

SpaceOps-2025, ID # 451

**MAPS, multi mission centre**  
**Nicolas Precigout<sup>a\*</sup>, Emilie Converset<sup>a</sup>, Fabien Scelles<sup>a</sup>, Vicent Debout<sup>a</sup>, Guillaume Durand<sup>a</sup>,  
Geoffroy Periot<sup>a</sup>, David Bourrel<sup>a</sup>, Guillaume Renard<sup>a</sup>**

<sup>a</sup> CS-Group, France, nicolas.precigout@cs-soprasteria.com,

\* Corresponding Author

**Abstract**

Since 2011, CS Group has been developing and maintaining various mission simulators for a range of missions, including CSO, PHR, MICROCARB, YODA, CELESTE, and EGIDE. This work has allowed us to model and analyse a wide variety of missions across multiple domains, such as space and deep-space observation, Earth observation, and military space operations.

This experience enabled us to gain a clear understanding of the issues at stake and what each mission had in common in terms of mission centres. Thus, we have set up for the CNES two common bases so that we wouldn't have to start from scratch with each new mission. These common foundations bring together the cross-cutting functionalities for each mission, thus experience from old missions can serve to the new ones.

Building on this expertise in mission centres and simulators, CS Group is now developing its own generic mission centre, MAPS (Mission And Planning Software). MAPS is a powerful and adaptable mission centre designed to manage satellite constellations, process user requests, and generate mission plans for these constellations. It is based on a modular and scalable framework, allowing for the integration of all necessary services specific to each mission.

Thanks to its modular design, MAPS can be tailored to different types of missions and satellites, including varying payloads, platforms, and instruments. It is specifically designed for multi-satellite missions and utilises state-of-the-art, open-source technologies. As a microservice system, MAPS is ready for deployment on clusters and can be integrated with other components of the Ground Segment. It also provides a multi-user web application for broad accessibility.

A key challenge in a mission centre is the wide range of user requests, which may come from various companies and governments. These requests can be grouped by the user centre to optimise processing times. User requests will be managed using a variety of Key Performance Indicators (KPIs), which may be dynamic during mission planning and related to access scheduling.

This paper will explore how we optimised the mission management process to meet the system's performance and responsiveness requirements.

**Keywords:** Mission Planning, Mission Centre, Scheduling

**Acronyms/Abbreviations**

Mission and Planning Software (MAPS)

Ground Operation System and Mission Intelligence Centre (GOSMIC)

Data Take Opportunity (DTO)

Earth Observation (EO)

Flight Dynamic System (FDS)

Orekit Flight Dynamic System (Oreflids)

Key Performance Indicator (KPI)

Human man Interface (HMI)

Application Programming Interface (API)

## 1. Introduction

To address the growing number of commercial space actors and nanosatellite constellations, CS Group has developed the GOSMIC[1] satellite ground segment. GOSMIC integrates multiple key components, including Control Centre (CSNano), Mission Planning (MAPS), Flight Dynamics System (OREFLIDS), Programming Systems, and Production Centres. Its Kubernetes-based architecture enables fast and reliable deployments on scalable clusters. Since all GOSMIC components share common technologies and design principles, they are inherently compatible and easy to integrate, while also remaining fully operational as standalone solutions.

This paper is structured as follows: first, we detail the MAPS architecture, its key advantages, and how it addresses the challenge of genericity. Next, we describe the different MAPS components, highlighting their modularity and customisability. Finally, the last section presents the Human-Machine Interface (HMI) and data processing in MAPS from an operator's perspective.

## 2. The generic Mission Centre

### 2.1 Architecture of the Ground Segment

The objective of the MAPS solution is to design and implement a Mission Centre that integrates all necessary functionalities for a specific mission. Each MAPS instance is dedicated to a single mission, requiring separate deployments for different missions, and is designed for effortless integration into a Satellite Ground Segment. The Mission Centre operates in coordination with other Ground Segment components to achieve the mission's overall objectives. Fig.1 illustrates the role of MAPS within the Gosmic Ground Segment architecture.

During the mission preparation phases, the Mission Centre primarily communicates with the Control Centre and the FDS to obtain computational data related to each satellite, such as ephemeris, station visibilities, constraints, and more. Once the mission plan is computed, it is transmitted to the Control Centre for upload. Upon completion, MAPS receives an acknowledgment confirming the upload. A final feedback loop completes the mission cycle: the Mission Centre collects data from the Image Chain to link the initial request to the resulting product. Overall, Ground Segment operations supported by MAPS functionalities are orchestrated by the Automation component, which schedules and monitors actions in synchronization with other system components.

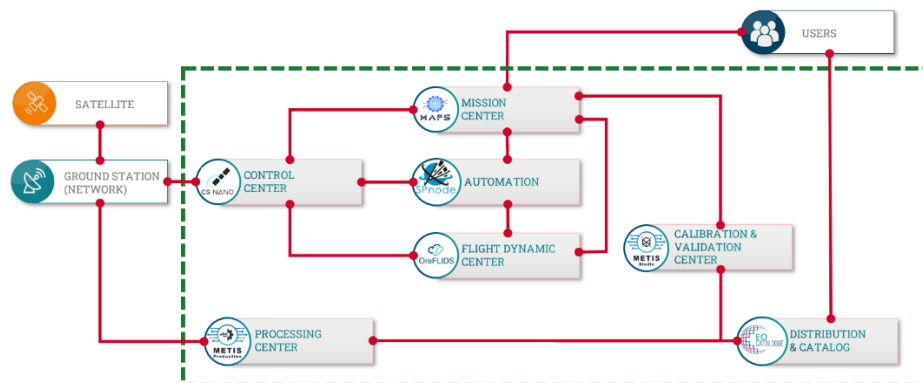


Fig.1 MAPS interaction in the Ground Segment

### 2.2 Characteristics of the Mission Centre

The MAPS framework is designed to be adaptable to a wide range of missions. Its highly configurable architecture allows customers to tailor each MAPS instance to accommodate mission evolution, such as heterogeneous constellations and different mission phases. This adaptability enables MAPS to handle efficiently diverse customer requirements and satellite roles.

While MAPS can interface with any mission library, each mission instance can also incorporate a high-fidelity mission library for enhanced accuracy. The mission centre is compatible with a variety of operational constraints [2], ensuring the generation of reliable mission plans.

The microservices-based architecture allows MAPS to scale efficiently for large constellations, as cluster resources can dynamically adjust based on the number of satellites. All computational tasks can be parallelized, except for the generation of the global constellation plan.

MAPS automatically generates optimised mission plans over a configurable time period, eliminating the need for manual intervention. Operators interact asynchronously to modify the request pool and adjust priorities, significantly reducing operational costs and risks [3]. This shift enables operators to focus on high-level analysis and customer interactions rather than routine mission planning tasks.

MAPS has been shown to meet the key characteristics expected of a commercial mission centre, as outlined by [4].

## 3. Customizable and modular

### 3.1 Modular architecture

The MAPS solution offers an architecture for the Mission Centre structured around two primary components:

- **User Request Chain:** Dedicated to managing the user’s initial image acquisition requirements.
- **Planning Chain:** Responsible for preparing and computing the satellite mission activities.

These two chains are separate, each covering distinct mission functionalities, as shown in Figure 2. The complete Mission Centre integrates these two chains, which work in cooperation. This design provides MAPS with a high degree of modularity, enabling the construction of a tailored Mission Centre instance for each mission. The system is initially designed to include a single instance of the User Request Chain within a MAPS instance, but for specific system requirements, it can support multiple ones.

Both chains (User Request and Planning) are deployable on a Kubernetes cluster and are implemented as distributed applications with multiple services. Each chain has a backend component that interacts with other external systems in the Ground Segment to exchange data and provide APIs for triggering mission-related computations. The specifics of the data exchange depend on the configuration of the considered MAPS instance. Services offer a REST API for external interactions.

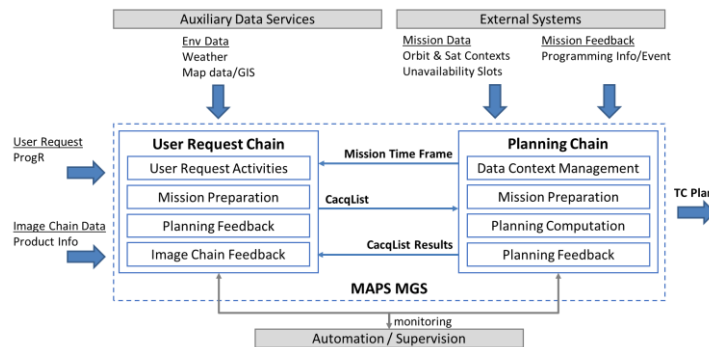


Fig.2 MAPS components

### 3.1.1 User Request Chain

The User Request Chain manages the user’s initial image acquisition requirements. This chain translates the user request into acquisition requests that are in line with mission constraints and satellite capabilities. These acquisition requests are then submitted to the Planning Chain to schedule the satellite’s acquisition activities.

The User Request management component aggregates all user activities, allowing the creation, modification, and tracking of image requests that the satellites will process. All user requests are stored by the component, and the lifecycle of each request is returned to the user.

A mission analysis for each user request provides a preview of the satellite activities required to fulfill the request. The User Request Chain receives information about the upcoming operations (Mission Time Frame), which enables the completion of acquisition requests. It provides services to produce a Candidate Acquisition List, containing possible acquisition opportunities in the next planning time frame. During the mission preparation phase, several processing steps occur, including census and automatic ranking. This Candidate Acquisition List is sent to the Planning Chain for potential inclusion in the mission plan.

In response to this Candidate Acquisition List, the User Request Chain receives a result detailing the effective planning status of each acquisition request. The associated User Requests are updated accordingly, providing the user with real-time information. Finally, feedback from the Image Chain links the completed acquisition with the product identification.

### 3.1.2 Planning Chain

The Planning Chain is responsible for preparing and computing the satellite mission activities, ensuring they cover the user requests and respect operational constraints. During the mission preparation phase, the Planning Chain determines the next planning time frames and conditions based on mission inputs and constraints. The resulting information is sent to external consumer components, including the User Request Chain.

A key mission step is the acquisition plan computation. Depending on mission requirements, this process handles user acquisition requests, mission constraints, satellite capabilities, and resource limitations. The goal is to produce a plan that optimises the use of the constellation (maximising acquisitions, balancing load) while respecting platform constraints (e.g., energy, onboard memory, system limitations).

The final step in the mission preparation process is the generation of the mission plan by satellite. This involves creating a sequence of satellite operations using the acquisition plan. The mission plan computation occurs in several

stages (kinematic, downloading, and payload plans), with each step defined by mission needs. The mission plan is then sent to the Control Chain, which manages communications with the satellite.

In response to the Mission plan, the Planning Chain receives operation and onboard feedback concerning success or failure. This information is used to update the planning status within the Planning Chain, allowing for ongoing monitoring and preparation for future planning time frames.

### *3.2 Highly customizable DTO*

MAPS cuts the customer requests into elementary geographic meshes and compute the potential accesses for every mesh [2]. The so-called DTOs represent time slots during which a satellite can perform acquisitions of a particular mesh. For a given programming horizon, MAPS generates a list of DTOs that are candidates for inclusion in the mission plan for available user requests. These DTOs are evaluated and assigned a global score, which is the weighted average of several KPIs.

KPIs are essentially normalized evaluations based on different criteria. They can be derived from higher-level customer demands (e.g., priority level, due date), the mesh (e.g., geographic position, altitude, landscape position, mesh size), inherent attributes of the DTO (e.g., forecast weather, acquisition elevation angle), or satellite-specific factors (e.g., available instruments, resources). Operators can customize the relative weights of KPIs and even create new complex KPIs using explicit mathematical formulas.

KPI configuration, as well as overall ranking configuration, is stored on a configuration server in a JSON structure. Each KPI can be customised and activated in real-time, meaning that the next scheduled ranking will incorporate the updated configuration.

## **4. Microservice design and online HMI**

### *4.1 Database and Resource management*

The User Request Chain and the Planning Chain are each associated with a persistence module through a database. The database for the User Request Chain stores user requests, while the Planning Chain database records the different programming periods for each satellite according to the mission chronology and stores a copy of the onboard state of each satellite.

These databases use PostgreSQL [5] in combination with the Hasura GraphQL Engine, enabling the generation of GraphQL APIs. The search functionality is driven primarily by database queries, which include basic geographical search capabilities [6]. Mission data, computation results, and metadata are stored and indexed in S3 buckets, improving access efficiency for various service requests, while still scaling with the number of satellites. A dedicated REST service handles external data exchanges, transforming data between specific internal mission formats. This service allows MAPS to connect to various external data sources, such as weather data, orbital information, and satellite state data.

### *4.2 HMI*

MAPS can be operated directly via the REST API services and scheduled by the automation system, or through a web-based HMI, minimising the hardware and configuration requirements for users. The visualisation includes a map to display customer requests and satellite tracks as shown on Fig. 3, a comprehensive mission timeline [7], and a follow-up of customer requests. The HMI also offers controls for creating or modifying demands.

Every action is processed through a microservice that updates the system, and saving or searching for requests is done using the GraphQL API [8].

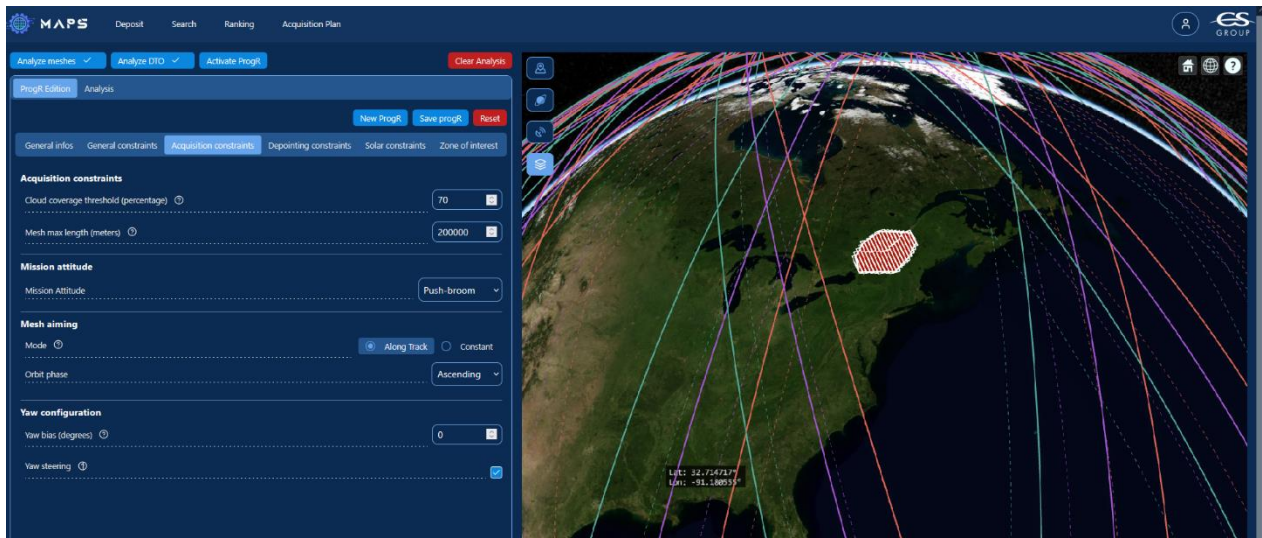


Fig.3 MAPS HMI

## Conclusion

MAPS, the Mission Centre developed by CS Group, has been presented as a solution to address the growing number of commercial Earth observation constellations, with a focus on the automatic generation of mission plans. Thanks to its microservice-based architecture, MAPS exhibits the expected characteristics of a modern Mission Centre. It is scalable to accommodate increasing numbers of satellites and highly adaptable to most Earth observation missions. Operators can customise it during the mission, offering high flexibility in configuration.

The MAPS solution facilitates faster and easier development, leading to cost reductions in the creation of the Mission Centre instance, that can be adapted to a wide range of Earth observation mission needs.

MAPS is built on a framework that provides scalability and modularity, enabling the composition and integration of all services required for a specific Mission Centre instance tailored to a particular mission. This solution combines a robust framework with a missionisation strategy to mitigate the challenges associated with integrating mission-specific features, such as custom algorithm libraries or services, and simplifies the overall implementation of the Mission Centre instance.

A web-based HMI allows users to comprehensively modify the demand pool and monitor the timely completion of requests.

## References

- [1] Cyril THOMAS, Eric SAVES, Advanced Ground Segment Systems for Modern Space Missions, #455
- [2] Vincent Debout, Nicolas Precigout, Fabien Scelles, Javier Pelegrin Garcia, Guillaume Durand, Geoffroy Periot, David Bourrel, Guillaume Renard, Emilie Converset. 2025, this issue #452
- [3] Michel M. Verstraete, David J. Diner, Jean-Loup Bézy, Planning for a spaceborne Earth Observation mission: From user expectations to measurement requirements, Environmental Science & Policy, Volume 54, 2015, Pages 419-427, ISSN 1462-9011, <https://doi.org/10.1016/j.envsci.2015.08.005>.
- [4] Iacopino, Claudio & Harrison, S & Brewer, Andy., 2015. Mission Planning Systems for Commercial Small-Sat Earth Observation Constellations.
- [5] PostgreSQL Global Development Group. (2023). PostgreSQL 15.4. <https://www.postgresql.org/>
- [6] PostGIS Project Steering Committee. (2023). PostGIS, spatial and geographic objects for PostgreSQL. <https://postgis.net>
- [7] Chien, Steve & Johnston, Mark & Frank, Jeremy & Giuliano, M. & Kavelaars, Alicia & Lenzen, Christoph & Policella, Nicola. (2012). Generalized Timeline Representation, Services, and Interface for Automating Space Mission Operations.
- [8] Hasura Inc. (2024). Hasura. <https://hasura.io/>