfully managing thousands of interactions with mission chains to track status and progress.

The outcomes confirm that the CS effectively bridges the gap between high-level user requests and low-level mission scheduling, providing an autonomous layer that optimizes global coverage while minimizing resource redundancy.

7.3 Satellite Communication and Resource Management Service (SCRMS) Outcomes

The SCRMS was evaluated on its ability to satisfy multi-mission communication needs while optimizing ground station (GS) slot allocation. The assessment compared a "Minimum Configuration" (mission-owned sites only) against the "Federated Configuration" (mission-owned plus GSaaS providers).

Fulfillment of Communication Needs. The integration of a federated GS network significantly improved the system's ability to meet high-bandwidth requirements:

- **X-Band Capacity:** In the minimum configuration, owned sites were often insufficient to meet the peak data downlink demands of the CO3D and Pléiades Neo constellations. The federated approach, by booking external GSaaS slots, consistently achieved 100% fulfillment of X-band needs.
- **S-Band Reliability:** S-band needs for command and control were satisfied across all scenarios, with the scheduler successfully managing priority levels for critical telemetry links.

Resource Optimization and Cost. The SCRMS demonstrated high efficiency in resource allocation through its dual optimization logic:

- **Grouping Efficiency:** The scheduler achieved significant "Grouping" of contacts, bundling multiple acquisitions into single ground station visibility windows to reduce antenna slewing and operational overhead.
- **Cost Management:** While the minimum configuration has zero external cost, it suffers from data latency. The federated configuration utilized the cost-per-pass and commitment models to find a Pareto-optimal balance between data freshness and booking expenditure.

Table 3: Key Performance Indicators (KPIs) for the SCRMS

Metric	Owned Stations Only	Federated GSaaS
X-Band Fulfillment Rate	64.2%	100%
Avg. Data Latency	High	Low
Peak Memory Usage	~450 MB	~1 GB
CPU Cores (Peak)	2	4

Computational Stability and Hardware Usage. The service maintained a modest hardware footprint even when managing complex multi-day horizons:

- **Memory Footprint:** Memory usage of the SCRMS Docker instance peaked at approximately 1 GB during the solving phase of large scenarios.
- **CPU Performance:** The solver reached a peak of 4 cores during the initial optimization of the 10-day horizon, stabilizing at 1 core for subsequent re-optimization and acting loops.
- **Acting Latency:** The automated "Act" loop, responsible for API interactions with GSaaS providers, maintained subsecond latency for slot booking confirmations.

The results validate the SCRMS as a critical enabler for the federation, transforming the "Communication Needs" generated by the Coverage Service into an executable and cost-efficient ground segment schedule.

8 CONCLUSION

The DOMINO-E project has successfully demonstrated the feasibility and operational benefits of a multi-mission federation layer for Earth Observation. By moving beyond traditional, fragmented mission management, the project has established an autonomous ecosystem-the "DOMINO" architecture-that effectively bridges the gap between complex user requirements and heterogeneous satellite constellations.

The results obtained across the three core services-VAS, CS, and SCRMS-consistently validate the project's objectives:

(i) **Accessibility:** The VAS reduced tasking time by 40% via natural language, democratizing access for non-experts.
(ii) **Efficiency:** The CS utilized LNS algorithms to reduce area waste to 20%, significantly optimizing large-scale monitoring compared to static methods.
(iii) **Scalability:** By transitioning to a GSaaS model, SCRMS achieved 100% fulfillment of X-band communication needs.

The "Closed-Loop" capabilities showcased in the End-to-End demonstration highlight the system's ability to autonomously redispatch tasks in response to real-world uncertainties. Ultimately, DOMINO-E provides a scalable framework for future European initiatives, establishing a foundation for a more responsive and efficient space segment.

ACKNOWLEDGMENTS

This work is part of the DOMINO-E project which received funding from the European Union's Horizon Europe Programme for Research and Innovation under Grant Agreement n°101082230.

REFERENCES

[1] S. Chien, B. Cichy, A. Davies, D. Tran, G. Rabideau, R. Castano, R. Sherwood, S. Nghiem, R. Greeley, T. Doggett, V. Baker, J. Dohm, F. Ip, D. Mandl, S. Frye, S. Shuman, S. Ungar, T. Brakke, L. Ong, J. Descloitres, J. Jones, S. Grosvenor, R. Wright, L. Flynn, A. Harris, R. Brakenridge, and S. Cacquard. 2006. An autonomous earth observing sensorWeb. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, Vol. 1. 8 pp.–. https://doi.org/10.1109/SUTC.2006.1636174

[2] Cyrille De Lussy and Jonathan Guerra. 2024. Domino-E Coverage Service. Poster to the 9th European Mission Operations Data System Architecture Workshop (ESAW). Darmstadt, Germany.

[3] Cyrille de Lussy, Jonathan Guerra, Vivien Nguyen, Gauthier Picard, Cédric Pralet, Filipo Studzinski Perotto, Stéphane Derrien, Matthieu Vansteene, Corentin Roux, and Jean-François Vinuesa. 2024. Domino-E coverage service - A flexible, smart automated tool for multi-mission federation. In *75th International Astronautical Congress - Earth Observation Symposium*.

[4] Gil Denis, Alain Claverie, Xavier Pasco, Jean-Pierre Darnis, Benoît de Maupeou, Murielle Lafaye, and Eric Morel. 2017. Towards disruptions in Earth observation? New Earth Observation systems and markets evolution: Possible scenarios and impacts. *Acta Astronautica* 137 (2017), 415–433.

[5] Jean-Loup Farges, Filipo Studzinski Perotto, Cédric Pralet, Gauthier Picard, Cyril de Lussy, Jonathan Guerra, Philippe Pavero, and Fabrice Planchou. 2024. Going beyond mono-mission earth observation: Using the multi-agent paradigm to federate multiple missions. In *23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS-24)*. International Foundation for Autonomous Agents and Multiagent Systems, 2674–2678.

[6] Paul T. Grogan, Hayden C. Daly, Matthew S. Brand, and Jerry J. Sellers. 2021. New Observing Strategies Testbed (NOS-T) Architecture: Evaluating Dynamic Response to Emergent Events. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. 1470–1473. https://doi.org/10.1109/IGARSS47720.2021.9555131

[7] maisonobe, Bryan Cazabonne, Evan Ward, Romain Serra, Maxime Journot, Sébastien Dinot, Guilhem Bonnefille, Thomas Neidhart, Luc Maisonobe, Clément Jonglez, Mark Rutten, yjeand, Julio Hernanz, lirw1984, Vincent Cucchietti, Andrew Goetz, Guiuux, gaetanpierre0, Lars Næsbye Christensen, Alberto Fossà, jvalet, Alberto Ferrero, gabb5, liscju, Christopher Schank, Tanner Mills, plan3d, Vyom Yadav, Shiva Iyer, and Petrus Hyvönen. 2024. *CS-SI/Orekit: 12.2*. https://doi.org/10.5281/zenodo.13950582

[8] Salvo Marcuccio, Silvia Ullo, Marco Carminati, and Olfa Kanoun. 2019. Smaller satellites, larger constellations: Trends and design issues for earth observation systems. *IEEE Aerospace and Electronic Systems Magazine* 34, 10 (2019), 50–59.

[9] Daniel Marcell Novak, Amina Annane, Régis Baillard, Alain Berry, Cédric Brandon, Vincent Desormeau, Sylvain Gaudan, Julien Joyeux, Stan Kaethler, Charlie Madier, Olivier Melet, and Yann Roux. 2022. Future ground segments with standardized interfaces: the DOMINO-X project. In *73rd International Astronautical Congress*.

[10] Philippe Pavero. 2024. Domino-E Satellite Communication and Resource Management Service. Poster to the 9th European Mission Operations Data System Architecture Workshop (ESAW). Darmstadt, Germany.

[11] Philippe Pavero, Jean-Loup Farges, Gauthier Picard, Jakub Rezler, and Jean-François Vinuesa. 2025. Satellite Communication Management Domino, for Constellation and Ground Station as a Service Interconnection. In *76th International Astronautical Congress - Earth Observation Symposium*.

[12] Philippe Pavero, Fabrice Planchou, Jean-Loup Farges, Gauthier Picard, Filipo Perotto, Jakub Rezler, and Jean-François Vinuesa. 2024. Satellite communication management Domino, for constellation and ground station as a service interconnection. In *75th International Astronautical Congress - Earth Observation Symposium*.

[13] Gauthier Picard, Clément Caron, Jean-Loup Farges, Jonathan Guerra, Cédric Pralet, and Stéphanie Roussel. 2021. Autonomous Agents and Multiagent Systems Challenges in Earth Observation Satellite Constellations. (2021).

[14] Cédric Pralet, Gauthier Picard, Cyrille de Lussy, and Jonathan Guerra. 2025. Mesh Dispatching for Area Coverage using Several Earth Observation Systems. In *IWPSS-International Workshops on Planning and Scheduling for Space*.

[15] Matthew Roberts. 2002. Artificial Actors for Real World Environments. In *2002 AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*. Association for the Advancement of Artificial Intelligence, 87–94.

[16] Raivis Skadiņš, Evita Korņējeva, and Modris Stūrmanis. 2015. *D6.5 – Demonstration of the VAS*. EU project Technical Report D29. Tilde SIA.

[17] Wasanchai Vongsantivanich, David LX Ho, and Quentin Verspieren. 2018. Using space for disaster management in emerging space states: A critical assessment. (2018).

[18] Hénoïk Willot, Jean-Loup Farges, Gauthier Picard, and Philippe Pavero. 2025. Deterministic and Probabilistic Decision Models for GSaaS-based Satellite Communication Resource Management. In *IWPSS-International Workshops on Planning and Scheduling for Space*.

[19] Hénoïk Willot, Jean-Loup Farges, Gauthier Picard, and Philippe Pavero. 2025. Satellite communication resources management in a earth observation federation of constellations. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, 235–251.