

SpaceOps-2025, ID # 128

## **Ground Station as a Service: A Cost-Effective and Scalable Solution for Space Mission Management**

**Jamie John Moses**

South African National Space Agency, Space Operations, Republic of South Africa, jmoses@sansa.org.za

### **Abstract**

**Ground Station as a Service (GSaaS)** offers a pay-as-you-use model for satellite tracking, telemetry and command (TT&C), launch support, in-orbit testing, mission control, and space navigation services. This model, provided by an Earth ground station, uses Radio Frequency (RF) antennas to interface a client with their payload, ensuring seamless communication and data transfer. This concept is being developed at the South African National Space Agency’s (SANSA) Space Operations programme. Whether a spacecraft is in its launch or orbit phase of operation, multiple ground station contact points will be required for continuous communication. This model aims to provide a single point of contact for the management and expansion of ground station support services for space missions.

This model is achieved by the integration of the following key components:

**Client Booking Platform**, a software application that allows users to specify their mission requirements, make reservation of systems, access data, and monitor ongoing missions acting as a digital mission control.

**Ground Station Platform**, another software application that handles the management of interlinking systems providing mission planning and scheduling from requirement inputs and available systems.

**Antenna Control System**, this system facilitates the setup of tracking the payload based on the schedule, configuring tracking data, RF equipment configuration, and data flow chains which incorporates data ingestion is handled by High Data-Rate Receiver (HDR) systems reserved as per the mission configuration.

**Centralised Database**, a database acting as the central storage element, configurator and command structuring device - storing mission information, system architecture information, etc., in an Object Relational Mapping (ORM) format for both system control and client feedback. By integrating these components, GSaaS ensures efficient, secure, and scalable ground station services, revolutionizing the way satellite missions are managed and executed. This paper explores the integration of hardware infrastructure with a software platform for the management of the key components for the model. The hardware infrastructure, referring to the antenna systems and RF equipment within it, is used for the purpose of tracking client spacecraft, such as antenna control units, baseband units, frequency converters, amplifiers, signal analysers, etc. A software-based prototype was created to be able to achieve the model’s goal of monitoring and control of hardware as well as managing client antenna reservations. This prototype not only proves that the model is achievable with the key components and the Application Programming Interface (API) between them but also showed the scalability for both client and ground station operations.

**Keywords:** Ground station, database, spacecraft, platform, application, software

### **Acronyms/Abbreviations**

Application Programming Interface (API)

Client Booking Platform (CBP)

Centralised Database (CDB)

Entity Relationship Diagrams (ERD)

Ground Station as a Service (GSaaS)

GSaaS API Server (GAS)

Ground Station Platform (GSP)

Graphical User Interface (GUI)

High Data Rate Receiver (HDR)

Integrated Development Environment (IDE)

Object Relational Mapping (ORM)

Operating System (OS)

Radio Frequency (RF)

South African National Space Agency (SANSA)

Software Development Life Cycle (SDLC)  
Transmission Control Protocol/Internet Protocol (TCP/IP)

## 1. Introduction

Companies, governments, or academic institutions that require the use of ground stations for support of their space missions, no longer require their own dedicated infrastructure for communication to their spacecraft. Just as how an institution uses the services of launch vehicles and does not develop their own launch capabilities, the same principle is used by ground station. As investigated by Gal-Edd et al. [1], low-cost ground station implementation is not immediately evident. There are however many aspects subject to cost reduction within a spacecraft space mission, even so, considering that NASA has decided to shift towards utilising commercial ground stations for operations, outlined by Gal-Edd et al. [1], a single platform to manage this utilisation would be beneficial.

The primary advantage of GSaaS is cost-saving, thereby increasing accessibility into the space operations sector. By utilizing this service, clients can avoid the substantial capital expenditure associated with building and maintaining their own ground station infrastructure. Additionally, GSaaS offers flexible scaling, allowing clients to adjust their operations without being constrained by their infrastructure. Mission security is another key benefit, with the service providing reliable connections through redundant systems and robust cyber security measures to protect data, which would already be in place at an established ground station.

The software aspect of the model comprises of two applications for user operation: The Client Booking Platform (CBP) and the Ground Station Platform (GSP). The development of this is explored in further detail in the development and operational methodology. The model is expandable by incorporating additional established ground stations, conforming to a single standard of operation, into a marketplace platform to increase capacity without further investment into new infrastructure. With an expansion of ground stations spanning multiple regions, the process can be achieved from a single platform throughout the life cycle of a space mission.

A visual representation of this model is given by Figures 1-3.

Consider the ground track formed by a satellite’s orbit, as shown in Fig. 1. The client of the satellite would require a ground station located somewhere within region of the track, represented by the blue line.

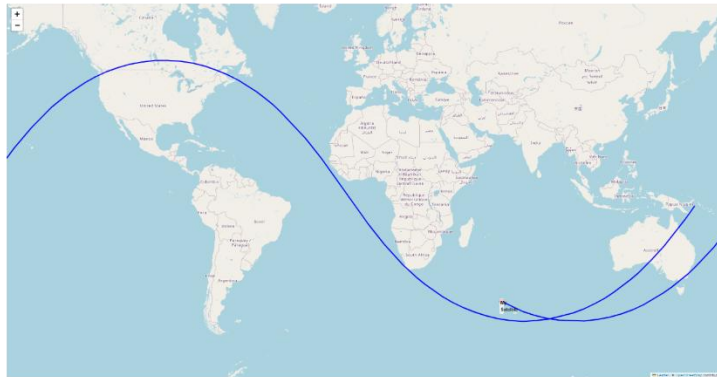


Fig. 1. Arbitrary Satellite Ground Track

As represented by the red dots in Fig. 2, the ground stations should effectively be within relatively close proximity of one another for communication to the satellite to occur. The circle represents the contact range of the ground station, essentially dictating the area where a satellite will be within communication range.

“Copyright 2025 by the Canadian Space Agency (CSA) on behalf of SpaceOps. All rights reserved. One or more authors of this work are employees of the government of South Africa which may preclude the work from being subject to copyright in South Africa, in which event no copyright is asserted in that country.”



Fig. 2. Satellite Ground Track with Client Ground Station Locations

Let's say that the client owns three out of the nine ground stations, shown in Fig 2., and the rest of the ground stations are part of a GSaaS network, if the satellite requires frequent contact to ground stations for operations, the client will be able to utilise the service and scale their operations with increased ground station capabilities. The expanded network of ground stations is represented in Fig. 3, which increases the amount of contact along the satellite ground track and shows why this model is essential.



Fig. 3. Expanded Ground Station Network using GSaaS

## 2.1 Literature Review

### 2.1.1 Cloud Computing and GSaaS

In contrast to the work done by Sarkarati et al. [2] and Safarik & Schuenemann [3], the model described by this paper will be that of a traditional network architecture. The reason for this choice lies not only with how common traditional network architecture is but also for reliance on already existing infrastructure instead of seeking out third party services. If startup costs are considered on networking, cloud computing has lower startup costs [4], however this paper investigates the case of existing infrastructure and that currently utilised in the test environment.

### 2.1.2 Resource Scheduling and Monitoring

Liu et al. [5] identifies the need for real-time monitoring of system telemetry. A database approach to system monitoring will not meet the real-time response times but can be achieved by API implementations as indicated in section 4.2.2.4. The management of scheduling however can meet the requirement as schedule changes do not require real-time monitoring but can achieve high polling rates without straining the database server's resources. Liu et al. [5] solves the problem of physical hardware through the virtualisation of key components in ground station systems, but this paper seeks to effectively manage available hardware and infrastructure.

## 3. GSaaS model

### 3.1 Benefits of GSaaS

#### 3.1.1 Reduced capital expenditure on startup

“Copyright 2025 by the Canadian Space Agency (CSA) on behalf of SpaceOps. All rights reserved. One or more authors of this work are employees of the government of South Africa which may preclude the work from being subject to copyright in South Africa, in which event no copyright is asserted in that country.”

The most evident cost reduction comes from renting time on satellite antenna infrastructure of existing ground stations. As mentioned on the AWS Ground Station website [6], a cost saving of up to 80% can be achieved by outsourcing the ground station operation to existing networks. Depending on the phase of the spacecraft operation, and the orbital characteristics the required contact frequency can vary. In the case where one chooses building the infrastructure, the worst-case scenario in terms of cost, is the need for continuous coverage. It would be impractical to build a ground station for a single space mission and considering that the infrastructure has running operational costs associated with it this impracticality is further evident. These costs can include the cost of staff salaries needed at the ground stations and perform operational or maintenance tasks, the cost of RF equipment and its maintenance costs [1]. To alleviate oneself from these varying costs, the option of outsourcing the ground station is key.

### *3.1.2 Scalability of ground station network*

#### *3.1.2.1 Requirement of continuous coverage to spacecraft*

From the perspective of the client that requires multiple ground station contact points, a cohesive way of ensuring the communications aspect is covered can be done by reserving ground station antenna systems part of a GSaaS network. Space mission planning becomes easier as the selection of ground stations along required contact points can be done from a single platform.

#### *3.1.2.2 Ground station network expansion*

A ground station operator that wants to expand their network can benefit by the GSaaS model with ease of integration thanks to the standardisation of the entity attributes and use of a central database structure. It is as simple as loading a new asset under a new ground station location within the platform.

### *3.1.3 Network and power system redundancy*

Established ground stations require reliable electricity sources to power the many systems on a site. Because a ground station’s revenue is dependent of dedicated contact time to a spacecraft, on-site backup generators are usually in place as redundancy which also benefits network reliability.

In this model, the network architecture is directly linked, where the network can be extended to additional ground stations on the GSaaS network as well as the client of the spacecraft see Appendix A. This architecture plays a vital role in the workings of the client and ground station platforms.

## *3.2 Developing the GSaaS model*

The system comprises of two main applications per ground station-client pair: The CBP and the GSP. In any contact between ground station and spacecraft, there will be one pair of platforms operational during this event. These two platforms are the operable applications for the GSaaS model, but a pair forms in any interaction of scheduling for a ground station’s antenna, illustrated in Appendix B. The pair forming occurs for the following functionalities: Ground station antenna reservation requests, request evaluation, antenna system monitoring & control, and tracking execution.

The platforms cannot access the database directly, the interface between the platforms and the database is the GSaaS API Server (GAS). GAS acts as the Application Programming Interface (API) server handling the requests for data received from each platform type. The deployment of this server will be at any ground station network operator’s main facility as shown in Appendix C. This application architecture makes it easy to give central control point to the ground station network owner. Ownership and access rights can be managed with user profile assignment by the ground station network operator.

Visualisation takes place on Graphical User Interface (GUI) dashboards within each of the platforms and their data is retrieved through communication via the GAS. The key entities of this model are related and developed using an Entity Relationship Diagram (ERD), shown with the basic attributes in Appendix D. All data within the model are managed and stored in the CDB.

## **4. Material and methods**

### *4.1 Materials*

The platforms developed were developed on a Windows Operating System (OS) laptop computer using the VSCode Integrated Development Environment (IDE) in the Python programming language with the Python toolkit, PyQt. Database operations and development was done using PostgreSQL and pgAdmin, the object-relational database management system and its GUI admin tool in conjunction with SQLAlchemy, the database toolkit and object

“Copyright 2025 by the Canadian Space Agency (CSA) on behalf of SpaceOps. All rights reserved. One or more authors of this work are employees of the government of South Africa which may preclude the work from being subject to copyright in South Africa, in which event no copyright is asserted in that country.”

relational mapper for Python. Python’s socket library was utilised for the TCPIP communications between platforms. Diagrams were designed in the Draw.io application. Operation tests and developments were conducted on an X-, and S-band satellite antenna system with a Cortex HDR, located at the Hartebeeshoek ground station of the SANSA Space Operations programme in Gauteng, South Africa.

## 4.2 Methodology

### 4.2.1 Development methodology

#### 4.2.1.1 Defining the entities and their attributes

For any given ground station and client, every physical entity within the process must be identified. We will use a single ground station-client pair, from Appendix B, to illustrate the identification process with a basic list of attributes for each entity.

Starting with the ground station, which can be broken down as follows: Antenna system, RF equipment, and HDR. We can now define each entity’s attributes. The antenna system can have attributes such as antenna name, antenna ID, location (station ID), uplink frequency, downlink frequency, demodulation scheme, etc., and this can be done for every other entity in the ground station. The client has the attributes of client name, client ID, contact email, organization, and so on. It also comprises of the spacecraft entity with its own attributes.

We continue in this manner until all entities in a ground station-client interaction has been identified with its respective attributes.

#### 4.2.1.2 Entity relationships and schema

An important point to consider is what the other entities would have as attributes, so the attributes chosen must be relatable, to some extent, to other entities. As an example, the antenna frequency attributes would relate to the spacecraft frequency attributes and can therefore be used in searching logic.

Once the components can be fully itemised, we can assign them as entities within our database management system, see Appendix A. In this case we assigned the entities of this model to be stations, antennas, clients, spacecraft, and schedules. Within the ERD we create the relationship between entities and in cardinality notation we will have the relationship of many antennas to one ground station, one client with many spacecrafts, and so on until we defined all relationships available.

Each entity will be represented as tables in the database with columns of attributes, as described in section 4.2.1.1. The complete schema of the tables in pgAdmin can be viewed as an ERD in Appendix D.

#### 4.2.1.3 Network architecture

The configuration of the network environment the model will be within, as mentioned in section 3.1.3, is a direct link of the traditional network architecture. This means that the client would have a direct internet connection to the network devices of the ground station as shown by Appendix B. As discussed by Blessing and Olusegun [7], a traditional network is a the most common architecture for networking, so the chances of the linking of multiple traditional networks is high. This network architecture doesn’t just allow the platforms, server, and database to all communicate using the Transmission Control Protocol/Internet Protocol (TCP/IP) communication protocols but also makes it possible for clients to have a direct connection for data streaming from devices like the HDR.

#### 4.2.1.4 Operation process and data flow

Now that the environment is set, the procedure for the system needs to be outlined and this will provide the algorithms of the software. A more detailed explanation of these algorithms is given in the operation methodology section to follow.

It is imperative to keep the flow of data in mind as well so that when a specific operation must take place, such as displaying a ground station antenna availability, we know what data will be required for the database query.

### 4.2.2 Operation methodology

As expressed in section 3.2, we have the functionalities that need developing: antenna reservation requests, request evaluation, antenna system monitoring & control, and tracking execution. In the scope of this paper, we will not extend to further API development other than that of the operations of GAS. Therefore, only scheduling request & monitoring, and request evaluation will be detailed.

#### 4.2.2.1 Schedule monitoring

Managing a schedule for a ground station requires a robust categorisation of the hardware and infrastructure in use, this was outlined in sections 4.2.1.1 and 4.2.1.2. We used the ERD tool of pgAdmin for creating the schema of the database. Now the Python library, SQLAlchemy, can be used as an ORM. This tool provides a standard interface and will allow the database tables to be translated into Python classes, and vice versa. The library will enable the movement of data between objects and the database [8], and therefore this logic will be managed by the GAS.

For monitoring to occur, the schedule information must be acquired by the platforms. Because the database logic is handled by GAS, the GSP and CBP will send a request to GAS. The request can take any form, a good format to use can be the JavaScript Object Notation (JSON), which is a common data-interchange format in web applications and has a readily available Python library, json [9]. The request made to GAS must contain the relevant asset ID or name, as shown in Appendix D, the schedule table contains the client, antenna, and spacecraft ID to which the event is linked.

The request gets interpreted by GAS and is associated with a mapped class that links to the database table. Using the SQLAlchemy Query API, the type of data request can match an available query function, such as filter, get, etc. [4]. A query can take the form of ‘get all schedules with status confirmed filtered by spacecraft ID’. The response will be returned to GAS which can then be returned to the querying platform and displayed in the dashboard.

#### 4.2.2.2 Antenna reservation request

This function, available to CBP for mission support, can be done to secure a reservation of a particular antenna system. Similarly to section 4.2.2.1, a query structure gets populated by some form of user interface on the platform, this can be a table, text, etc. Required info such as antenna name, antenna ID, start and end time would be necessary. This structure gets formed into the query structure that will be sent to GAS for interpretation. Before the request gets finalised, a schedule monitor request must be made to validate the availability of the selected antenna against the desired time. After the validation completes, this query is converted by GAS into an insert operation into the database with a pending status. The status attribute is important as it dictates whether an antenna gets reserved or not.

#### 4.2.2.3 Reservation request evaluation

The process of requesting a reservation of an antenna system starts on the GSP where the platform makes a request to GAS to return all schedules with a pending status. The returned list can be evaluated by an operator by either accepting or declining the request. For an accepted request, GAS will perform an update operation on the status attribute of a schedule event, changing it from pending to confirmed. Similarly, the declined request gets the status attribute changed from pending to rejected. Completing the evaluation process.

#### 4.2.2.4 Additional operations

As previously mentioned, sections 4.2.2.1 – 4.2.2.3 do not cover every possible operation within the GSaaS model. Events such as maintenance reservation, antenna system configuration commands, antenna tracking execution commands, ground station and antenna addition, client and spacecraft addition, are but a few of the possible extended operations that can be integrated into the model. Most importantly in a business model, the process of invoicing should be defined too. As shown by the simplicity of the previous operations, to generate an invoice, GAS will be given a request to return schedules with a completed status where the chargeable time can be obtained from the scheduled start and end time. Of course, for a more accurate representation, additions to the model can be made to store logs of antenna systems where split-second level of timing accuracy can be returned.

## 5. Results and discussion

The development of a prototype, using the ERD structure shown in Appendix D, proved that a basic model can be achieved with even a small number of entities and attributes. Because of the nature of the network architecture, the expansion of client and ground station platform operators is achieved with Python’s socket library for communication between applications on network protocols such as TCP. The management of assets within the ground station operation is effectively managed from a central point through API control if the information added is accurate. In Appendix E, we can see the simplicity of visualising the data from the database in table form, noting that the data is all fabricated for testing purposes and does not represent actual space missions. Unique ID of CBP and GSP platform usage was manually set during testing, but this can be centrally managed through the assignment by IP ranges, which would differ from station to station and client to client.

*“Copyright 2025 by the Canadian Space Agency (CSA) on behalf of SpaceOps. All rights reserved. One or more authors of this work are employees of the government of South Africa which may preclude the work from being subject to copyright in South Africa, in which event no copyright is asserted in that country.”*

## **6. Conclusions**

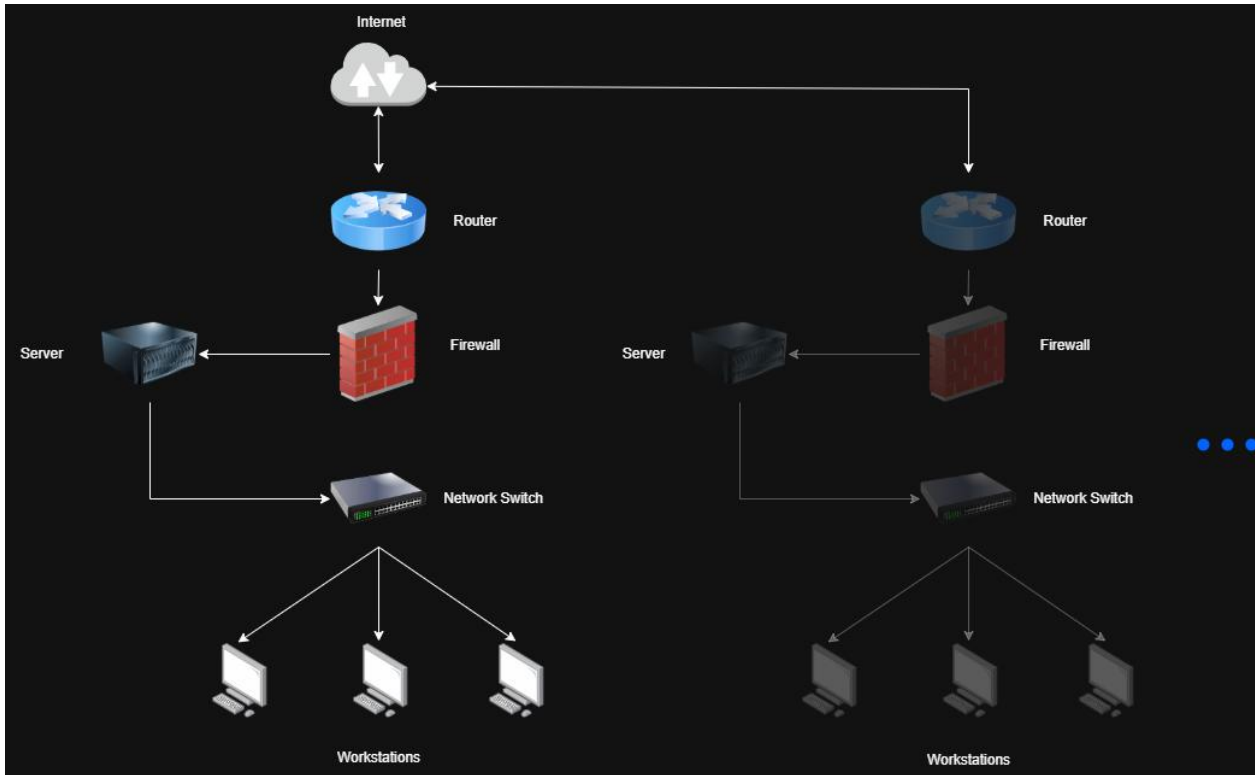
There are many available service providers offering some form of GSaaS model, but a proof of concept has been developed that shows that the GSaaS model requirements are met using traditional network architectures directly linked via the internet, provides management of assets through the use of a centralised database with accompanying API server for handling queries, and user interface platforms for the visualisation of the scheduling information but expandable due to its modular structure. Further operational expansion can be achieved by the integration of hardware control APIs with similar operational methodology as GAS.

## **Acknowledgements**

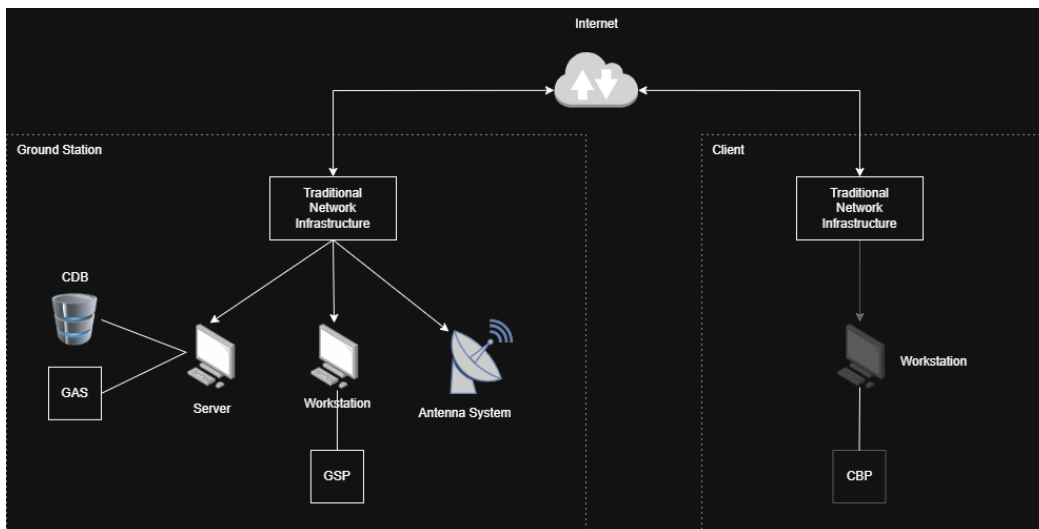
I would like to extend my gratitude to the South African National Space Agency for affording me this opportunity and to my colleagues at Space Operations who have provided me with guidance. To my partner, Tineelia Rajoo, who has provided me with continuous support throughout this endeavour, I am forever grateful.

*“Copyright 2025 by the Canadian Space Agency (CSA) on behalf of SpaceOps. All rights reserved. One or more authors of this work are employees of the government of South Africa which may preclude the work from being subject to copyright in South Africa, in which event no copyright is asserted in that country.”*

### Appendix A (Simplified, Expandable Traditional Network Architecture)

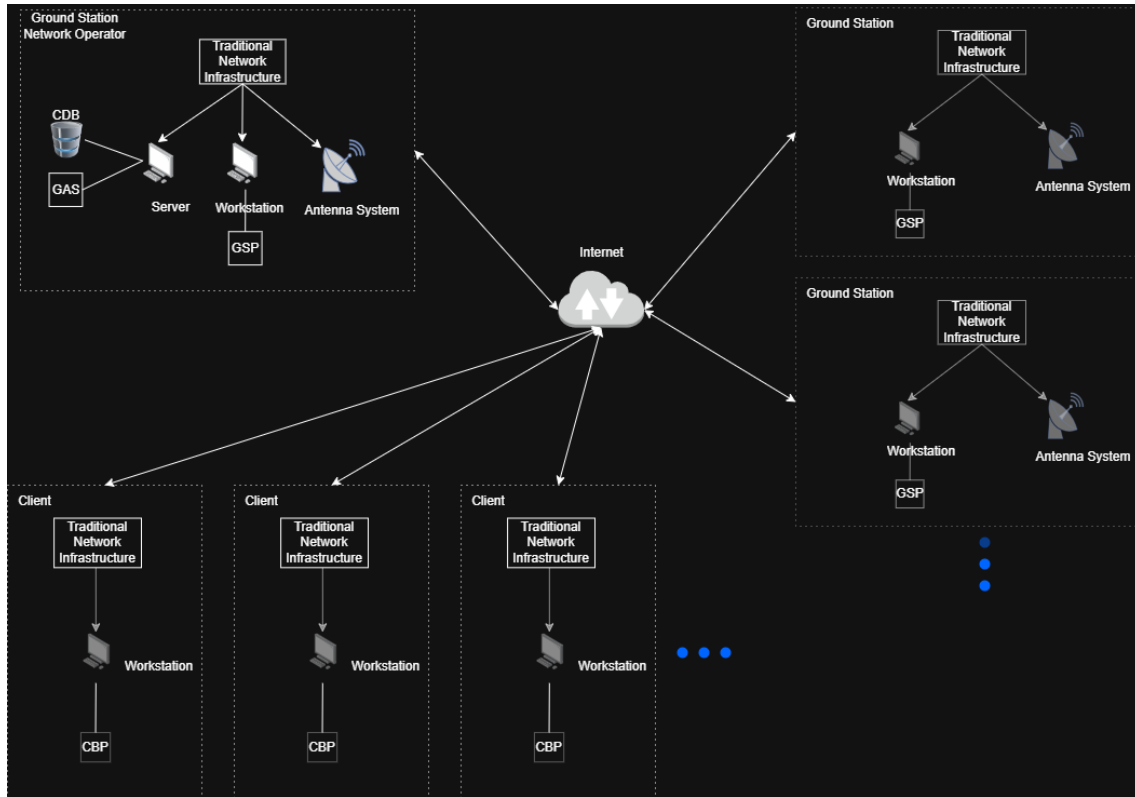


### Appendix B (Ground Station-Client Pairing)

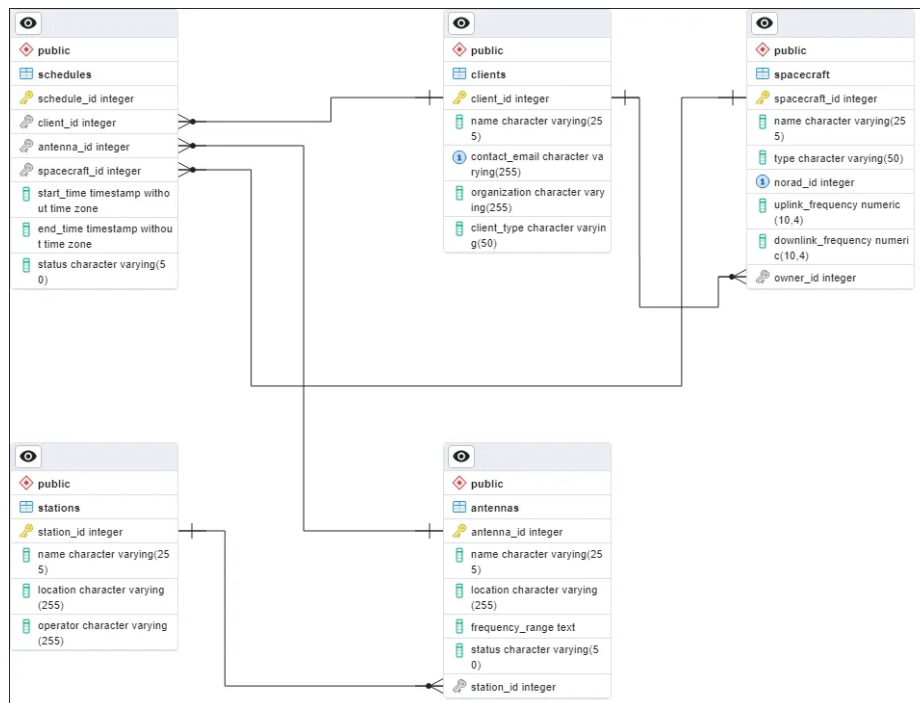


*“Copyright 2025 by the Canadian Space Agency (CSA) on behalf of SpaceOps. All rights reserved. One or more authors of this work are employees of the government of South Africa which may preclude the work from being subject to copyright in South Africa, in which event no copyright is asserted in that country.”*

### Appendix C (Expansion of GSaaS Network Structure)



### Appendix D (Entity Relationship Diagram for a Basic GSaaS Model in pgAdmin)



## Appendix E (Schedule Management Dashboard GUI Prototype)

**Ground Station Schedule Dashboard**

Schedules Antennas Clients Spacecraft Stations

Date Range: From: 31/03/2025 To: 07/04/2025 Status: All Client: All Clients Apply Filters

**Upcoming Satellite Communication Schedule**

ID	Client	Spacecraft	Antenna	Start Time	End Time	Duration	Status
1 4	Planet Labs	SuperDove-21	UHF-1	2025-03-31 17:15	2025-03-31 19:15	2h 0m	confirmed
2 6	Planet Labs	SuperDove-21	UHF-1	2025-04-05 19:15	2025-04-05 20:00	0h 45m	confirmed
3 9	NASA JPL	Sentinel-2B	DSS-43	2025-04-02 08:30	2025-04-02 09:15	0h 45m	confirmed
4 10	ESA	Ariane 5 ECA	X-Band-2	2025-04-04 08:45	2025-04-04 09:45	1h 0m	
5 11	ESA	Sentinel-2B	DSS-14	2025-04-03 04:30	2025-04-03 06:00	1h 30m	
6 13	MIT Haystack ...	Ariane 5 ECA	DSS-14	2025-04-06 23:15	2025-04-07 01:15	2h 0m	confirmed
7 14	John Smith	Falcon 9 B1051	Ka-Band-1	2025-04-06 04:15	2025-04-06 04:45	0h 30m	confirmed
8 15	Planet Labs	SuperDove-21	DSS-63	2025-04-07 21:30	2025-04-07 22:00	0h 30m	
9 16	SpaceX	Falcon 9 B1051	DSS-14	2025-04-06 09:30	2025-04-06 11:00	1h 30m	
10 17	Planet Labs	SuperDove-21	SG-03	2025-04-03 21:15	2025-04-03 22:15	1h 0m	confirmed
11 18	ESA	Sentinel-2B	DSS-63	2025-04-02 10:30	2025-04-02 12:30	2h 0m	
12 19	SpaceX	Starlink-1234	DSS-14	2025-04-06 08:45	2025-04-06 10:45	2h 0m	
13 21	SpaceX	Starlink-1234	SG-03	2025-04-05 18:00	2025-04-05 19:30	1h 30m	
14 23	MIT Haystack ...	Sentinel-2B	SG-03	2025-03-31 00:00	2025-03-31 00:45	0h 45m	confirmed

View Details

## References

- [1] J. Gal-Edd, J. Steck, C. C. Fatig, and R. Zepp, “Low cost ground systems - Fantasy or reality,” *Proceedings - IEEE Aerospace Conference*, Jun. 2003.
- [2] M. Sankarati, M. Merri, M. Spada, V. Navarro, J. Marak, and O. P. Mamert, “Mission Operations as a Service: Cloud Computing for Space Missions beyond Infrastructure-as-a-Service,” *2018 SpaceOps Conference*, May 2014.
- [3] P. Safarik and S. Schuenemann, “Ground segment as a service,” *2018 SpaceOps Conference*, May 2016
- [4] Azt. I. Solutions, “Key Differences Between Cloud Computing vs. Traditional,” *aztech*, Oct. 06, 2023. <https://www.aztechit.co.uk/blog/cloud-computing-vs-traditional/>, (accessed 03.03.25)
- [5] Y. Liu, Y. Chen, Y. Jiao, H. Ma, and T. Wu, “A shared satellite ground station using User-Oriented virtualization technology,” *IEEE Access*, vol. 8, pp. 63923–63934, Jan. 2020.
- [6] “Ground Station as a service - AWS Ground Station - AWS,” *Amazon Web Services, Inc.* <https://aws.amazon.com/ground-station/>, (accessed 06.03.25)
- [7] Blessing, Moses & Olusegun, John. *The Impact of Software-Defined Networking (SDN) on Traditional Network Architectures: Opportunities and Challenges.* (2024).
- [8] “SQLAlchemy ORM tutorial for Python Developers,” *Auth0 - Blog.* <https://auth0.com/blog/sqlalchemy-orm-tutorial-for-python-developers/>, (accessed 08.03.25)
- [9] “JSON.” <https://www.json.org/json-en.html>, (accessed 08.03.25)