Implementation of Software Microservices for the Design and Development of a Satellite Detection and Monitoring System as Proof of Concept.

Manuela Zapata Quirós

Degree work to obtain the title of Aerospace Engineer

Internal Advisor

Juan Francisco, Puerta Ibarra, MSc.

External Advisor

Radim Badsi, MSc.

Universidad de Antioquia
Faculty of Engineering
Aerospace Engineering
Carmen de Viboral, Antioquia, Colombia

2024

Dedicatoria

A mi madre Isabel Cristina y a mi hermana Mariana, que siempre han estado a mi lado. Gracias por apoyarme siempre y por querer compartir conmigo tanto los buenos como los malos a pesar de la distancia.

Agradecimientos

Quiero hacer un agradecimiento especial a mi asesor externo, Radim Badsi, por su guía experta, por creer en mis capacidades y por brindarme no solo la oportunidad de aplicar mis conocimientos en su compañía, Groundspace, sino también por ayudarme para continuar mis estudios superiores en Francia y seguir trabajando para su empresa. Su confianza, generosidad y apoyo han sido y son cruciales para mi desarrollo profesional y personal. También quiero hacer un agradecimiento especial para Latifa, Océane y todos los empleados y practicantes de Groundspace, que me acompañaron y acogieron en este proceso.

No puedo dejar de mencionar a mis compañeros y amigos de la universidad: Santi, Samu, Sebas, Angélica, Adri, Juanjo y David a todos mis compañeros de los semilleros Delta V y Voyager. Gracias por acompañarme por este camino, por las risas, las horas de estudio, por los trabajos hechos en conjunto que nos han hecho crecer y soñar, y por todos los momentos que nos han unido.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Acronyms and Abbreviations

AI	Artificial Intelligence
AzEl	Azimuth-Elevation
API	Application Programming Interface
BSS	Broadband Satellite Service
EM	Electromagnetic
FSS	Fixed-Satellite Services
GEO	Geostationary Earth Orbit
GS	Ground Station
ITU	International Telecommunication Union
LEO	Low Earth Orbit
ML	Machine Learning
PoC	Proof of Concept
RAAN	Right Ascension of the Ascending Node
RFI	Radio Frequency Interference
UI	User Interface
TFS	Terrestrial-Fixed Services

ABSTRACT

This report presents the development of the author's internship that took place in Groundspace, a start-up located in Montpellier, France. This project focuses on the implementation of software microservices into Satmon, an upcoming on-development product designed to provide assistance to satellite operators against interference problems resulting from the coexistence of multiple satellite constellations and assure operational regulations compliance. The applied methodology for the software development was based around the GitLab suite an VSCode code editor, that allows to have a collaborative development workspace for task management, code review, and deployment. All microservices developed utilizes Python as the main programming language, and Skyfield and Astral packages for orbital position computations. The sky scan was the main developed feature, which performs a "blind" scan for a specified observation window to verify anomalous transmission activity and compliance of registered systems. Although challenges were encountered, including the Starwin antenna's lack of manual beam control, hardware constraints and initial setup issues; the prototype successfully detected GEO and LEO satellites and achieved software-hardware synchronization, with ongoing improvements expected to enhance functionality and user experience. The project anticipates future developments and a presentation at the International Astronautical Congress 2025 in Sydney, Australia.

***Keywords* — Microservice, Radio Frequency Interference, Remote Monitoring, Satellite**

RESUMEN

Este informe presenta el desarrollo de las prácticas académicas realizadas por la autora que tuvieron lugar en Groundspace, una start-up ubicada en Montpellier, Francia. Este proyecto se centra en la implementación de microservicios de software en Satmon, un nuevo producto en desarrollo diseñado para proporcionar asistencia a los operadores de satélites contra los problemas de interferencia resultantes de la coexistencia de múltiples constelaciones satelitales y asegurar el cumplimiento de las normas operativas. La metodología aplicada para el desarrollo de software se basó en la suite GitLab y el editor de código VSCode, que permiten tener un espacio de trabajo de desarrollo colaborativo para la gestión de tareas, revisión de código y despliegue. Todos los microservicios desarrollados utilizaron Python como lenguaje de programación principal, y los paquetes Skyfield y Astral para el cálculo de posiciones orbitales. El *sky scan* fue la principal característica desarrollada, que realiza un escaneo "ciego" durante una ventana de observación especificada para verificar la actividad de transmisión anómala y verificar el cumplimiento de los sistemas registrados. A pesar de las dificultades que se presentaron durante su desarrollo, tales como la falta de control manual del haz de la antena Starwin, las limitaciones de hardware y los problemas iniciales de pruebas de campo; el prototipo detectó con éxito satélites GEO y LEO, y además se logró la sincronización software-hardware, con mejoras continuas previstas para aumentar la funcionalidad y la experiencia del usuario. El proyecto prevé futuros avances y una presentación en el Congreso Internacional de Astronáutica de 2025 en Sydney, Australia.

***Palabras clave* — Interferencia de Radiofrecuencia, Microservicio, Monitoreo a Distancia, Satélite**

I. INTRODUCTION

In recent decades, Low Earth Orbit (LEO) satellite constellations have gained a lot of popularity among users due to their advantage in providing internet services, telephony, disaster management and defense applications [2]. As the number of constellations increase, so do the constraints on their operation. Therefore, there is a high demand for bandwidth but also a high scarcity of available radio frequencies and likelihood of interference. This leads to commercial, technical and regulatory obstacles [3].

In response to these challenges, some algorithms have been developed to implement radio-frequency interference (RFI) detection through Machine Learning (ML) [4, 5], but not so many applications have been able to actively monitor and avoid these interferences. Groundspace, a company that specializes in the development of software solutions for the ground segment of the latest generation of space missions, is currently working on a new software product called Satmon. This product is aimed to satellite operators and mainly focused on monitoring the Radio Frequency (RF) environment of large satellite constellations to detect anomalies and undesirable behavior, such as radio interference, that could lead to degradation of service for end users.

This work presents the evolution of the initial plan for the intern's participation in this project, focusing on the sky scan feature, and is organized as follows: First, Section I presents a brief introduction to the internship's project, which includes a general description of the product and objectives; Section III gathers the basic information required to better understand the implementation and context of the project; Section IV contains a more detailed description of internship's initial plan; Section V explains how this plan evolved throughout the internship, which objectives were achieved, which were not and why; Section E overviews the sky scan feature design, implementation and tests, which took most part of the author's work. Finally, Section VI summarizes the conclusions.

A. Satmon Product Overview

The exponential growth of high throughput LEO satellite constellations has led to many challenges, such as radio spectrum saturation and loss of most transmissions due to small beam size. This is why there is a need for an adaptive and precise monitoring approach to overcome the problems related to traditional monitoring stations [6].

Satmon will be a software monitoring solution whose primary objective is not to decode the signals but rather to characterize signal properties such as frequency, power, and modulation. The system will be able to monitor the positions in the sky where satellites transmit and check compliance with relevant licenses and coordination agreements.

With respect to the software, this system will be connected to an API (Fig. 1) through which multiple microservices will be provided for monitoring satellite spectrum, transponder activity, and more. It also allows the user to configure the frequency setup for all measurements based on the currently deployed node.

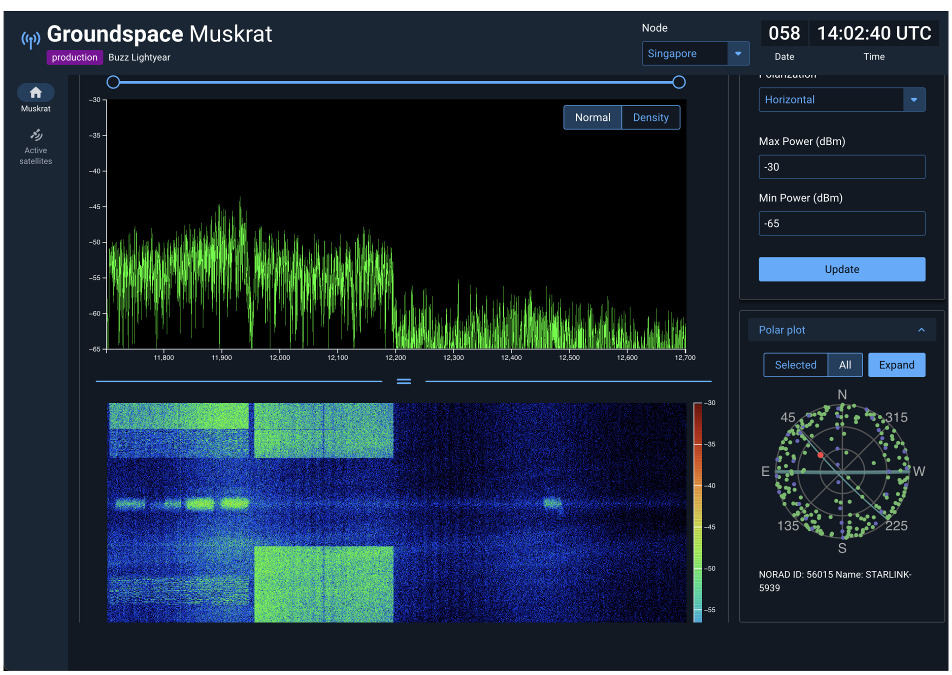

Fig. 1. Satmon User Interface (UI).

With respect to the hardware, Satmon is based on a distributed network of deployand-forget nodes, also known as antennas, communicated by the central node over a wired link, a 5G network or a suitable satellite connection. There's currently two available antennas for testing in Perpignan and Singapore, and another will soon be installed in Qatar. There will be three types of nodes: MUSKRAT, SIGRID and EPFDGuard.

MUSKRAT: This node consists of an electronically steerable (phased array) antenna and a digitizer connected to an edge compute unit. The antenna can track any LEO satellite visible in the sky and switch tracking to another in just a few milliseconds. This capability allows for the rapid scanning of all visible satellites, the accurate detection of anomalous transmissions, the visualization of live spectrum data, and the collection of unambiguous proof of unauthorized activity. There are three operating features available in MUSKRAT:

- Live spectrum: Allows the user to choose a satellite to monitor, and it will display its instantaneous spectrum and a waterfall plot showing how the spectrum changes with time. In Figure 2 a conjunction between satellites can be appreciated.
- Spectrum store: In this mode, previously recorded spectrum traces are available for later analysis.
- Active satellites: This feature shows on a real-time polar plot which visible satellites are currently transmitting signals towards the antenna. Its is connected to the Live spectrum when the user clicks on top of a satellite.

SIGRID: This node captures signals using a wide-angle antenna, enabling it to receive transmissions from all visible satellites within the frequency band of interest. By calculating channel power (with known channel boundaries for satellites) and correlating it with the satellite's distance, the source of each detected carrier can be accurately identified. Due to the antenna's low gain and the selected signal processing method, SIGRID cannot provide a usable live view of the spectrum or display a list of currently transmitting satellites. Instead, for each visible satellite pass, it generates a report listing the channels and the probabilities of transmission by the satellite.

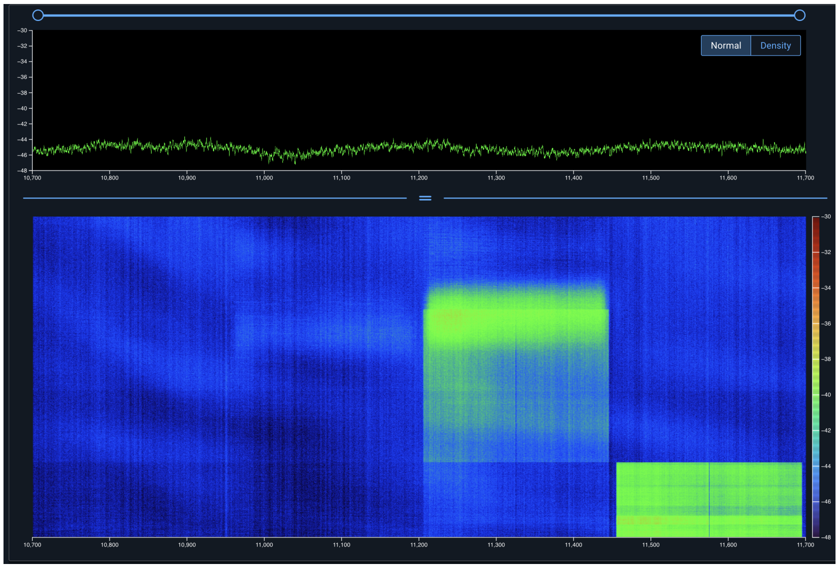

Fig. 2. Conjunction between Satellites Spotted on Waterfall Feature.

EPFDGuard: It is essentially a MUSKRAT node combined with a traditional high-gain antenna. The electronically steerable antenna is used to detect how many satellites of a given constellation are active simultaneously, while the larger high-gain antenna tracks as many active satellites as possible to build up a heatmap of the sky, recording the received power. The data is then compared to the license issued to the operator. Key elements to verify include ensuring that signal levels near the geostationary belt do not interfere with Geostationary Orbit (GEO) systems, as LEO satellites must avoid such interference.

II. OBJECTIVES

The objectives outlined by the author for her participation in this project are as follows:

A. General Objective

To design, develop and integrate software microservices based on orbital mechanics and Artificial Intelligence (AI) into Satmon's current Proof of Concept (PoC).

B. Specific Objectives

- To implement and test prototype space signal acquisition systems.
- To develop an algorithm to compute correlations between satellite orbits and received signal parameters.
- To develop signal recognition algorithms.
- To establish a centralized database of signal signatures.

III. LITERATURE REVIEW

A. Two-Line Element (TLE) Concept

A TLE set is a data format encoding a list of orbital elements (see Figure 3) of an Earth-orbiting object for a given point in time, known as the epoch. According to Dr. T.S. Kelso, together with NORAD's SGP4/SDP4 orbital model the object's position and velocity in space [7]. Because of this, it is widely used for satellite tracking.

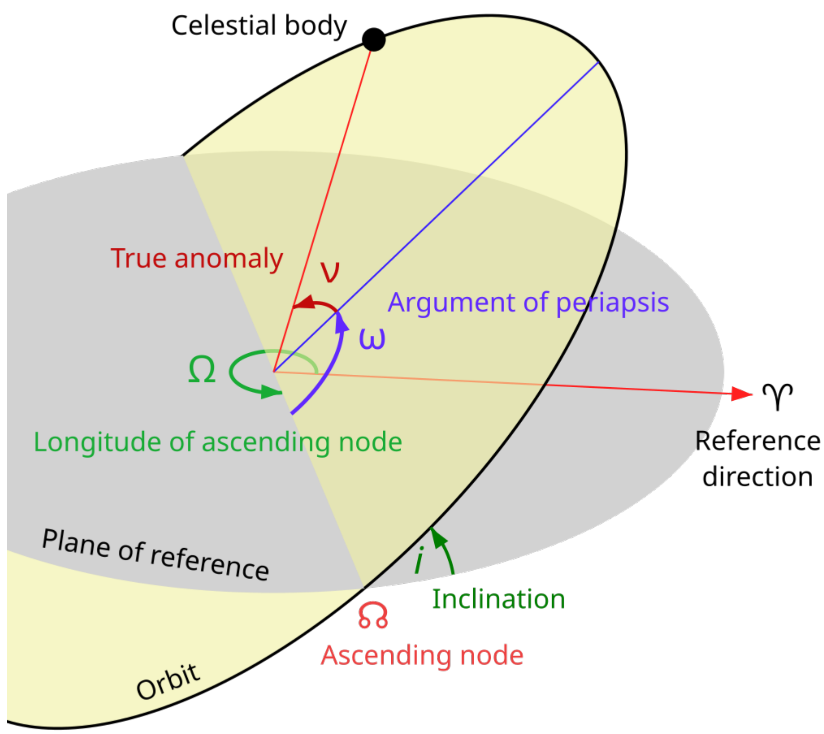

Fig. 3. Keplerian Orbit Elements Illustration.

This format includes the following elements [8]:

- **First line:**
  - Satellite number, also known as NORAD Catalog Number
  - International designator (launch year and piece of the launch)
  - Epoch date and Julian date fraction
  - Ballistic coefficient, also known as the first derivative of the mean motion (object's orbital decay rate)

  - Second derivative of the mean motion
  - Drag term, also known as radiation pressure coefficient (or BSTAR)
  - Element set number and checksum
- **Second line:**
  - Satellite number (repeated from Line 1)
  - Inclination in degrees
  - Right Ascension of the Ascending Node (RAAN) in degrees
  - Eccentricity
  - Argument of perigee in degrees
  - Mean anomaly in degrees
  - Mean motion in revolutions per day
  - Revolution number at epoch and checksum

Due to gravitational perturbations and atmospheric drag, these formats get deprecated and should be constantly updated in order to have precise results. For this project, the TLE is considered deprecated when the difference between the prediction's and the TLE's epoch is greater than 20 days.

B. Radio Frequency Spectrum

The Radio Frequency Spectrum (RFS) refers to the range of electromagnetic (EM) frequencies used for transmitting data wirelessly. Most spacecraft communication systems operate within the radio frequency range, which spans from 30 MHz to 60 GHz [9]. However, the usable spectrum is limited due to physical constraints like attenuation, which restricts the usable frequencies (see Figure 4). Additionally, the extensive use of these bands has led to congestion and challenges in acquiring licensing. As a result, this segment must be shared among all satellites, requiring coordination to prevent interference.[9].

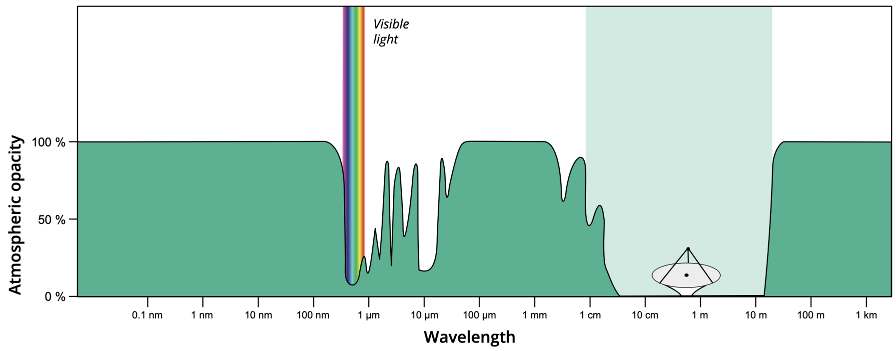

Fig. 4. Available Radio Spectrum.

Satellite communication systems, despite their many applications such as communications, internet, disaster management, and defense can be generally simplified into two segments: the ground segment and the space segment. The ground segment consists of one or more ground stations located on Earth, while the space segment comprises one or more spacecraft and their respective communication payloads [9].

In recent years, there has been a shift towards higher frequency communication systems to achieve higher data rates and increased availability. The most recent CubeSat deployments are moving towards the S and X bands, with Ka-band also being used for recent and future small satellite communications. Additionally, the Ku- and K-bands are becoming more attractive to SmallSat designers [9]. As an example, Figure 5 shows how the Ku-band is distributed between Terrestrial Fixed Services (TFS), Fixed Satellite Services (FSS) and Broadband Satellite Services (BSS).

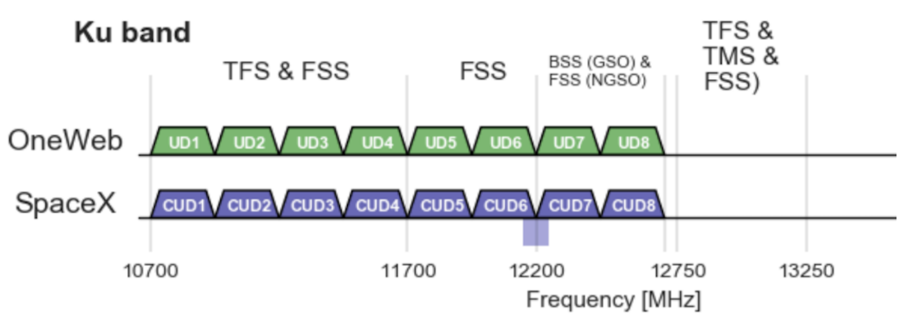

Fig. 5. Ku-band Services Distribution.

Previously, satellite-based telecommunication services relied on a few geostationary (GEO) satellites with mechanically-steerable parabolic antennas, which remain stationary in the sky. Traditional ground stations (GS) using these antennas offer high performance, reliability, and moderate cost but can only support a limited number of satellites within their narrow beam. The emergence of Low Earth Orbit (LEO) satellite constellations, consisting of thousands of satellites with dozens visible at any time, demands a much larger number of antennas to maintain coverage.

This limitation can only be compensated by an increase in the number of antennas. However, creating large arrays of antennas, or "antenna farms", is not feasible due to regulatory restrictions. Consequently, traditional reflector antennas cannot meet the management and control requirements of future ultra-dense LEO satellite constellations [6].

Current monitoring stations have not fully adapted to these advancements, making it difficult to track these constellations efficiently [6]. To address this and to overcome the increased free space loss caused by atmospheric and rain attenuation, higher-gain antennas need to be implemented [9].

Phased array antennas present a promising alternative for GS to overcome challenges associated with traditional systems. These antennas do not have any moving parts and generate an arbitrary number of beams by adjusting the phase between the signals sent to each emitter in the array, a technique known as beam-forming [10]. This allows them to support multiple satellites simultaneously with faster beam steering. It also offers better interfere mitigation, higher reliability, and lower cost [6].

C. Applicable Regulations to Bandwidth Usage

The space industry, though relatively young, has experienced exponential growth, making it necessary to create regulatory measures for its operations and production. In the case of the use of bandwidth by satellite systems, according to M. Bousquet:

> The spectrum is a finite resource, so its allocation requires effective and efficient coordination at global level. Radio regulations are necessary to ensure an effi-

cient and economical use of the radio-frequency spectrum by all communications systems, both terrestrial and satellite [11].

For this purpose, the International Telecommunication Union (ITU) is the main organization responsible to promote, coordinate, and harmonize the efforts of its members to fulfill these possibly conflicting objectives [11]. Consequently, for a satellite operator to legally employ a portion of the RFS, the entity has to obtain a national license to operate under available radio frequencies and orbital positions. However, depending on the frequency bands, the licence may not be exclusive, i.e. shared frequencies. Once this is secured, the State where the license was obtained, and in coordination with the ITU, has to apply follow-up for international coordination and registration processes to ensure interference-free operation [11].

IV. INITIAL INTERNSHIP'S PLAN

The applied methodology during the internship, which was proposed to be divided into four stages, relied on the GitLab suite, a platform for creating collaborative repositories and the VSCode editor for coding. The standard procedure for a new task consisted of the following steps:

1. Raise an issue on Gitlab assigned to the developer, with respect to the implementation, correction or tests requested.
2. From the issue, a Merge Request (MR) is created by the assignee. This corresponds to a feature branch of the related repository.
3. Once the assignee finishes the code, by pushing all changes from VSCode to Gitlab, a review from someone capable to merge into the main branch should be requested.
4. If there is any suggestion or thread to be solved, according to the reviewer, the assignee proceeds to make the corrections.
5. The assignee then creates individual tests for the developed functions (and the endpoint, if applicable) to verify code quality and type-object management.
6. Once all threads have been solved, the reviewer proceeds to merge the changes into the main branch.

This procedure can be applied to both collaborative and individual repositories, but it should be noted that the individual one is used only when a feature is been developed from zero. The collaborative repository, such as Products, is used once the feature is already defined and can take the "shape" of an endpoint inside the respective dependency.

To conduct tests with real data, a branch deployment on one of the available antennas is required. It is also necessary to coordinate the already-developed microservices responsible for hardware-software integration. Finally, GitLab tags are used to load the software onto the server where the antenna is located, and the test is run.

For the tasks, an initial plan schedule for the internship was established on the project's proposal. This was divided into five principal stages, which are presented in Figure 6 as a Gantt diagram:

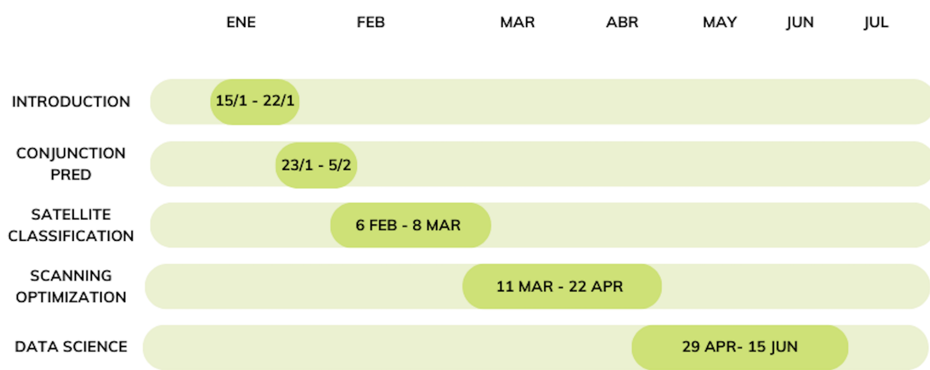

Fig. 6. Gantt Diagram of Internship's Schedule.

A. Introduction

Introduction to the platforms and tools used by the company for its projects, applied to a prediction exercise of apparent conjunctions between pairs of satellites for a configurable observation window.

B. Satellite Behavior analysis

At this stage, algorithmic approaches will be devised to select specific satellites within a constellation for monitoring purposes. This step involves the creation of predictive models that can anticipate satellite trajectories and behavior, allowing to concentrate monitoring efforts on critical objects and minimize analysis time.

Conjunction Prediction: Since several constellations use the same frequencies, there is a risk that insufficient coordination of activities could lead to radio interference and general degradation of service. Such occurrences can be detected by monitoring the behavior of the satellites when satellites "cross over" from a specific location. The intern will develop an algorithm to identify these apparent conjunctions so that a monitoring antenna can be programmed to verify that proper performance.

Satellite Classification: In order to effectively monitor the behavior of a constellation, it is necessary to understand whether a satellite is operational, inactive, in the process of deployment or parked as a spare for future use. The practitioner will investigate and implement an approach to identify:

- Orbital planes.
- Orbital elevation maneuvers.
- Drifted spacecrafts (no orbital corrections).
- Unusual spacecraft locations (e.g., in close proximity to another spacecraft)

The analysis will be performed on a historical TLE database spanning from the launch of the satellites.

C. Scanning Protocols Optimization

A comprehensive analysis of monitoring equipment requirements and capabilities will be performed to determine the most effective scanning techniques. The intern will create adaptive programming methods that take into account both the technical limitations of satellite tracking hardware and the dynamic nature of satellite orbits. The goal is to maximize coverage and data acquisition while reducing spurious measurements, e.g., LEO satellites temporarily appearing in a sky scan.

D. Data Analysis and Intelligence

The data collected by the distributed surveillance nodes will be added and analyzed using data fusion techniques. The goal is to transform the raw observational data into a cohesive intelligence framework, which can be leveraged to make informed decisions about radio spectrum management.

This analysis will extend to the calculation of correlations, specifically aimed at extracting detailed information about the configuration of satellite transmission beams, such as their shape and size. This phase will include the development of a user-friendly interface that presents the processed data in an understandable and actionable format for interested parties.

V. EVOLUTION OF INTERNSHIP'S PLAN

For the intern's participation in this project, she developed three individual repositories, four microservices and one implementation along with the Perpignan's antenna for MUSKRAT. Compared to the original plan, only stages 1 to 3 were completed (not necessarily in this order). This was due to significant client interest in the sky scan feature, so priority was given to achieving a quick implementation rather than optimizing it through AI.

Additionally, it was not possible to do all field tests that included the antenna's hardware due to some delays in scheduling and general calibration of the system. Nonetheless, given that the project is established as a proof of concept (PoC) and Satmon is a product in development, such delays were expected.

A. Conjunction Prediction

As presented on Section B, the first task that the intern made, as an introductory process to get to know the different platforms used in the company, was to create an individual repository on Gitlab to predict the conjunctions between ordered pairs of satellites from Starlink and Oneweb constellations for a given observation window. This feature was partially integrated by the intern into Products as an endpoint called "next starlink radio conjunction", part of the "events calculator" dependency. As inputs, the algorithm takes:

- The observer's location given by its latitude and longitude in degrees.
- The minimum elevation for the observation window in degrees.
- Time span in seconds.
- Minimum separation between satellites in degrees to consider it an apparent conjunction.

The identified conjunctions are presented as a list of string which specify in chronological order:

- The Starlink and Oneweb satellites NORAD IDs.
- The epoch in UTC at which these satellites are the closest.
- The Azimuth-Elevation (AzEl) coordinates of the first satellite at that epoch.
- The reached separation in degrees.

To propagate the satellite's position, the intern used the Skyfield library in Python. This package allows for the computation of the position of stars, planets and satellites in orbit in multiple formats, such as AzEl and Cartesian coordinates [12]. An existing polar plot application, which displays the positions of all visible satellites at a given epoch, was used to verify that the conjunction was spotted at the specified coordinates. This verification was successful.

B. Visibility Heat Map

For this task, the main focus was to get a visual reference of how are active satellites distributed around the globe and to give information about how to customize the GSs coverage according to that. The individual repository created for this purpose keeps track of the number of visible satellites from Starlink and Oneweb constellations for a specific observation window, then based on how frequent these satellites accumulate for a given location, it computes the percentage of coverage as a function of the number of antennas. Unlike the previous task, this feature was not integrated as an endpoint into Products. The inputs were:

- The latitude and longitude of an specific location in degrees.
- The minimum elevation in degrees.
- The time span for the observation window in hours.

In return, the output includes the following (see Figure 7):

- The maximum number of satellites from Starlink and Oneweb seen at any Earth's location for a specific period of time.

- The percentage of satellite's coverage of an antenna for an specific constellation, given the number of antennas used.
- A plot of the number of visible satellites seen from a specific Earth's location vs time.
- A heat map of the globe which assigns a color to the maximum number of visible satellites for a specific period of time seen from a specific Earth's location.

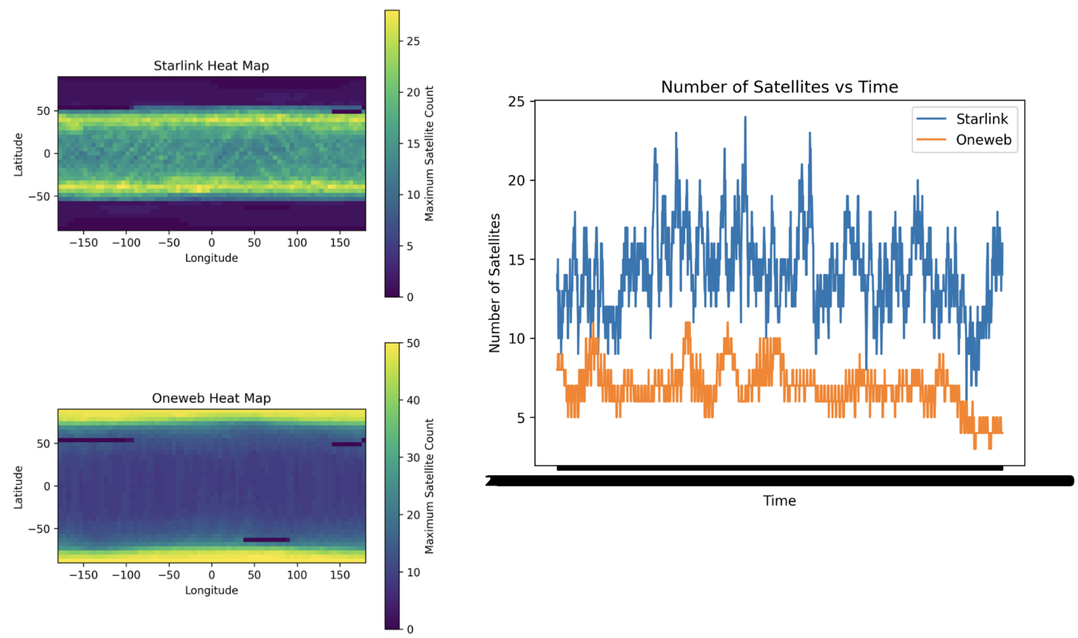

Fig. 7. Visibility Heat-map Outputs 3-hour Time Span Example.

Some concepts from the Satellite Classification stage (see Section B) were applied in order to consider if a satellite was operational or not, based on its orbit's inclination and altitude. Oneweb satellites are operational when they reach a minimum altitude of 1200 km [13]. A value of 1100 km was considered for tolerance purposes. On the other hand, for Starlink it depends on the shell they belong as shown in Table I:

There was no specific method to verify the results. However, OneWeb's heatmap aligns with its polar arrangement, prioritizing coverage of the Earth's poles, while Starlink's heat-map corresponds to its orbital planes arrangement, which is concentrated between and around latitudes ±52°. Regarding the number of satellites, it can be observed in Figure 7 that there

TABLE I
STARLINK SHELLS DEFINITION [1].

Shell	Inclination [º]	Orbital Altitude [km]	Nº Satellites
1	53.0	550	1584
2	70.0	570	720
3	97.6	560	348
4	53.2	540	1584
5	97.6	560	172

are typically around six OneWeb satellites and 15 Starlink satellites within the predefined observation window for Montpellier. This observation is consistent with the polar plot application previously used.

C. Sun's Position

The inclusion of a Sun's position endpoint into the Products was requested, marking the intern's first complete endpoint integration. The implementation was straightforward: the user inputs an epoch and a location (using the same units as in previous tasks), and the endpoint returns the corresponding Sun's AzEl coordinates. The main purpose of this task was to familiarize the intern with the structure of the Products repository at a dependency level.

Initially, the Skyfield package was used to compute the Sun's position. However, because the Ephemeris DE421 file is automatically downloaded each time the position is calculated [14], it was replaced with the Astral package. Astral was as fast as Skyfield and did not require downloading any additional files.

To verify the results, multiple websites designed to compute the Sun's position for a given epoch and location were used, and the results matched successfully.

D. Solar Conjunction

Similar to the previous task, another simple endpoint was integrated into Products in order to detect interference between the Sun and LEO satellites due to apparent conjunctions. For this case, the objective was to understand how to join different endpoint responses inside the same dependency. The endpoints developed in Sections A and C where reused for this. Consequently, it can be expected to have a combination between their corresponding inputs and outputs. This was verified successfully using also a combination between the methods used in the mentioned endpoints.

E. Sky Scan

The sky scan feature discretizes and blindly scans with an antenna a specified observation window to retrieve the received power for subsequent analysis and visualization. This process enables the identification of geosynchronous *zombie* satellites that interfere with other satellites'downlinks and also helps verify that registered systems comply with legal operational requirements.

Design and Development: The first version of the sky scan was developed in an individual repository, which only manages to generate the coordinates of each observation to be scanned according to the following inputs:

- LEO avoidance: True or False.
- GEO avoidance: True or False.
- Latitude and longitude of the antenna to be used, in degrees.
- Minimum elevation in degrees.
- Start time epoch to execute the sky scan in UTC, with the current one as default.
- Half-size of the antenna's beam in degrees.
- Overlap factor between the beams in percentage.
- Scan randomly: True or False.

These AzEl coordinates, which also correspond to the beam centers of the antenna, are generated using a hexagonal grid as a reference. The points are then randomized if requested. An illustration of this process is presented in Figure 8:

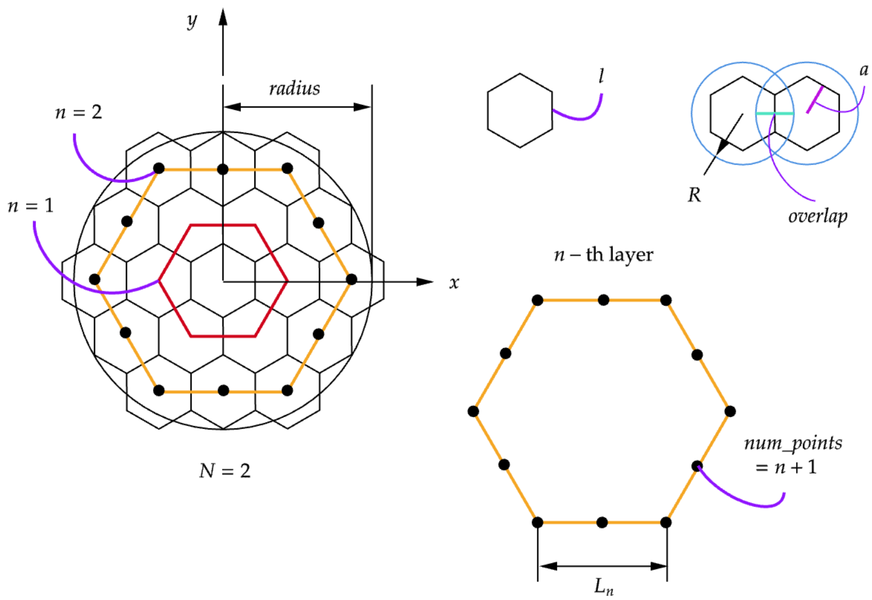

Fig. 8. Beam Grid Generation Illustration.

Where the variables are:

$x \rightarrow$ overlap factor in percentage

$a \rightarrow$ apothem of reference hexagon

$R \rightarrow$ beam's radius

$E \rightarrow$ minimum elevation

$N \rightarrow$ number of layers

$n \rightarrow$ layer's number

$l \rightarrow$ side length of reference hexagon

$L \rightarrow$ layer's side length

And the equations to obtain the dimensions for each layer are:

$$
\begin{aligned}
\text{radius} &= 90 - E \\
l &= 2 \cdot R \cdot \left(1 - \frac{x}{100}\right) / \sqrt{3} \\
a &= l \cdot \frac{\sqrt{3}}{2} \\
L_n &= l \cdot n \cdot \sqrt{3} \\
L_N &= \text{radius} \cdot 2/\sqrt{3} \\
N &= \text{ceil}\left(\frac{L_N - a}{2a}\right)
\end{aligned}
$$

The coordinates are then filtered according to the selected avoidance criteria. For each observation, an epoch is assigned every 0.1 seconds (Starwin antenna declared repointing speed), starting from the specified initial time, to determine whether any visible satellite falls within the beam. If the separation between the beam center and the satellite's position is less than or equal to 10 % of the beam's radius, the observation is discarded (see Figure 9).

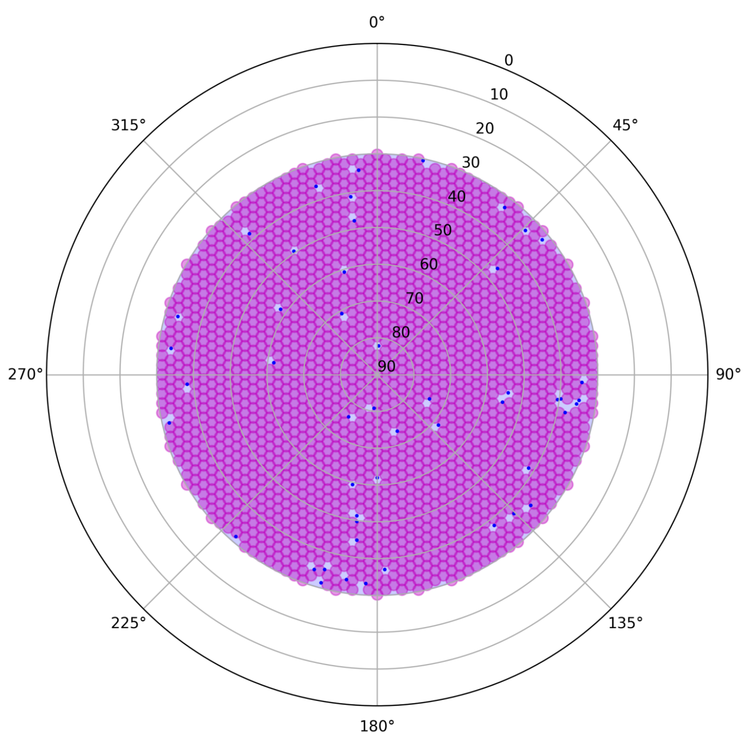

Fig. 9. Sky Scan Beam Coordinates Illustration.

Finally, the outputs of the sky scan are the following:

- List of dictionaries containing the observations'related information, which is the AzEl coordinate and its epoch.
- A GIF file showing the scanning process as a polar plot, where each beam pops up at the corresponding epoch inside the observation window.

It must be noticed that this version has no interaction with the antenna, so the only possible verification to do was the beam-satellite conjunctions as done in Section A. Additionally, a time step of 0.1 seconds implies that the sky scan ideally takes between 6 to 10 minutes.

Endpoint Integration into Satmon: The created endpoint within the *events calculator* dependency provides a list of observations similar to those in the individual repository, with the key difference that the resulting coordinates are not plotted. Additionally, other endpoints were created or reused to execute these observations using the antenna. This includes the *single sweep* endpoint from the BB60C Collector dependency, and the *start single satellite tracking* and *start single AzEl tracking* endpoints from the Starwin controller dependency.

The endpoints inside the Starwin controller are responsible of sending the steering commands to the Starwin antenna. The *start single satellite tracking* sends a TLE command while the *start single AzEl tracking* sends an AzEl command, but they both send a frequency command which is defined by the frequency limits of the selected band.

The single sweep endpoint is the one responsible for the getting the trace from the antenna for each observation. It configures the center and the span of frequency and then interpolates the retrieved power bins according to the number of bins, which in this case is 4096. Then, in the script the average of the 10 greatest signal levels out of the trace is computed. This number is used to assign the color of each beam in the sky scan plot.

Hardware Restrictions: In order to connect the developed endpoint with the antenna, a lot of requirements must be fulfilled. With respect to the available hardware, some tests were being done in parallel by another employee, concluding that:

- The Starwin antenna takes at least 400 ms to point to a specified location.
- The Starwin antenna also takes around 75 ms to change frequency.
- Tha Starwin antenna does not receive AzEl inputs yet, only TLEs.
- The collector takes at least 55 ms to get a single trace from an observation.
- For reliable results, the minimum elevation must be between 30° and 90°.
- Some frequency configuration commands are dismissed by the Starwin antenna.
- TLE commands should be sent three times to ensure the Starwin antenna receives it.

Prototype Implementation: A separate environment within the Products repository, called *prototypes*, was created to manage direct HTTP requests to the antenna's server in Perpignan. A "tunnel" connection was established between the intern's computer and the node, eliminating the need to use the Kafka protocol. This approach was implemented to achieve quick results and begin calibration.

The functions from the respective individual repository were adapted for the prototype's script to include the interaction with the antenna. For illustration purposes, the flux diagram of the complete sky scan prototype execution is shown in Figure 13. The key difference in this adaptation is that, after obtaining the observations, each one must be transformed into an artificial TLE due to the Starwin antenna's lack of manual beam control. This transformation ensures precise alignment to the desired coordinates. The elements that compose the TLE are computed as follows:

- Inclination 90°.
- Argument of perigee = 0.00001°.
- Eccentricity = 0.00001°.
- Orbit's height = 40000 km.
- Satellite's NORAD ID = 99999.
- Satellite's name = "DUMMY SAT".
- Semimajor axis: $R_e + h = 6378 + 40000 = 46738$ km
- Mean motion: $n = \sqrt{\frac{\mu}{a^3}} \cdot \frac{86400 \text{ s}}{1 \text{ d}} \cdot \frac{1}{2\pi}$, where $\mu = 398600{,}4418 \text{ km}^3/\text{s}^2$

- The mean anomaly and the RAAN are optimized in order to get the smallest error with respect to the given AzEl coordinate at the TLEs epoch.

Since one TLE conversion takes around 0.2 seconds, this adds approximately 10 to 20 minutes before the sky scan, which can be a disadvantage to the user. After this, a *for* loop is used to await each observation's epoch and then send two HTTP requests: one to the Starwin controller and another to the BB60C Collector endpoints. This is done to save each observation's signal level. With the AzEl coordinates and processed signal levels, the script returns the following outputs:

- A list of predicted satellites interfering with observations during the sky scan, when no avoidance is applied for "hot spots" identification.
- A list of signal levels in dB.
- A sky scan heat-map showing the location of each observation and its respective signal level as the color.

The analysis of these results focuses on identifying several key indicators:

- High and fixed transmitted power spots (bright yellow circles) close to the geostationary arc, which indicate areas of strong and consistent signal transmission from geostationary satellites.
- A "hot spot" around azimuth 165° and elevation 39°, corresponding to the fixed locations of very powerful GEO satellites, such as Eutelsat 28G (28.3°E), Astra 1M (19.2°E), Hotbird 13F (13°E), Intelsat 10-02 (0.8°W), and others [15].
- When applying LEO avoidance, powerful spots found outside the geostationary arc should coincide with some of the predicted jamming locations, serving as proof of a LEO transmission.

Initially, there were no discernible patterns in the sky scan's polar plot due to no proxy connection to the Starwin antenna's controller. As a result, the antenna was not actively steering and remained fixed on the geostationary arc, leading to the collecting just low noise signal. This issue is illustrated in Figure 10. Additionally, a time step of 0.1 seconds was still being used because hardware restrictions had not yet been identified.

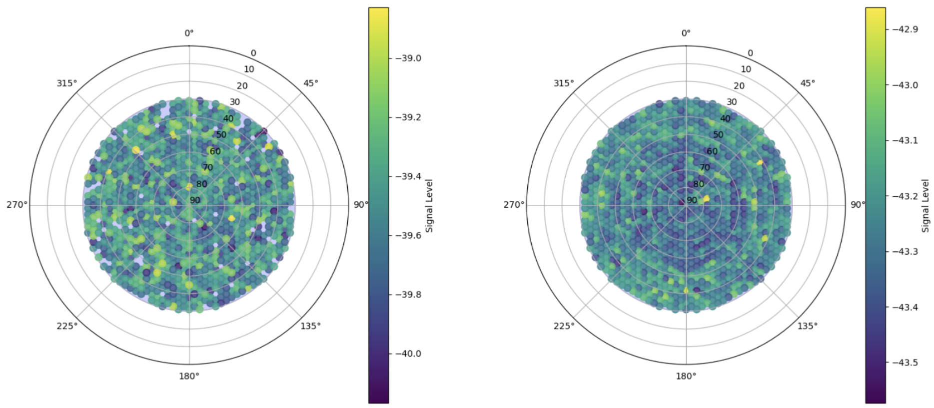

Fig. 10. Sky Scans without Starwin Controller Proxy's Connection (GEO-arc Noise).

For the next round of tests, the Starwin antenna crashed in the middle of the sky scan due to commands overflow. The antenna must have been stuck outside the geostationary arc, because the received signal level range was broader than the last one (see Figure 11).

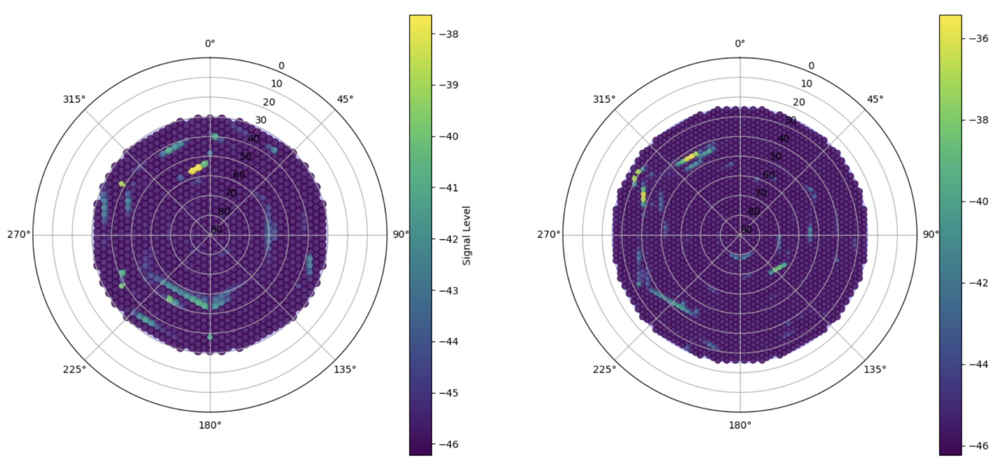

Fig. 11. Sky Scans Obtained with Starwin Controller Proxy's Connection.

After adapting to the hardware restrictions described in the previous section, a minimum time step of 0.35 seconds was established. However, discrepancies were observed between the predicted and actual execution times of the sky scan, indicating that there is still an accumulation of commands. This issue was resolved by increasing the time step to 0.5 seconds.

Figure 12 shows the key expected features on the polar plot, such as strong GEO and apparent LEO satellites, but further analysis is required to confirm the adequacy of the avoidance implementation. of this figure, none of the detected transmissions (yellow spots) align with the predicted beam-LEO conjunctions (red dots).

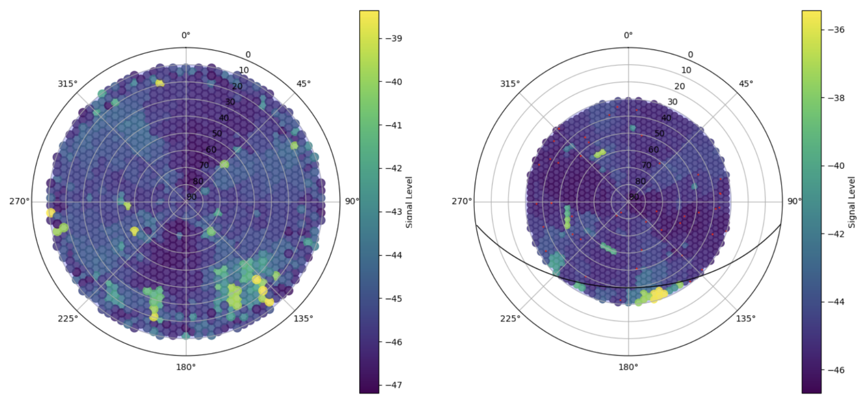

Fig. 12. Sky Scan with Applied Hardware Restrictions.

F. Field Tests

Toulouse Test at Kratos: A field test was scheduled for May 17 at Kratos, a Groundspace partner with a branch in Toulouse, to test the first version of the sky scan with a newly installed Ka-band antenna. The main advantage of this antenna is its ability to receive AzEl coordinates. Unfortunately, the antenna was not fully set up at the time and was unable to detect any geostationary satellites, which is essential for system calibration and overall operation. Consequently, the test was postponed for a future opportunity.

Perpignan Test: Due to the unsuccessful attempt of the previous test, an alternative approach was implemented: setting up a "tunnel" connection to Perpignan's node and accessing the required microservices through HTTP requests to execute the sky scan. Consequently, all tests were conducted using the prototype until the end of the internship. It should be noted that this was a quick implementation designed to test the developed code and does not represent the final version within Satmon. It might take around 4 to 6 months for Starwin to achieve manual beam-control, before executing a skyscan as it was intended.

G. Future Work

The author will continue to develop the sky scan as an employee of Groundspace in order to continue with its integration with the antenna's hardware and contribute to the frontend's development, once the final version is approved. The next tasks to be done are the following:

- Since the sky scan does not require a change in frequency, there should be an auxiliary endpoint to sent one single frequency command at the beginning.
- LEO avoidance must be checked, which means that most yellow spots must coincide with red dots.
- The sky scan endpoint must be updated to support Kafka in order to interact with the Collector and the Starwin controller without HTTP requests.
- AzEl commands should be used instead of the TLEs once the Starwin antenna supports them as an input.
- Tasks must be coordinated through an scheduler that uses the epochs of each beam to do the respective observations.

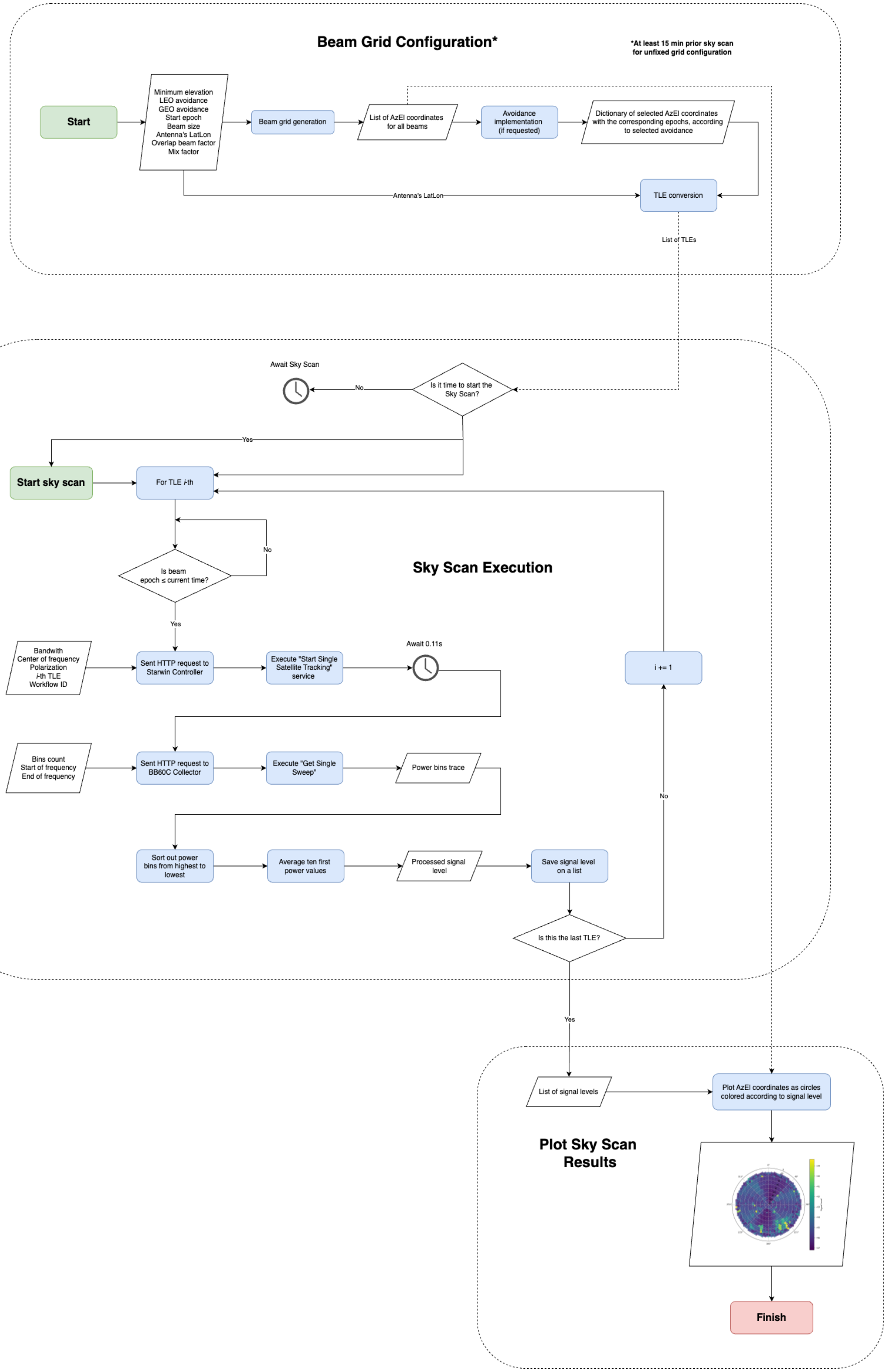

Fig. 13. Sky Scan Flux Diagram.

VI. CONCLUSIONS

The internship successfully contributed to Satmon's Proof of Concept and allowed the author to apply her knowledge in orbital mechanics and programming in a industrial environment. The key achievements include:

1. The intern developed four microservices and integrated one of them with Perpignan's antenna for the MUSKRAT project. And despite deviations from the original plan, the prioritized sky scan feature showed promising results in identifying signal interference.
2. The appropriate selection of the time step enabled the successful detection of powerful GEO and some apparent LEO satellites, as well as achieving hardware and software synchronization.
3. The progress made in the sky scan will lay the foundations for the implementation of required Kafka protocols and make required modifications to optimize LEO avoidance.
4. The intern plans to continue contributing to the project with Groundspace and present a paper on the final implementation at the International Astronautical Congress 2025.

After the tests, it was concluded that the sky scan performance largely depends on the type of antenna being used. The time required to execute a sky scan depends on the type of input it receives, steering speed, commands reception, etc. However, to cover the entire sky would be theoretically impossible to execute in less than 10 minutes. Despite this limitation, the feature demonstrated significant capabilities.

Looking ahead, the intern will continue to collaborate with Groundspace on this project as a part-time employee. The next steps involve refining the sky scan feature and its frontend's development supervision. Also, a paper on the final implementation of the sky scan at the International Astronautical Congress 2025 in Sydney, Australia. This presentation will provide an opportunity to share insights and advancements with the broader community, highlighting the project's impact and future potential.

In summary, the internship has made substantial contribution to Satmon's development and was accomplished according to the company's interests. Also, laid the foundation for the sky scan's standard endpoint integration through Kafka, despite some initial limitations. The outcomes demonstrate the feasibility and potential impact of this main feature, paving the way for future advancements and implementations. The successful detection of satellitesáctivity and the effective synchronization of hardware and software are notable achievements that reflects this. The continued collaboration and upcoming presentation at gathers the significance of her contribution into this project.

REFERENCES

[1] T. Sesnic, "Starlink Group 6-15: Falcon 9 block 5," Jul 2023. [Online]. Available: https://everydayastronaut.com/starlink-group-6-15-falcon-9-block-5-2/#:~:text=The%20first%20orbital%20shell%20of,22%20satellites%20in%20each%20plane.

[2] J. Zhang, Y. Cai, C. Xue, Z. Xue, and H. Cai, "Leo Mega Constellations: Review of development, impact, surveillance, and governance," *Space: Science amp; Technology*, vol. 2022, Jan 2022.

[3] J. Tang, D. Bian, G. Li, J. Hu, and J. Cheng, "Resource Allocation for LEO Beam-Hopping Satellites in a Spectrum Sharing Scenario," *IEEE Access*, vol. 9, pp. 56 468–56 478, 2021.

[4] A. Camps, J. Gourrion, J. M. Tarongi, M. Vall Llossera, A. Gutierrez, J. Barbosa, and R. Castro, "Radio-Frequency Interference Detection and Mitigation Algorithms for Synthetic Aperture Radiometers," *Algorithms*, vol. 4, no. 3, pp. 155–182, 2011. [Online]. Available: https://www.mdpi.com/1999-4893/4/3/155

[5] G. Quintana-Diaz, D. Nodar-López, A. González Muíño, F. Aguado Agelet, C. Cappelletti, and T. Ekman, "Detection of radio interference in the UHF amateur radio band with the Serpens satellite," *Advances in Space Research*, vol. 69, no. 2, pp. 1159–1169, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117721007778

[6] G. He, X. Gao, L. Sun, and R. Zhang, "A Review of Multibeam Phased Array Antennas as LEO Satellite Constellation Ground Station," *IEEE Access*, vol. 9, pp. 147 142–147 154, 2021.

[7] T. S. Kelso, "CelesTrak: "FAQs: Two-Line Element Set Format"." [Online]. Available: https://celestrak.org/columns/v04n03/

[8] NASA, "Human Space Flight (HSF) - Realtime Data," Mar 2000. [Online]. Available: https://web.archive.org/web/20000301052035/http://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html

[9] Feb 2024. [Online]. Available: https://www.nasa.gov/smallsat-institute/sst-soa/soa-communications/#:~:text=Radio%20frequency%20communications%20for%20spacecraft,crowding%20and%20challenges%20acquiring%20licensing.

[10] Cadence, "Phased Array Antennas: Principles, Advantages, and Types." [Online]. Available: https://resources.system-analysis.cadence.com/blog/msa2021phased-array-antennas-principles-advantages-and-types

[11] J. N. Pelton, S. Madry, and S. Camacho-Lara, *Handbook of Satellite Applications.* Springer, 2017.

[12] B. Rhodes. [Online]. Available: https://rhodesmill.org/skyfield/

[13] E. Group, "Our Network." [Online]. Available: https://oneweb.net/our-network

[14] J. P. Laboratory. [Online]. Available: https://pypi.org/project/de421/

[15] LyngSat, "Europe, Africa Middle East." [Online]. Available: https://www.lyngsat.com/europe.html