18*th* International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 20

Perform real time Spacecraft Operations from your mobile phone – an ESA's Mobile Spacecraft Operations Application for Android© and iOS©

Szymon Stuglik[a], Marta Pantoquilho[b], Franz Ritter[c], Jan Oertlin[d]

[a] *European Space Agency (ESA) / European Space Control Centre (ESOC), Szymon.Stuglik@esa.int*
[b] *European Space Agency (ESA) / European Space Control Centre (ESOC), Marta.Pantoquilho@esa.int*
[c] *SpaceCube GmbH, f.ritter@space-cube.de*
[d] *SpaceCube GmbH, j.oertlin@space-cube.de*

Abstract

The pandemic required Flight Control Teams (FCT) and Software Support Teams (SWS) to work from home, but mission operations at the European Space Operations Centre (ESOC) remained successful thanks to their dedication. To make remote work easier and safer, the Mobile Spacecraft Operations (MSO) application was developed. The MSO app, available on mobile devices (Android/iOS) and on a web browser, provides near real-time spacecraft telemetry, events, telecommand history, real-time commanding stacks, and operators' log entries, for missions operated from ESOC.

The MSO app allows users with relevant permissions, including FCT, SWS, and remote Project and Industry Teams, to access mission status remotely, potentially reducing unnecessary on-site travel. Challenges encountered included scaling information for mobile screens and gaining the trust of operators and managers that it is possible to (almost) fly any spacecraft from any mobile device, with the given permissions.

The app uses cross-platform technologies like Ionic/Cordova, possibly transitioning to Flutter/Dart soon. The backend includes authentication, authorization services, and an aggregator server for microservices at ESOC. This paper discusses the initial idea and introduction of the app through the innovation Hub, the chosen technology, and the app's potential to transform spacecraft operations even beyond extraordinary circumstances such as the pandemics.

Keywords: Mobile, Spacecraft, Operations

Acronyms/Abbreviations

ARES .... Operational data off-line Analysis, Correlation and Reporting System
CSUP .... Command Supervisor
DTO .... Data Transfer Objects
ESA .... European Space Agency
ESOC .... European Space Operations Center
FCT .... Flight Control Team
InnoCup .... ESA innovation cup
LDAP .... Lightweight Directory Access Protocol
MMaaS .... Multi-Mission-as-a-Service
MSO .... Mobile Spacecraft Operations
MVC .... Model-View-Controller
RBAC .... Role-Based Access Control
SWS .... Software Support Team
UI .... User Interface

UX .... User Experience

1. Introduction

In the last 20 years the number of ESA flying spacecraft has tripled. To efficiently support such an expansion, new tools, systems, processes, and methodologies must be brought to life. Furthermore, the COVID-19 pandemic showed a shortage of utilities allowing for quick spacecraft monitoring, especially in the context of remote and hybrid work.

The Mobile Spacecraft Operations (MSO) application has been pitched to – and subsequently selected by – ESA/ESOC innovation committee during the first ESOC Innovation Cup challenge as a solution for the above issues.

This paper describes the process of bringing MSO from a concept and ESA Innovation Cup spin-off to an operationally ready product, the process of technology selection, working with limited resources (time, budget, and workforce) as well as the lean and agile development process driven by user feedback (Flight Control Team members). We will conclude by outlining the plans and challenges ahead.

2. ESA Innovation Cup

The ESA Innovation Cup is a strategic program launched at the European Space Operations Center (ESOC) to promote the culture of innovation and collaboration within ESOC. This initiative encourages ESOC employees to exploit their vast experience stemming from years of work in or with mission operations together with newly acquired technological insights and propose innovative, cutting-edge – and occasionally – disruptive ideas.

Mobile phones revolutionized how we interact with the world, and have spread to almost all areas of human life – why should spacecraft operations be any different?

During the first edition of the ESA Innovation Cup (InnoCup) the Mobile Spacecraft Operations Application idea was pitched and subsequently selected by the InnoCup committee for development. Virtually at the same time the world has been affected by the Covid-19 pandemic, which very quickly showed that better tools are needed for Spacecraft Monitoring and Control – especially in the context of remote and hybrid work (caused by lockdowns during the pandemic).

Participation in the ESA InnoCup projects is voluntary, and selected ideas are given some kick-off budget and are allowed to devote some of their working time to the InnoCup projects. The MSO application has been brought to life by the members of the ESOC Data Systems section with the significant help of ESA trainees and interns (and as such, served as terrific educational project in the fields of spacecraft operations, systems engineering and DevOps).

3. Challenges in Space Operations

Missions have grown substantially in complexity over the last decades: be it due to more demanding instruments, more demanding routes to distant planets and stars, or even in the number of spacecrafts in a constellation with coordinated orbits. In parallel, missions have also been moving towards increased automation on-board as well as on-ground, since years. Nowadays there are considerably fewer missions that require the attention of a trained human around the clock, and 24/7 Spacecraft Controller services are currently reduced at ESA/ESOC. The goal is to have completely autonomous missions, outside critical phases. This allows for a better usage of trained individuals in other roles around Spacecraft Operations, while also decreasing costs of the missions operations.

Increased automation, however, still relies, and will continue to rely, on human decisions for any contingency or unexpected scenario. Mission engineers are continuously on-call, off-site, in case there is an unexpected problem with the spacecraft. Information is therefore crucial to any on-call engineer. There are several levels of information that are useful in operations: 1) when, 2) why, and 3) where. The “When” provides information that triggers the engineers to look into the spacecraft status and health. It is like an alarm. The “Why” tells the engineers why it is worth to look at the spacecraft and the “Where” tells them where to investigate, such as which subsystem, or which procedure is applicable. Once a problem is signaled, quick access to the information needed for the investigation is crucial, before a decision is made on which action to take. Having the most information available, in an accessible and workable way, allows taking initial steps into the investigation of problem solution while enroute to the control center, even without access to a laptop.

Many different web-based solutions exist at ESOC for monitoring spacecraft health and status since a long time, each providing access to different types of data and information. What did not yet exist was a single point of entry for all those tools that aggregates data from all of them and displays them on a mobile device.

4. Key Features

The MSO application is designed to provide on-call engineers with quick and convenient access to critical spacecraft data, enabling them to obtain an overview of the spacecraft's state. This is achieved by integrating existing ESA services into the MSO infrastructure and rendering the service data in a mobile-friendly application (see **Error! Reference source not found.**). The integration is implemented through the respective service APIs, allowing data to be processed, aggregated, cached, and made available to the frontend. Existing services are designed for direct operational input and control, whereas MSO serves as a complementary tool focused on providing a clear and accessible overview of spacecraft data. To maintain this purpose, MSO operates in a strictly read-only manner, with no support for commanding or other user-driven modifications

MSO follows a Multi-Mission-as-a-Service (MMaaS) approach within ESA. Rather than being tailored to a specific mission, it is designed as a generic platform capable of supporting all missions that utilize at least one of the integrated services. This ensures that engineers working across multiple missions can use a single system, simplifying their workflow and eliminating the need to adapt to different mission-specific tools.

Since MSO was conceived as a mobile-first application, its user interface has been designed to be responsive to varying viewport sizes from the outset. The primary development focus is on ensuring usability on mobile devices while maintaining compatibility with larger screens (see Section 5.1. for details).

At present, MSO integrates three services – ARES, Überlog, and CSUP – with additional integrations planned for future development:

- **Operational data off-line Analysis, Correlation and Reporting System (ARES)** provides access to telemetry and telecommand histories, spacecraft events, and real-time telemetry data as well as parameter definitions.
- **Überlog** is an electronic, web-based logbook system.
- **Command Supervisor (CSUP)** offers insight into currently loaded telecommands on the command stack.

Once logged into the application, users can select from a list of missions for which they have been granted access (see Figure 2; for details on permissions, refer to Section 5.2.2). Below the mission selection, users can access all integrated services associated with the selected mission. The application presents a dashboard that provides a condensed overview of all integrated services, offering a centralized visualization of key mission data.

MSO provides access to the following mission data:

- Telemetry history
- Telecommand history
- Spacecraft events
- Telemetry parameters (visualized as graphs)
- Command stacks and their contents
- Logbook entries

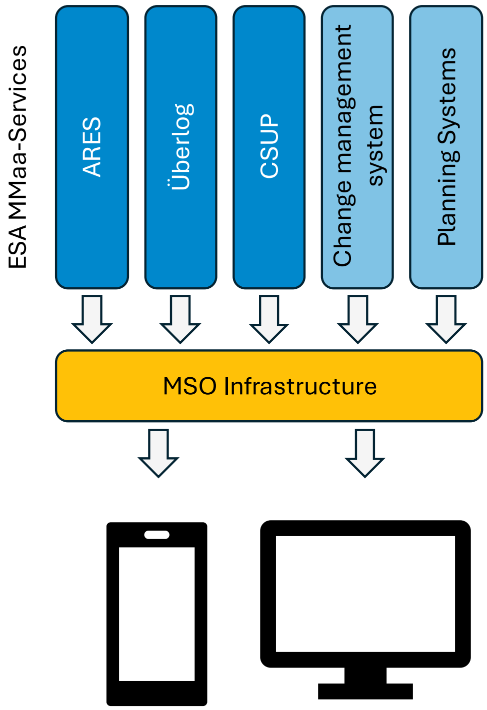

Figure 1: The MSO infrastructure serves mobile- and desktop-friendly user interfaces while using several services like ARES, Überlog, and CSUP. A change management system for procedures and a planning system are on the roadmap to be integrated.

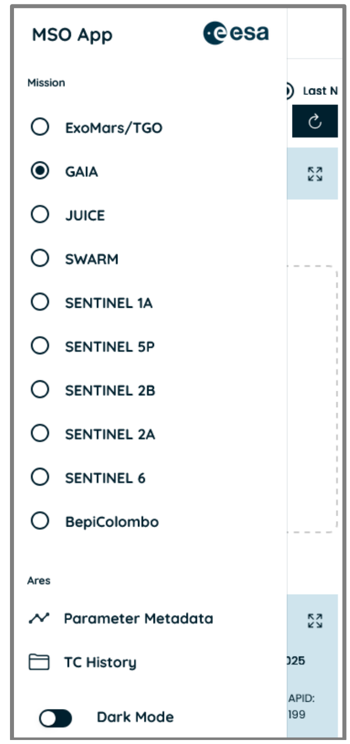

Figure 2: Mission-selection area in the navigation of MSO rendered on a mobile device. All missions the user has permission to access are available for selection.

4.1 Data Representation and Interaction

The telemetry history, telecommand history, and spacecraft events are presented in a tabular format, allowing users to inspect relevant details. A harmonized search and filtering system, including a time-range selector, facilitates data retrieval, enabling users to quickly locate specific events or parameters of interest.

Telemetry parameter values can be selected and visualized in a line chart, allowing users to plot multiple parameters simultaneously within the same graph (see Figure 3). To enhance usability, MSO allows users to save specific chart configurations as "queries," which can be reloaded for future use. Additionally, queries can be shared with other users by marking them as "public," enabling collaborative analysis and reducing redundant configuration efforts.

Users can also inspect the current command stacks, with the ability to drill down into individual stacks and examine each command along with its associated parameters. The command stack data is rendered using the same table structure as the telemetry history ensuring a consistent user experience and enabling quick search and filtering across all datasets.

The logbook is displayed in MSO both as an aggregated view and as a detailed list of entries. Users have access to the same search and filtering tools as in the telemetry history table, with an additional filter for log entry severity levels. Since missions may maintain multiple logbooks, MSO allows users to select specific logbooks from which data should be displayed, ensuring flexibility in data retrieval.

5. System Overview and Features

This section outlines the motivation behind adopting a mobile-first approach for MSO, emphasizing its necessity for spacecraft data access in operational scenarios. It details how this approach enhances usability, responsiveness, and efficiency for on-call engineers. Next, the technical implementation of the mobile-first design is explored, covering both frontend and backend architecture.

5.1 The Mobile-First Approach

The mobile-first approach prioritizes application design for small screens before scaling up to larger displays, with an emphasis on usability, performance, and accessibility. In contrast, a desktop-first approach optimizes data rendering for large screens, which, when scaled down to smaller devices, may lead to suboptimal visualizations and reduced usability. By designing with small screens in mind from the outset, the mobile-first approach ensures an optimal user experience for mobile users.

In operational scenarios, on-call engineers do not always have direct access to a desktop workstation with ESA VPN connectivity. Instead, they rely on a dedicated mobile phone, enabling quick access to critical spacecraft data outside of office environments. Rapid response is essential in space operations, and MSO enables engineers to assess the situation remotely and determine the appropriate next steps – such as whether an on-site intervention is necessary. Having a mobile-first design ensures that mission-critical, real-time information remains accessible from any location, provided an internet connection is available.

Next to the mobile-first approach, User Experience (UX) plays a central role in ensuring quick and intuitive interaction with the application. A well-designed UX enables engineers to efficiently access critical information, regardless of the device they are using. MSO is designed to provide a consistent interface across devices, allowing seamless transitions between screen sizes without requiring users to familiarize themselves with a different application layout. A key focus is ensuring a self-explanatory and harmonized user interface (UI) and UX across the whole application, minimizing the learning curve and eliminating the need for specialized training or extensive documentation.

User-centered design is a core principle of MSO's development process. An iterative development approach ensures that regular user feedback results in improvements to both the interface and workflows. The involvement of users across multiple missions provides valuable real-world insights, allowing for refinements based on operational needs. Additionally, maintaining a short development cycle, from feature development to deployment, ensures that users receive updates – including new functionalities and bug fixes – in short cycles.

5.1.1 Mobile Usability

To enhance usability on mobile devices, consistent, touch-friendly interface elements and intuitive gestures (e.g., swipe, tap) are incorporated throughout the application. These interactions improve accessibility and allow users to efficiently navigate data-heavy interfaces on smaller screens.

One of the most significant challenges in a mobile-first design is data representation, particularly for tables and charts. Traditional table layouts, which rely on multiple columns for displaying information, work well on larger screens but are impractical on mobile devices, where screen width is limited. Instead of attempting to replicate desktop-style tables, MSO employs responsive row designs that present mission-critical information concisely while still allowing efficient filtering and sorting. For example, the Command Stack List is displayed as a card-based layout on mobile devices, while retaining a traditional tabular format on desktops. Color coding further enhances readability by visually distinguishing different command stack states.

Similarly, visualizing telemetry data in charts presents unique challenges on mobile screens. Line charts, which typically use a horizontal time axis on desktop displays, face space constraints on mobile devices, where the horizontal dimension is more limited. MSO optimizes chart rendering by maximizing available screen space and providing intuitive navigation options. Users can pan through the timeline with swipe gestures and zoom in and out using pinch gestures, ensuring that charts remain readable and interactive across different screen sizes and orientations (see Figure *3*).

5.2. Technical Architecture

The architecture of MSO (see Figure 4) is divided into two main parts: the frontend and the backend. The frontend is built with Angular and leverages the Ionic framework to support the mobile-first approach. The backend is based on the Java Spring framework, enabling fast integration with the required systems and ESA services. Additionally, the backend incorporates key components like Keycloak for user authentication and authorization, and a robust database layer for managing mission data and configurations.

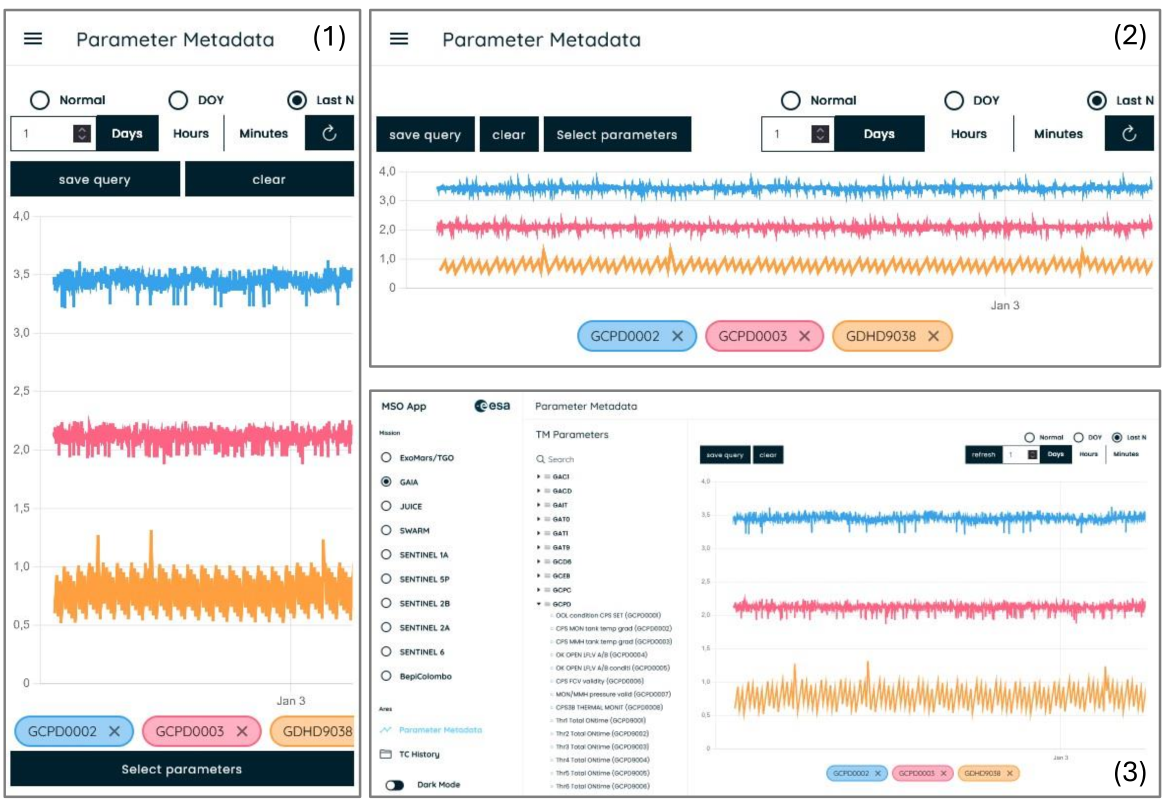

Figure 3:Visualization of three telemetry parameters. (1) portrait representation on a mobile devices; (2) landscape representation on a mobile device; (3) desktop representation.

5.2.1. Backend

The backend architecture (see Figure 4) of MSO is designed around an aggregation pattern, enabling integration with various Data Systems such as ARES, Überlog, and CSUP. The backend is responsible for making API calls to these systems, each of which exposes its own API. The retrieved data is then aggregated, processed, and structured within the backend before being forwarded to the frontend for visualization and interaction.

This approach does not only abstract system-specific complexities but also ensures alignment with the application's core objective of delivering a unified and structured view of spacecraft data. By providing a flexible and modular architecture, the backend enables efficient data aggregation while supporting future enhancements without being constrained by third-party system limitations.

To meet the modularity requirements, the Spring Framework was chosen. Spring provides a comprehensive set of modules that align with the software architecture and functional needs of the system.

One key module is Spring Web Model View Controller (MVC), which facilitates RESTful API compliance and streamlined web server integration. For interacting with third-party APIs, Spring WebFlux offers a highly efficient, non-blocking web client with state-of-the-art configuration options, including secure credentials management.

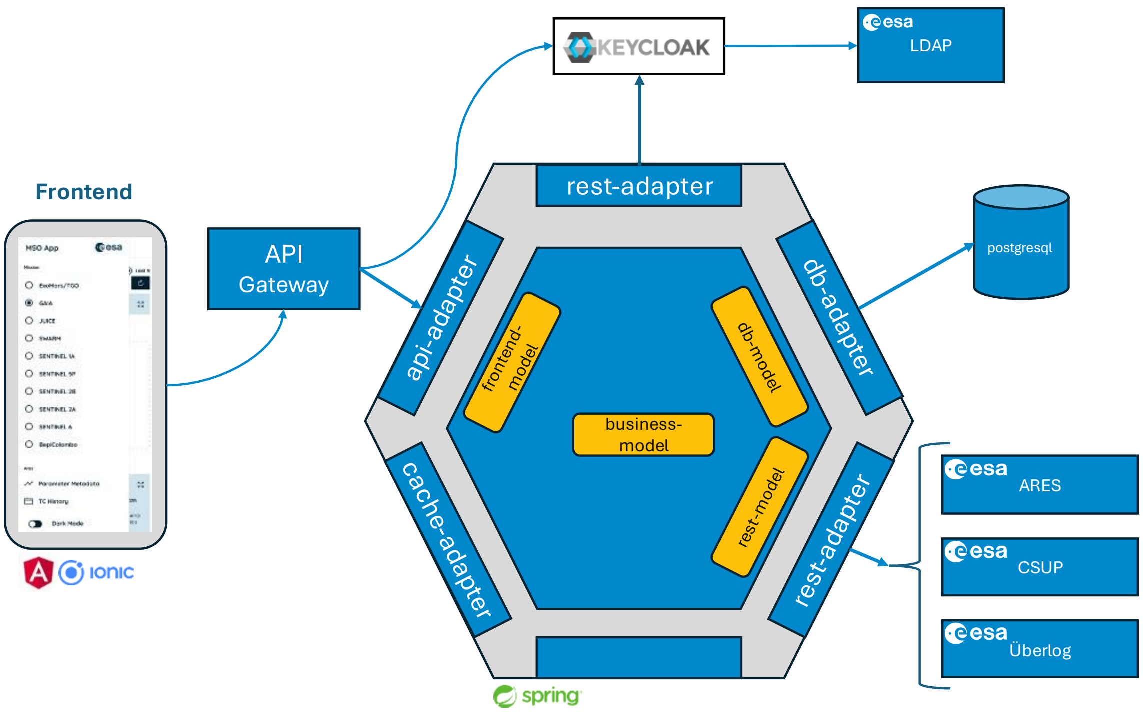

Figure 4: MSO architecture overview

Another critical requirement is database integration, as the system must store user and mission-specific data for configuration purposes.

Security is another essential aspect, given the sensitive nature of the data. Spring Security enables robust authentication and authorization, with built-in support for OpenID Connect (OIDC) on top of OAuth2, ensuring compliance with modern security standards.

Additionally, due to the aggregation pattern, frequent API calls to external systems can be resource-intensive. Spring's caching mechanisms help mitigate this challenge by reducing redundant requests. Furthermore, caching can be enhanced using Redis or other state-of-the-art solutions to optimize performance and scalability.

The project structure is heavily inspired by a hexagonal architecture, though it is not fully compliant with it. Two key considerations influenced this decision:

1. **Adapter Pattern for Communication Channels**
   Communication between the backend and both external systems and internal components is managed through adapters. These adapters ensure that the backend remains decoupled from the specifics of different systems,

providing flexibility and scalability for future integrations. Adapters allow the backend to interact with various services or components without needing to know the internal details of those systems. Following adapters are supported by the system.

- **REST-adapter:** Communication with external services (such as ESA's systems) is handled by REST-based adapters. These adapters are responsible for translating the data into the appropriate format for communication with each external service.
- **DB-adapter:** For database interaction, a database adapter is used, which employs the JDBC protocol. This adapter actively requests data from the database, providing a standardized way to interact with the database.
- **API-adapter:** Incoming communication is managed via the web client adapter, which handles requests from the frontend to the backend. This adapter translates the user requests into appropriate service calls to fetch and process data, ensuring smooth communication between the frontend and backend. The web client adapter also ensures that the incoming data is formatted and processed according to the business logic requirements of the backend.

This adapter pattern enables extensibility by allowing new services to be added to the system easily. New adapters, and therefore new external services, can be integrated without requiring modifications to the core application logic.

2. **Data Mapping Between Layers**
To ensure consistency and smooth flow of data between layers, the backend employs a structured data mapping process. This transforms data as it flows from one layer to another, ensuring that the data is in the appropriate format for the business logic layer and frontend components. The mapping logic is crucial for the decoupling between the systems and ensuring flexibility when integrating new services or changing external systems. Following mappings are applied in the system:

- **External Service to Backend Data Model**: Data retrieved from external services (e.g., ARES, Überlog, CSUP) is mapped from the external service data model into a backend data model used for the internal processing through the backend business logic. This transformation ensures that the backend business logic is independent from the data structures of the external services.

- **Backend Data Model to Data Transfer Objects (DTOs)**: For communication with the frontend, the backend data model used in the business logic is mapped to DTOs. DTOs are tailored representations of the data that simplify transmission to the frontend, ensuring that only the necessary information is provided.

- **Backend Data Model to Database**: When data needs to be persisted, the backend data model is mapped to the data structures used by the database.

5.2.2 Security

Security is a critical aspect of the MSO application, ensuring that sensitive spacecraft data is protected while providing authorized users with the necessary access. This section covers the various security mechanisms in place, including access control and authentication.

5.2.2.1 Authentication

User authentication in MSO is handled via Keycloak, utilizing OpenID Connect (OIDC) as a layer on top of the OAuth 2.0 standard. This approach externalizes authentication logic, reducing security risks associated with custom implementations. While ESA's Lightweight Directory Access Protocol (LDAP) integration is planned, it is not yet implemented. Currently, MSO requires dedicated user accounts, but future LDAP integration will streamline onboarding by enabling users to authenticate with their existing ESA credentials.

5.2.2.2 Access Control

To manage access control, MSO implements a mission-based authorization system, where permissions must be explicitly granted to each user. This centralized approach allows administrators to efficiently grant and revoke access without requiring users to maintain separate accounts for each mission. Once logged into MSO, users can access all assigned missions through a unified interface (see Figure 1). Additionally, a mission does not need to utilize all integrated services; MSO dynamically displays only the services available for the currently selected mission.

Access control in MSO operates at two levels. The first level, mission-based access, is managed directly by the MSO application as described previously. The second level, role-based access control (RBAC), is implemented via Keycloak to manage access to the centralized MSO interface and the application itself. Roles in Keycloak determine which users can access the administrator interface, while general user roles control access to the main application. Only users assigned the user role in Keycloak are authorized to log into the application.

By combining both mission-based authorization and RBAC, MSO provides a flexible, easy-to-configure system for managing access while ensuring the required level of security for mission-critical data and operations.

5.2.2.3 Network Security

Given the mission-critical nature of the integrated services and their data, strict access control measures are in place. As a primary security mechanism, data access is restricted to users connected to the ESA Virtual Private Network.

5.2.3. Frontend

The frontend is developed as an Angular application, leveraging the Ionic framework for UI components and Capacitor for native application builds. The architecture follows the Model-View-Controller (MVC) pattern, ensuring a clear separation of concerns and maintaining logic encapsulation. UI elements are structured into reusable components and modules, while business logic is encapsulated within services. Additionally, controller services abstract communication with the backend, handling data retrieval and processing through well-defined data model.

Given that MSO has moderate performance requirements, a single codebase is maintained and compiled for multiple platforms, including web, Android, and iOS (see Figure *5*). The web application is built using the standard Angular build process, while the Capacitor runtime is employed to generate native applications for Android and iOS, enabling seamless cross-platform deployment.

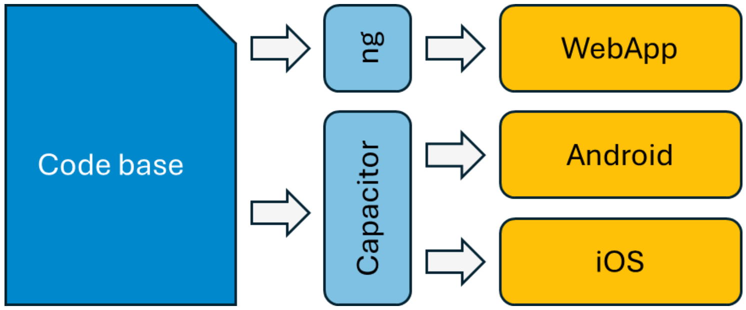

Figure 5: One code base is compiled for three different platforms

7. Future Enhancements

The rollout of the MSO application is still in its early stages, although core functionalities and initial service integrations have already been implemented. The MSO roadmap outlines several planned enhancements aimed at expanding the application's capabilities, increasing its user base, and improving operational efficiency. In particular, new service integrations and enhanced data aggregation tools will significantly improve the efficiency of anomaly investigation by consolidating information from multiple sources.

The roadmap includes usability improvements, additional service integrations, and technical advancements, such as:

- **LDAP integration** (see Section 5.2.2.1) to streamline authentication and access control.
- **Push notifications** to alert users when critical telemetry parameters exceed predefined limits. Integration with the **Advanced Alarm Management System** is planned to enhance real-time monitoring capabilities.
- **Integration of a change management system** for flight operation procedures (FOPs), operational actions, and mission database management within SCOS-2000-based mission control systems (MCS). This integration could enrich existing views by providing action lists and out-of-limit indicators.
- **Planning system integration** to allow users to inspect ongoing and upcoming operational activities.
- **Flight Dynamics System (GODOT) integration –** enable retrieval of mission specific flight dynamics data
- **EGS-CC integration:** Implement adapter for the European Ground Systems Common Core (EGS-CC) Infrastructure-Data-Access component to retrieve parameter-, telemetry- and activity data.
- **UI enhancements** to improve cross-service data correlation for more efficient problem investigation.
- **Frontend technology stack evaluation**, including an assessment of alternative frameworks such as Flutter/Dart, React Native, and Progressive Web Apps (PWAs) to optimize performance and maintainability.

8. Conclusion

The introduction of a mobile application to support centralized near-real-time and real-time spacecraft data access was welcomed by the flight control teams at ESOC. The restriction of the application availability to corporate mobile phones is, however, very limiting and is the main challenge to overcome, either through the wider spread of corporate mobile devices to the general workforce, or the availability of another secure solution allowing the application to be used from private mobile devices (e.g. via VPN application).

Since the MSO application is multi-mission, and thus applicable to any mission, with any subset of subscribed services, it has the potential to become part of the default ground data systems, provided with the mission control system and applications suite at ESOC. The current version of the application interfaces the SCOS-200 (S2K) baselined Mission Control Systems, and it will be extended to interface the new generation of mission control systems baselined on EGS-CC, and provided as an additional service for any mission on that baseline.

The future of MSO, as highlighted in this paper, consists as well in the continued aggregation of existing and immerging services, useful to mission operations, like a single entry point for all operational relevant information.

Although the application is currently focused solely on spacecraft monitoring capabilities, technically it could provide spacecraft control capabilities as well. With MSO, perhaps soon ESA could have its first spacecraft operated from a mobile device!

Acknowledgements

This work is part of the Innovation Cup program and funded by the European Space Agency. We would like to thank Thomas Hammon – former ESA trainee for his very significant contribution to the Mobile Spacecraft Operations project. His commitment and proactive attitude had a major impact on the evolution of the project. We would like to thank Artur Palowski from ESA/ESOC Innovation Office for his continuous support.