

SpaceOps-2025, ID # 468

Constellation Mission Tasking
Matthew Scott^{a*}, Gergana Belcheva^b, Vasil Nedelchev^c, Boyan-Nikola Zafirov^d

^a *EnduroSat SAS, Toulouse, France, matthew.scott@endurosat.com*

^b *EnduroSat AD, Sofia, Bulgaria, gergana.belcheva@endurosat.com*

^c *EnduroSat AD, Sofia, Bulgaria, vasil.nedelchev@endurosat.com*

^d *EnduroSat AD, Sofia, Bulgaria, boyan.zafirov@endurosat.com*

* Corresponding Author

Abstract

The burgeoning demand for Earth observation data necessitates increasingly sophisticated and efficient satellite mission planning and tasking systems. In response, EnduroSat is developing an advanced mission tasking service as part of its upcoming Earth observation satellite constellations. This service is fully integrated into EnduroSat's cloud-based mission operations platform, SpaceOps, delivering a seamless and automated workflow from user requests to satellite mission execution.

The mission tasking service is designed to streamline the entire process of satellite acquisition planning. Beginning with the submission of a user request, the service rapidly identifies acquisition opportunities across the constellation. By leveraging advanced algorithms and real-time constellation data, the service can determine the optimal opportunities that align with the desired observation parameters and geospatial locations.

Once potential acquisition opportunities are identified, the mission tasking service converts these into detailed satellite mission plans. This process factors in request prioritization and satellite availability, ensuring that higher-priority and time-sensitive tasks are addressed promptly. The automated workflow eradicates the need for manual intervention, significantly reducing operational latency and enhancing responsiveness.

Key technological enablers, such as satellite telemetry integration and cloud computing, will be discussed to highlight their roles in solving potential challenges and solutions associated with implementing such an automated workflow on a large constellation scale. Issues such as request collision management, dynamic priority adjustments, and real-time re-tasking in response to new requests will be addressed.

In conclusion, EnduroSat's mission tasking service mark* into the SpaceOps platform, the service offers users a streamlined, responsive, and reliable means of requesting Earth observation data, thereby fostering greater accessibility and utilization of satellite resources. Our paper aims to contribute to the space operations community by sharing the innovative strategies and technologies underpinning this service, encouraging further advancements and collaborations in Earth observation mission planning.

Keywords: (Mission Tasking, Automation, Optimization, Cloud Architecture)

1. Introduction

Recent trends in miniaturization, cost reduction, and increased capabilities of SmallSats have led to proliferated Low Earth Orbit (LEO) constellations. Amidst this rapid evolution, automated mission planning and tasking systems are emerging as vital components of successful satellite operations. Tasking a constellation of EO satellites requires sophisticated workflows, which must be intelligent, scalable, and responsive to dynamic priorities.

EnduroSat is developing an advanced mission tasking service that integrates seamlessly within EnduroSat's SpaceOps mission operations platform, delivering a fully automated workflow—from the initial user request to mission execution onboard the satellite itself.

This paper focuses on the challenges and solutions associated with implementing EnduroSat's constellation mission tasking. We highlight the importance of real-time re-tasking and explore the role of key technological enablers in the cloud-based environment that supports large-scale constellation operations. We present a thorough exploration of technical approaches that improve speed, reliability, and efficiency in satellite mission planning as well as addressing some of the future challenges such as the need for inter-operability with other systems and rationalizing tasking across multiple different types of sensors.

2. Use cases

2.1 Core Functionality

Accurate modeling of satellite capabilities and the request-handling algorithmic engine form the core of the mission planning system. From instrument fields of view (FOVs) to real-time constraints, each subsystem and operational parameter must be abstracted into mathematical models that support objective, constraint-aware optimization. The current generation of the mission tasking for EnduroSat is focused on optical earth observation missions, in anticipation of the Balkan Constellation [1], but has been designed to take varying different missions into account in the future.

2.2 System Modelling

The modelling and propagation of the orbit itself is the fundamental way to describe the capabilities of the constellation by obtaining revisit of the satellites over the area of interest. The field of view of the sensor will also be modelled. The attitude of the satellites is also modelled, and this should include a fairly accurate representation of the different control modes that will be used on the satellites during the imaging campaigns to correctly predict the swath shape that will be produced by a satellite. Advanced attitude control modes include pitch corrections to provide forward-motion-correction (FMC) during the fly over and yaw compensation to offset the Earth's rotation while imaging. The instruments will also have different modes – a snapshot mode where FMC is applied over a given tile to be imaged and a push-broom mode or screening mode where the sensor collects the data permanently for a large proportion of the orbit.

A second driver for the tasking of the system is the latency to send commands and download the data from the satellite through the ground station network. The Mission Tasking component will need to model the contacts with the ground stations. As many EnduroSat missions rely on 3rd party ground station providers it will also need to retrieve the availability of the ground stations in the various networks available to a mission through SpaceOps [2].

The constraints of the satellite in terms of onboard data capacity, satellite duty cycles and link budgets or data budgets need to be modelled by mission tasking. To avoid propagating detailed subsystem models of the satellite these will be obtained from the EnduroSat digital twin and then populated in the mission tasking in the form of configuration. In this way detailed information such as the time to transition between ADCS modes to be available for imaging and a more precise power model can be converted into simplified budgets for the mission tasking.

The other factor that will affect the tasking is the local environmental conditions that will affect the data being collected. For optical imaging this will include the lighting conditions and the weather conditions at the area of interest. Lighting conditions can be generally assessed from the orbital propagator as the relative positions of the satellite, earth and sun are known. The cloud coverage can be assessed from a pre-configured statistical climate model and then refined by using real-time weather forecasting API's.

2.3 Request Handling

Request handling relies heavily on an intelligent mission plan optimization process, where user requests—often specifying an area of interest—must be broken into discrete achievable tasks. Because a large geographic region may not be imaged in a single pass by a single satellite, the optimizer must divide the request into multiple, smaller image footprints. This segmentation may then be assigned to different satellites in the constellation, each with unique coverage windows, orbital paths, and resource constraints. Algorithms such as mixed-integer linear programming, genetic algorithms, or constraint programming can tackle these complexity layers by aiming to maximize image coverage while respecting onboard power, memory capacity, and downlink schedules.

When the area-based request is large or urgent, multiple satellites working in tandem can reduce turnaround time, thereby improving responsiveness and overall performance. The optimization engine must therefore determine how tasks are allocated across different spacecraft, evaluating the relative positions of each satellite, potential overlaps in

coverage, and available imaging modes (e.g., stripmap vs. snapshot). By distributing imaging tasks across the constellation, the system improves spatial coverage and mitigates single-satellite bottlenecks, leading to higher responsiveness to user needs.

Prioritization of requests is typically integrated into the optimization loop, ensuring that high-value, time-critical, or emergency tasks take precedence. A variety of ranking schemes exist, ranging from straightforward first-come-first-served to more advanced weighted-sum techniques that combine user-assigned priority levels, request deadlines, and resource impact. Some systems implement dynamic weighting methods, enabling real-time adjustments when conditions change—such as unexpected weather patterns or urgent requests triggered by disaster events.

Rescheduling is crucial for maintaining mission agility. As new requests arrive or constraints shift (e.g., orbit changes, payload anomalies), the system must adapt. Rolling-horizon scheduling or iterative planning helps revise earlier decisions without discarding the entire plan. By re-optimizing the schedule around critical tasks, rescheduling preserves overall mission objectives while retaining flexibility to accommodate unanticipated events. This capability is especially important in multi-satellite constellations, where conflicts in ground-station usage or imaging capacity emerge more frequently.

Finally, once the plan is finalized, the generation of satellite telecommands translates high-level directives—such as “capture imagery over region X at time T”—into low-level instructions understandable by onboard flight computers. These include attitude control sequences, payload activation periods, and data downlink allocations. By automating telecommand generation, the system minimizes operator workload and ensures consistency across each stage of request handling.

3. EnduroSat’s Mission Tasking Architecture

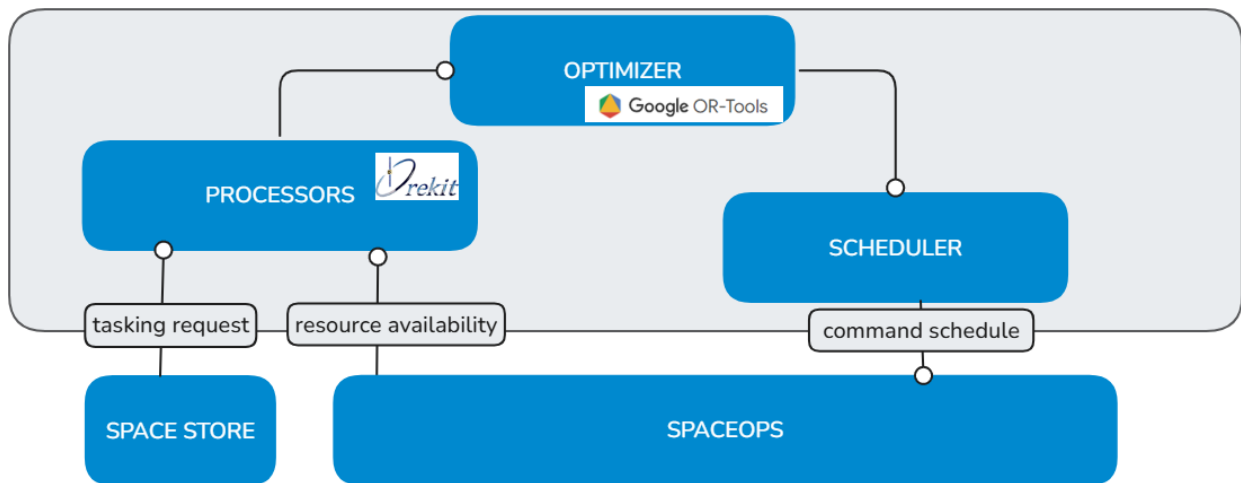


Fig. 1. EnduroSat’s Mission Tasking Architecture

3.1 Interface with SpaceStore

The interface with SpaceStore forms the primary gateway by which users submit mission tasking requests and manage their imaging needs. EnduroSat’s SpaceStore commercial portal offers multiple functions, including user account management, area of interest (AOI) definition, and the import of geospatial data in GeoJSON, KML, or Shapefile formats. Through this interface, users can specify detailed imaging parameters such as sun elevation, off-nadir angle, or cloud percentage thresholds. The portal also allows customers to initiate new image acquisition requests and monitor the status of pending missions. As requests enter the system, they are stored in a dedicated tasking request database that feeds into the mission management module. This seamless integration ensures that the latest operator and user inputs are immediately available for subsequent processing in the tasking workflow.

3.2 Interface with SpaceStore

Within the architecture, processors are the computational engines responsible for translating incoming requests into actionable data. These processors perform pre-processing tasks such as tasking plan construction and update, while also embedding satellite tasking algorithms that consider system constraints (i.e., satellite duty cycle, onboard data storage, and operational periods of uplink/downlink contacts). The processors apply filtering, and sanity checks to incoming requests, confirming that the defined areas of interest and imaging criteria are feasible given the current orbital state and weather forecasts. They also ensure that satellite resource limitations are considered and that requests maintain priorities pre-assigned by the user or auto determined by the system. To provide the necessary level of orbital propagation and modelling of the satellite environment, attitude and instrument geometry the Orekit library [3] is used. As a mature open-source solution with a strong heritage in space mission operations, Orekit's extensive validation through years of use in both academic and industrial projects underscores its reliability.

3.3 Opportunity Optimizer

A core component of the mission architecture is the Opportunity Optimizer, which analyzes the feasibility of each imaging opportunity. Because a user's AOI may require multiple images—often from various satellites in the constellation—the optimizer decomposes large area-of-interest requests into a series of linked imaging tasks. In our implementation, we leverage Google OR-Tools' CP-Solver [4] to handle the combinatorial complexity inherent in scheduling and resource allocation across the satellite constellation.

Using OR-Tools, the optimizer models each imaging request as a constraint satisfaction problem. The user's AOI, often covering a large geographic region that demands multiple overlapping images from different satellites, is segmented into discrete imaging tasks. The CP-Solver then examines each task's feasibility based on several factors: the satellite's current orbital position, instrument field-of-view constraints, pointing capabilities, and operational limitations such as duty cycle and data storage thresholds.

To achieve optimal assignment of imaging tasks, the CP-Solver uses a mix of constraint programming and heuristics to search for solutions that minimize overlapping coverage, balance the load among satellites, and respect resource and temporal constraints. Objective functions are configured to prioritize high-priority requests while minimizing costs such as fuel consumption or operational delays. For example, the optimizer can implement constraints ensuring that imaging tasks for adjacent segments of the AOI are scheduled sequentially, enabling efficient mosaic generation while leveraging overlapping coverage opportunities across multiple satellites.



Fig. 2. Swath's selected by the mission tasking algorithm for a given request

3.3 Command Scheduling

After optimizing opportunities, the Command Scheduling module translates the refined mission plan into executable telecommands. This module is tasked with sequencing tasks, aligning them with ground station contact windows and ensuring that uplink and downlink communication opportunities are leveraged efficiently. It generates command sets that include adjustments for pointing, instrument activation, and subsystem management. By scheduling commands in a time-ordered sequence, the module guarantees that satellite actions occur in compliance with predefined priorities while accommodating real-time changes through potential rescheduling loops. The result is an automated feed of satellite telecommands that is ready for uplink, ensuring rapid and reliable mission execution.

The other benefit that this provides is the ability to perform real-time validation in the EnduroSat Digital Twin. To validate that the constraints have been correctly programmed, the mission tasking will be able to automatically schedule a simulation run of satellite performing the mission tasking based on contextual information about the state of the satellite as retrieved from SpaceOps. In the event that any of the mission operational constraints are not respected or that the procedure fails to achieve the tasking goal, the operator will be notified and alternative tasking options proposed.

4. Scalability and Automation

EnduroSat's mission tasking system leverages cloud deployment in AWS to ensure the platform is scalable, robust, and tightly integrated with the SpaceOps environment. By hosting the entire system on AWS, the architecture takes full advantage of managed services leading to a cost-effective and highly available solution. This cloud-based approach facilitates the co-hosting of mission tasking alongside SpaceOps components, ensuring rapid communication between image processing, task scheduling, and satellite command operations. AWS enables scalability on demand, so as the constellation grows or user demand increases, additional compute and storage resources are provisioned almost in real-time.

A key architectural feature is the ability to parallelize processors by satellite and by request. The system is structured to run parallel workloads across multiple instances, with each processor handling either individual satellites or distinct request streams. This parallelization allows for simultaneous processing of numerous scheduling tasks, reducing latency and improving throughput. By using managed container services on AWS and incorporating tools for orchestration, the system distributes incoming tasking requests across the processing fleet while maintaining

consistency and coherency across operations. This design also supports processing tasks that can be segregated by satellite-specific telemetry or by user-defined acquisition requests, ensuring that edge-case scenarios do not bottleneck the overall system.

The mission tasking system is engineered to run predominantly in an automated manner. Automation is central to the architecture, interfacing smoothly with EnduroSat's event-driven architecture. This event-based design means that the system can automatically trigger planning and scheduling operations based on real-time operational events such as satellite state changes, new incoming acquisition requests, or unexpected anomalies. For example, if a satellite sends an updated telemetry packet indicating a shift in its nominal orbit, the event-driven system can immediately trigger a recalculation of tasking opportunities. Similarly, user submissions through the SpaceStore interface can automatically invoke the tasking engine, ensuring a seamless transition from user request to satellite command preparation.

Furthermore, a profound design choice has been the separation of mission tasking algorithms from mission-specific business logic. By decoupling the core algorithmic components from the contextual business rules, the system maintains a high degree of modularity and adaptability. The mission tasking algorithms, which include scheduling, satellite selection, and optimization routines (often using tools such as Google OR-Tools), operate independently of the specific commercial or operational constraints defined by end users. This separation allows the tasking engine to be re-adapted and reused in a variety of contexts without significant re-engineering. For instance, while one mission may emphasize high-resolution imaging for agriculture, another may prioritize rapid revisit times for disaster monitoring. In both scenarios, the core tasking algorithms remain consistent, while mission-specific rules are plugged in as configuration or separate business logic layers. This design not only accelerates development cycles but also simplifies maintenance and future enhancements.

By deploying in AWS, parallelizing processing tasks effectively, embracing an automated, event-driven model, and cleanly separating algorithmic logic from business rules, EnduroSat's mission tasking architecture is prepared for scalability, rapid adaptation, and long-term evolution. This robust foundation enables the system to evolve alongside technological advances and growing mission requirements, ensuring it remains agile, efficient, and resilient in a dynamic space operations environment.

5. Challenges and Future Work

EnduroSat's mission tasking system continues to evolve, and several challenges and future work directions remain. One primary challenge is to support a wider variety of mission profiles beyond the current imaging tasks. As satellite technology advances and mission requirements diversify—ranging from hyperspectral imaging to in-orbit servicing—the tasking system must adapt its scheduling and resource allocation algorithms to meet these new demands. This involves updating models to account for different operational constraints, sensor characteristics, and priority metrics tied to various mission objectives

Another significant challenge lies in enabling the tasking algorithm to accommodate multiple sensor swath profiles that contribute to the same request. Currently, the system handles single-sensor configurations well; however, when a request for a large area of interest can be serviced by overlapping sensor swaths from different satellites, complexity increases. Future enhancements must incorporate algorithms that not only optimize individual sensor footprints but also intelligently merge data from multiple sensors [5] [6]. This would improve coverage, enhance resolution, and reduce redundancy by ensuring that the same area isn't repeatedly imaged unless necessary for a critical mission requirement.

Alignment with established interoperability frameworks such as Domino-X is also a key future objective. Domino-X focuses on promoting standardized integration and data exchange across satellite systems and ground segments [7]. By ensuring EnduroSat's mission tasking architecture conforms to such frameworks, the platform can achieve enhanced compatibility with other space systems and stakeholders, facilitating collaborative missions and data sharing. This interoperability would ideally set the stage for broader ecosystems where multiple organizations can seamlessly integrate their tasking solutions, protocols, and data processing pipelines.

Overall, addressing these challenges requires an interdisciplinary approach—refining algorithms, expanding sensor models, and adhering to international standards—to realize a more versatile, adaptive, and interoperable mission tasking solution that meets evolving operational needs.

6. Conclusion

EnduroSat's constellation mission tasking service signals a major leap forward in automating Earth observation satellite operations. The system emerges from a confluence of cloud technologies, advanced scheduling algorithms, and direct integration with satellite telemetry. This end-to-end automation reduces the latency associated with collecting, scheduling, and delivering valuable Earth observation data to users.

By enabling rapid identification of imaging opportunities, dynamic re-tasking, and automated conflict resolution, the service addresses the market's critical need for timely, efficient satellite imaging. From a technical standpoint, future lines of research include improved multi-agent coordination leveraging inter-satellite links, advanced machine learning for predictive maintenance and anomaly detection, and deeper integration of cross-platform environmental data to comprehensively satisfy user's needs.

Through developing and deploying this mission tasking architecture, EnduroSat contributes to the space operations community by offering a practical model for optimizing the full data chain for both commercial and research-driven missions. As the mission planning landscape continues to evolve, collaboration among industry leaders along with academic and government agencies can further accelerate advancements in scheduling algorithms, cloud infrastructure, and data analytics that serve the entire EO community.

References

- [1] EnduroSat, 2025, <https://www.endurosat.com/projects/the-balkan-constellation/>
- [2] EnduroSat, 2025, <https://www.endurosat.com/spaceops/>
- [3] Orekit, 2025, <https://www.orekit.org/>
- [4] Google, 2025, <https://developers.google.com/optimization>
- [5] D. Eddy and M. J. Kochenderfer, A Maximum Independent Set Method for Scheduling Earth Observing Satellite Constellations, 2020, <https://arxiv.org/abs/2008.08446>.
- [6] Loft Orbital Solutions Inc. (2023). Scheduling and Optimization for Heterogeneous Space Systems. In Proceedings of the 17th International Conference on Space Operations, Dubai, United Arab Emirates, March 6–10, 2023.
- [7] European Commission / Domino-X Consortium. (Issue v1.0). Design and Interface Requirement for Coverage Management (Deliverable D3.2).