Revista Colombiana de Computación
Vol. 25, No. 2. July – Dec 2024, pp. 1-11
e-ISSN: 2539-2115, https://doi.org/10.29375/25392115.5272
Selected paper previously presented at the Latin America High Performance Computing Conference (CARLA 2023), an event held in Cartagena de Indias, Colombia, September 18-22, 2023.

REVISTA COLOMBIANA DE COMPUTACIÓN

# Space Mission as a Service (SMaaS): General-purpose Computing on Space

## Misión Espacial como Servicio (*SMaaS*): Computación de propósito general en el espacio

Leonardo Camargo-Forero[1,2,3,4], Julián G. Rodríguez-Ferreira[2,3,4,5], and Carlos J. Barrios Hernández[2,3,4,6]

[1] UbiHPC, www.ubihpc.com, Bucaramanga, Colombia
[2] Universidad Industrial de Santander, www.uis.edu.co, Bucaramanga, Colombia
[3] Supercomputación y Cálculo Científico UIS (SC3UIS), www.sc3.uis.edu.co
[4] Grupo de Investigación en Cómputo Avanzado y de Gran Escala (CAGE)
[5] Grupo de Investigación en Control, Electrónica, Simulación y Modelamiento (CEMOS), http://e3t.uis.edu.co/eisi/grupo/cemos/, Bucaramanga, Colombia
[6] Grupo de Investigación en Comunicaciones y Tecnologías de la Información (COMIT), https://comit.virtual.uniandes.edu.co/, Bogotá, Colombia

leonardo@ubihpc.com, cbarrios@uis.edu.co, jgrodrif@uis.edu.co

(Received: 6 February 2024; accepted: 18 June 2024, Published online: 31 December 2024)

**Abstract.** Given the critical nature of their missions, space systems such as satellites, probes, spacecraft, etc., are commonly embedded with specific hardware and software solutions. While this approach has led to numerous achievements, it has also limited the available capacities of a spacecraft and the potential integration between multiple units. In this work, we propose the concept of *Space Mission as a Service* (*SMaaS*), a set of strategies for deploying space systems capable of general-purpose computing, embedded Artificial Intelligence, user transparency, and flexibility towards the integration between multiple spacecraft. Such strategies will include evaluating standard operating systems and embedded companion computers, such as the NVIDIA® Jetson Series, under space conditions, common AI frameworks, High-Performance Embedded Computing, and cloud computing as an integrator between space computing devices and earth ground stations. As a demonstration, we intend to evaluate such strategies within the scope of a project consisting of devising the computing on-board system of a nanosatellite belonging to the Colombian Air Force. The possibilities are endless if a spacecraft were embedded with a companion computer with the necessary hardware and software components to execute general-purpose computing and artificial intelligence software. On-board data preprocessing, optimized space-earth download bandwidth, vision-based navigation, autonomous collision avoidance, and overall higher levels of autonomy are a few examples of the potential of this approach that could lead to the implementation of a supercomputer in space.

**Keywords:** High-Performance Computing, Embedded Computing, Space Artificial Intelligence, Supercomputer in Space, Framework.

**Resumen.** Dada la naturaleza crítica de sus misiones, los sistemas espaciales como satélites, sondas, naves espaciales, etc., suelen llevar incorporadas soluciones específicas de hardware y software. Aunque este planteamiento ha dado lugar a numerosos logros, también ha limitado las capacidades disponibles de una nave espacial y la integración potencial entre múltiples unidades. En este trabajo, proponemos el concepto de *Misión Espacial como Servicio* (*SMaaS*, en su acrónimo en inglés), un conjunto de estrategias para desplegar sistemas espaciales capaces de computación de propósito general, Inteligencia Artificial embebida, transparencia para el usuario y flexibilidad hacia la integración entre múltiples naves espaciales. Dichas estrategias incluirán la evaluación de sistemas operativos estándar y computadores complementarios integrados, como la serie NVIDIA® Jetson, en condiciones espaciales, marcos comunes de IA, computación embebida de alto rendimiento y computación en nube como integrador entre los dispositivos de computación espacial y las estaciones terrestres. Como demostración, se pretende evaluar tales estrategias en el marco de un proyecto consistente en idear el sistema informático a bordo de un nanosatélite perteneciente a la Fuerza Aérea Colombiana. Las posibilidades son ilimitadas si a una nave espacial se le incorpora un computador de acompañamiento con los componentes de hardware y software necesarios para ejecutar software de computación de propósito general e inteligencia artificial. El preprocesamiento de datos a bordo, el ancho de banda de descarga espacio-tierra optimizado, la navegación basada en visión, la evasión autónoma de colisiones y, en general, superiores niveles de autonomía son algunos ejemplos del potencial de este enfoque que podría conducir a la implementación de un supercomputador en el espacio.

**Palabras claves:** Computación de alto rendimiento, Computación embebida, Inteligencia artificial espacial, Supercomputador en el espacio, Framework.

# 1 Introduction

Remote sensing, telecommunications, planet exploration, the search for extraterrestrial life, astrobiology, the hunt for exoplanets, military applications, earth navigation, gravitational waves, anti- and dark matter, and the understanding of the origins of the universe are a few examples of space missions, those carried out for the benefit of humanity and towards the conquest of one of the major frontiers in scientific exploration: space.

Within a myriad of differences, space missions have shared something in common since the beginning of the space race. The utilization of specific-purpose hardware and software. This approach has led us to worldwide communications, landing a rover on Mars, human footprints on the moon, and much more. However, space exploration is bound to evolve, given the appearance of general-purpose embedded computing devices that are optimized for energy consumption and weight and have strong computing capacities. Previous attempts to exploit general-purpose companion computers in space, such as the AstroPI (Raspberry Pi Foundation, 2023), a Raspberry PI (2024) optimized for space conditions (e.g. high radiation levels) or the NASA Astrobee robot (Flückiger & Coltin, 2019) have already proven helpful in the International Space Station, e.g. by aiding astronauts in their daily activities. However, if a more powerful computer, e.g. the NVIDIA Jetson Nano (NVIDIA, 2024b) were to be qualified for space missions, in-parallel execution of multiple Artificial Neural Networks (ANNs) could provide a spacecraft with higher levels of autonomy and data-preprocessing capacities that would result in the optimization of the limited bandwidth between space and earth ground stations.

Similar attempts have been carried out with earthly devices such as robots, Unmanned Aerial Vehicles (UAVs, a.k.a. drones), Unmanned Ground Vehicles (UGVs, a.k.a. rovers), etc., where the usage of different types of companion computers capable of High-Performance Embedded Computing (HPEC), contributed to the beginning of the Artificial Intelligence of Things – AioT's era. For example, vision-based landing (Cocchioni et al., 2016), real-time detection and tracking (Jeon et al., 2016), and even supercomputers made of robots Camargo-Forero et al. (2018; 2019). A good revision can be found in Camargo Forero (2019).

However, to our knowledge, a companion computer, such as those belonging to the NVIDIA Jetson Series (NVIDIA, 2024a), has not been implemented in a spacecraft. As a first step towards the implementation of the concept of *Space Mission as a Service* (SMaaS), we intend to assess the feasibility of embedding such computer or similar architectures under space conditions, e.g., high radiation level and to evaluate its requirements in terms of energy consumption, weight and else, given the limited capacities in a spacecraft such as a nanosatellite.

Following, we will implement a co-design approach between a group of universities in Colombia and under the scope of a Colombian Air Force (2024) / Minciencias (2024) project to repurpose an existing nanosatellite for an earth-observation mission. The methodology to be followed is portrayed in Figure 1.

As such, this project aims to propose the SMaaS concept, based on HPEC architectures, to support critical missions and applications carried out with artificial space satellites. Moreover, it is composed of the following specific objectives:

- Analyze and identify nanosatellites' computing workloads, determining levels of architecture and service, data processing, and energy constraints, mainly related to computational efficiency (processing/energy consumption), to propose performance evaluation guidelines.
- Propose a computing architecture to support HPEC and Artificial Intelligence in space missions.

- Design the concept of SMaaS and select or develop a software prototype to evaluate HPEC and Artificial Intelligence capabilities in space missions under the idea of SMaaS.

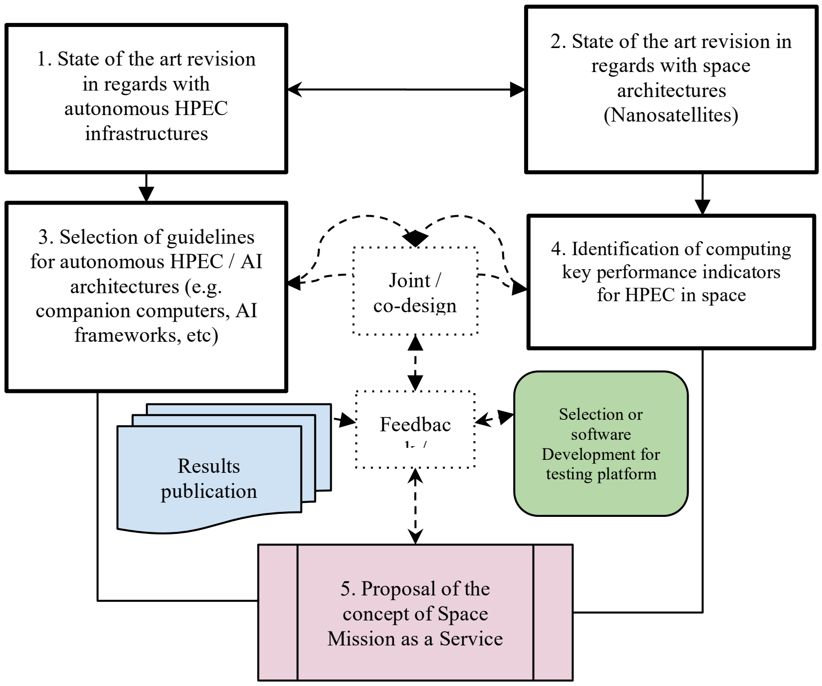


Figure 1. Proposed methodology.
The methodology proposes 5 main stages in a continuous pipeline to define service components addressed to co-design and reproducibility. The main goal of this methodology is to add elements to define Space Mission as a Service from a concept.

With the results of this project, we intend to devise flexible and scalable strategies to provide a spacecraft with sufficient computing power to transform it into a general-purpose autonomous computing infrastructure and to deliver a testbed, usable by the scientific community, to the evaluation of space missions under the scope of AIoT (Seng et al., 2022). Overall, we expect to contribute to the enhancement of the participation of the Latin American community in the space era. This article presents the concept of SMaaS: space mission as a service as a general-purpose computing implementation and deployment in space. It shows the elements that build the idea, subsequently considering aspects related to the implementation and deployment of the model, as well as a first proof of concept, to generate a discussion.

## 2 Space Mission as a Service: the concept

A supercomputer set in space with open access to HPC (High-Performance Computing) resources offers multiple advantages: first, the democratization of space, where any user on earth can selectively request the utilization of space resources plus the processing of the products (data) generated by said utilization; second, the reutilization of existing space resources; next, the collaboration and support between space operators, space agencies, and other actors; and finally, a new generation of IoT (Internet of things) intelligent spacecraft because of embedding general-purpose computing devices. Observing this, we introduce the Space Mission as a Service (SMaaS) concept. However, it's necessary to link the core ideas related to using HPC capabilities given by supercomputing. Indeed, supercomputing seeks to provide the infrastructure, techniques, and technologies required to solve complex problems and massive data.

As known, in a typical organization in an HPC cluster, a supercomputer consists of a set of computers (nodes) connected via a high-speed and low-latency network. Each node is installed and configured with a group of software layers whose objective is to combine the individual computing power in each node and to facilitate a multi-user/multi-purpose approach. The supercomputer user can request a subset of the computing resources (nodes, CPU cores, GPUs, etc.) during a time window to run a specific use, known as a job in the context of HPC. This point is where SMaaS focuses (as portrayed in Figure 2) following a model inspired by cloud computing but addressed for HPC uses, very well known as HPC as a Service (AbdelBaky et al., 2012) or HPC in the Cloud (Sharma et al., 2023), High-Performance Computing as a Service (HPCaaS) refers to using parallel data processing to improve computing performance and perform complex calculations using HPC. It involves the facilitation of supercomputing technology and resources to organizations requiring HPC. Then, HPCaaS provides access to existing hardware and software necessary for high-performance computing and is available on a pay-per-use model via the cloud.

Another aspect considered is the possible technological limitations in space, where it is still non-viable to combine the computing power present in multiple companion computers, those to be embedded in space vehicles (Jeon et al., 2016).

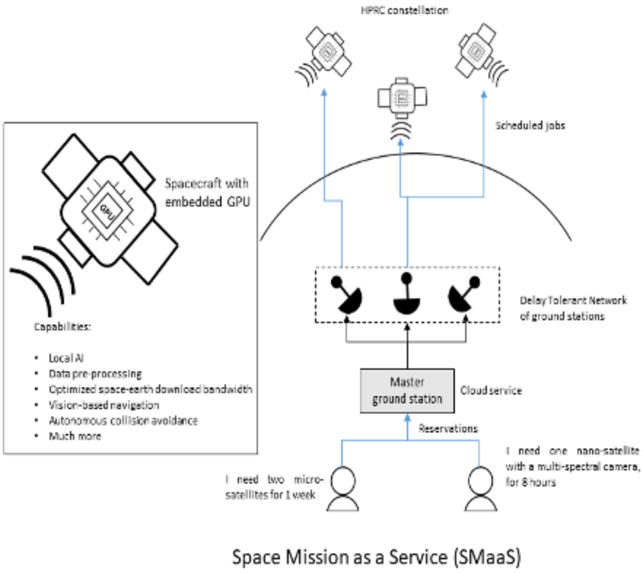


Figure 2. Space Mission as a Service Concept.

Using the methodology shown in Figure 1, the key elements of the service involving tasks, actors, and data flows can be identified. The identification of an actor must depend on the assigned tasks.

Then, the proposed SMaaS is to perform a new approach that allows users on earth to request space vehicles (entities) to perform a Space Mission Workflow (SMW). SMW is a set of tasks to be carried out both in space and on earth, where each task focuses on obtaining or managing data and for which a group of requirements/resources are necessary. The SMaaS concepts are detailed in Table 1.

Table 1. SMaaS Concepts

| Concept | Description |
| --- | --- |
| **Entity** | An entity corresponds to a space vehicle operating in the environment it was developed for, a device/ground installation used for space exploration, or the like. Examples include satellites, space probes, and lunar or planetary robots. |
| **Node** | A node is a companion computer embedded in an entity. It is used for onboard-onboard processing and can interact with services on the ground. An entity can include zero nodes, classified as a dummy, even if it includes an OBC designed exclusively for handling the spacecraft's subsystems but without the local data processing capacity. On the other hand, an entity can embed any number of nodes, even configured as an HPC cluster. |

| Concept | Description |
|---|---|
| **Resources** | A resource refers to those features that contribute/facilitate the execution of software, for example, the number of nodes, CPUs, RAM, GPUs, software licenses, associated data, etc. Consequently, a resource can constitute those mentioned above concerning HPC and those related to the execution of a space mission. For example, entities, cameras, other types of sensors, devices for data collection, e.g., MinION. MinION a device for sequencing DNA or RNA in real-time, which is being used in the ISS within a project to genetically analyze microorganisms in space and with the collaboration of NASA with Hewlett Packard Enterprise for Spaceborne (HPE, 2021), and NASA with IBM for The New Space Age: IBM Develops A Unique, Custom Edge Computing Solution in Space (Altaf, 2021). With this frame of mind, data itself is considered a resource. In addition, an essential resource is a location, that is, the geographical position of a spacecraft at a specific time. To clarify and as an example, users may request to use an entity while it is passing through a particular country to analyze its biodiversity or else. Furthermore, resources in the context of SMaaS also refer to computational resources to be used once data is downloaded to earth. Finally, the resources available in the space vehicles become requirements for a job (HPC), task (HPRC, ubiquitous supercomputing) and must be automatically accessible via earth-space connectivity. |
| **Task** | A task (idem HPC job) consists of software executed over a set of resources during a specific time. SMaaS offers sufficient flexibility, so even with dummy entities, a task may solely request to retrieve data from a subset of them to be processed on earth. Moreover, tasks, as part of a Space Mission Workflow, are submitted to a queue, as described in the following. |
| **Queue** | The concept of a queue is defined as facilitating access to resources. A queue is an abstract element that defines a set of resources grouped according to specific characteristics. i.e., orbital queues (e.g., space entities in a specific orbit), nanosatellite queues or multi-spectral camera queues. Queues are intended to both group resources and to facilitate user prioritization schemas. |
| **Users** | People with the need/desire for space-collected data or experimentation in space-related environments. Users are to be organized within organizations (for example, companies or research institutions). Through this approach, it is possible to offer service levels or hierarchy; all managed via queues. |
| **Service level** | Due to the nature of space vehicles and the environment in which missions are carried out, it is not always possible to guarantee earth-space connectivity, its quality, or successful contacts, in addition to the possibility of mechanical or electronic failure collisions. For these reasons, SMaaS defines service/quality levels that must be associated with specific queues. That is, each queue would have a particular service level. |
| **Space Mission Workflow** | For space systems, the concept of mission refers to the purpose for which the corresponding vehicle is used. However, in SMaaS, the concept of Space Mission Workflow (SMW) is introduced based on the ideas behind General-Purpose Computing Missions (GPCM), from ubiquitous supercomputing (Camargo Forero, 2019), which defines a set of interconnected tasks for which more information will be given in the following. However, the concepts of events, actions, loops, and parallel tasks (those to be executed in an HPC cluster embedded in an entity or to be performed with a Multiple Instruction Multiple Data – MIMD, some computer organizations and their effectiveness approach that is via messaging between multiple entities) (Camargo Forero, 2019) will be omitted from the SMW concept in this version since those would require a continuous entities' communication system, which is currently not feasible to be implemented given space conditions. |
| **Reservation** | A reservation occurs once an SMW is removed from a queue upon successfully allocating the requested resources. Extremely importantly, just as in HPC, HPRC, and ubiquitous supercomputing, resources are allocated to selected users and will not be shared with other users during a reservation. |

This work introduces the Space Mission as a Service specification version 1.0, yet the concept is foreseen with scalability in mind. That is, when continuous communication among space entities is possible, SMaaS will adapt accordingly. Moreover, as with any information system, SMaaS is defined via components and an interaction model described in the following subsection.

## 2.1 Operation model and main components

The SMaaS operation model is described through layers, organized into two main components: backend and frontend, as shown in Figure 3. Correspondingly, the backend component is composed of layers.

- *Interaction layer*: This layer is intended for interaction with users, the terrestrial communications layer, and the management of services for data storage, processing, and publication.
- *Terrestrial communications layer*: Network of terrestrial antennas to guarantee periodic communication with space vehicles.
- *Space layer*: Space vehicles.

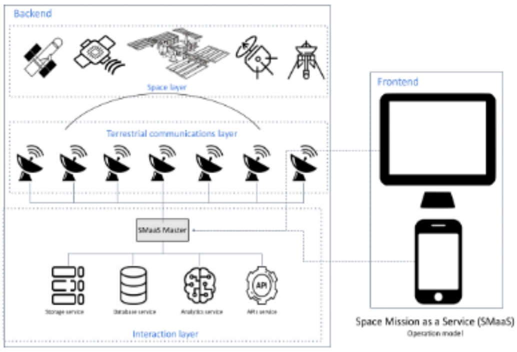


Figure 3. SMaaS layers and main components.
Following the general visibility model of a distributed computing system providing a service, it is possible to identify the components associated with each one of the layers. In the figure is possible to identify the space layer, terrestrial communication layer, and the integration layer.

The frontend component is briefly described as an interface for users requesting access to space resources. Following, we tell each backend layer.

## 2.2 Space Mission Workflow

The workflow of a space mission involves a series of intricate steps and requirements to ensure its success. As introduced in Table 1 and shown in Figure 4, a task consists of software executed over a set of resources during a specific time. Moreover, a task can be classified as:

- SPACE: A task to be executed in a space vehicle. In a space vehicle, task execution involves carrying out various operations and activities necessary for the mission's success. Here are some key aspects of task execution in a space vehicle:
  *Autonomous and Remote Control*: Uncrewed spacecraft can have varying levels of autonomy, ranging from remote control to complete autonomy with pre-programmed operations. In this case, the spacecraft may receive control signals from Earth, while others operate autonomously based on pre-defined instructions.
  *Mission Payload and Bus*: A space vehicle consists of the mission payload, which includes scientific instruments or equipment, and the bus or platform. The bus provides physical structure, thermal control, electrical power, attitude control, and telemetry.
  *Telemetry and Data Transmission*: Robotic spacecraft use telemetry to transmit acquired data and vehicle status information back to Earth. This data is crucial for monitoring the spacecraft's performance and gathering scientific information.
  *Attitude and Articulation Control*: Maintaining the spacecraft's attitude (orientation) is essential for various tasks, such as accurate communication with Earth, precise data collection, thermal control, and guidance for maneuvers.
  *Maneuvering and Propulsion*: Spacecraft often carry small rocket engines for maneuvering and orienting in space.
  *Navigation and Guidance*: Spacecraft use navigation systems to determine their position, velocity, and attitude. It's important to note that the specific tasks executed in a space vehicle can vary depending on the mission objectives, the type of spacecraft, and the stage of the mission. The execution of tasks is carefully planned and monitored to ensure the mission's success.
- DATA: A task to be executed with a data processing service, e.g., AWS S3, DynamoDB, or any informatic or cloud provider. This task involves actions such as:
  *Data Collection*: Collecting relevant data from various sources, such as databases, files, APIs, or sensors. This data can be structured, semi-structured, or unstructured.
  *Data Cleaning or Data Cleansing*: This involves identifying and correcting errors, inconsistencies, missing values, outliers, and duplicates in the collected data. This step ensures the data is accurate and reliable for further processing.

*Data Transformation*: Involves converting the collected data into a more suitable format for analysis. This can include tasks such as data normalization, aggregation, encoding, and feature engineering. The goal is to prepare the data for the specific data mining or analysis task.
*Data Integration*: Involves combining data from multiple sources into a unified dataset. This task ensures that data from different sources can be analyzed together and provides a comprehensive view of the information.
*Data Reduction*: Data reduction techniques are used to reduce the size of the dataset while preserving its important characteristics. This can involve sampling, dimensionality reduction, or feature selection. Data reduction helps to improve the efficiency of subsequent data processing and analysis tasks.
*Data Analysis*: After processing and preparing the data, various analysis techniques can be applied to gain insights and extract meaningful information. This can include statistical analysis, machine learning algorithms, data visualization, and more.
*Data Output*: The goal is to generate the desired output. This can involve creating reports addressing visualizations, dashboards, or any other form of data-valuable presentation for decision-making or further analysis.

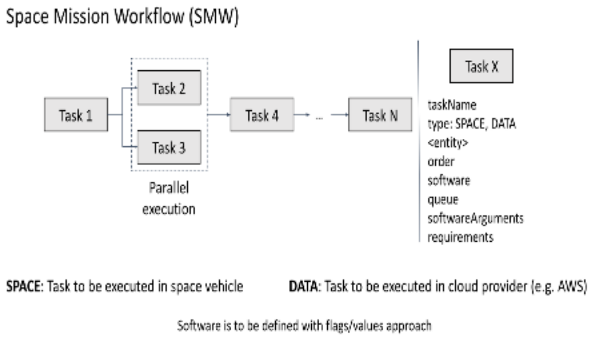


Figure 4. Space Mission Workflow.
To define the Space Mission Workflow is critical to understand that the tasks to be executed in the space vehicle are inside the SPACE value, and the DATA corresponds to the tasks to be executed in a cloud provided. This approach is possible due to the definition of the software with flags and values.

Consequently, a Space Mission Workflow (SMW) is a set of interconnected tasks following a particular order, i.e., the tasks pipeline. In addition, multiple tasks can be executed in the same order, which is defined as parallel execution, as observed in Figure 4.

To facilitate programmatic interaction between users and SMaaS services and inter-component interaction, we have devised the SMaaS language, which is based on the UbiSL language (Ubiquitous Supercomputing Language) proposed in Camargo Forero (2019) and the YAML language Ain't (YAML Language Development Team, 2021).

The SMaaS language is a work in progress, but so far, we have developed the SMW YAML and other templates, which can be found in the SMaaS simulator GitLab repository, the GitLab repository where the SMaaS HITL/SITL simulator can be seen as well. The SMW YAML template defines the mission's name, description, contact information, and tasks. For tasks of type SPACE, the template allows for the specification of the desired entities for the mission and its associated software, including its arguments. Very importantly, an output data folder would connect with the next task, e.g., if that task is of type DATA, where an AWS data service is to be specified. Details can be found in the comments in the template; other templates will be discussed next. Then, we would address the HITL/SITL simulator developed to demonstrate the ideas behind SMaaS.

## 2.3 SMaaS HITL/SITL simulator

We introduce a HITL and SITL simulator. As previously discussed, the simulator is classified as HITL because for its execution, companion computers, specifically the NVIDIA Jetson Nano (NVIDIA, 2019), have been used, as it would be with real space entities, using space-qualified versions, but still maintaining the same operating system (Ubuntu) and necessary software libraries.

Moreover, the simulator behaves as SITL because the SC service is executed in the companion computers, plus the SM service is performed in a machine easily deployable to the cloud. Figure 5 shows the simulator architecture, and a set of considerations are described to note:

- The front-end and SM services are deployed on the same machine.
- Each entity is to be configured as an independent companion computer.
- Apache Kafka service has been used to facilitate a microservices distributed infrastructure. Therefore, its master service is deployed in the same machine where the SM service is deployed.
- AWS credentials must be configured in the machine where the S service is deployed to facilitate programmatic access to AWS services.
- The simulator was written using Python 3, and the frontend software was developed using Streamlit (Snowflake Inc, 2024), a Python framework for creating data apps as web applications.
- The simulator was developed using the following libraries: PyAstronomy (PyA group, 2013), Kafka-python (Powers et al., 2016), Flask (Pallets, 2019), and Mapbox (2011).

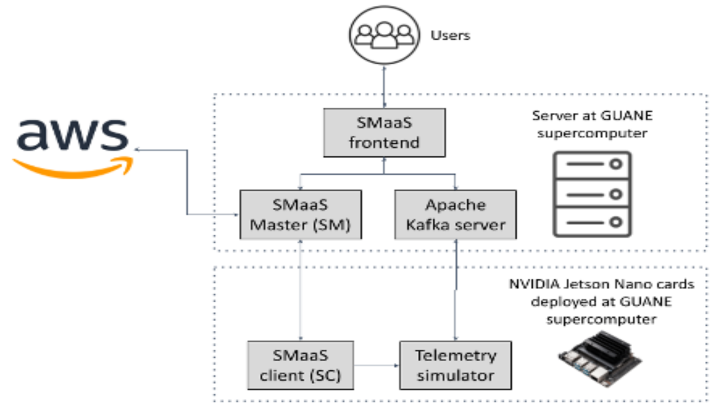


Figure 5. Simulator general architecture.
The simulator allows different tests of the concept, with some specific configurations, independently of the cloud provider but implemented in-situ infrastructure. This figure shows the simulator using AWS, the GUANE supercomputer, and the NVIDIA Jetson Nano.

It is possible to review the SC3 GitLab repository (Universidad Industrial de Santander, 2024) demanding authorization to test the main components: master.py (SM service), the manager unit for all SMaaS master service features; client.py (SC service), the component to manage all SMaaS client service features; core.py, to provide the SMaaSEntity class, which can be used to represent any SMaaS vehicle, to simulate telemetry, the interaction with the company computer operating system, and use orbital parameters to define the entity's orbit; the orbit.yaml template to set such parameters; and finally, the telemetrySimulator.py, using the SMaaSEntity class to simulate the vehicle's telemetry, that is to compute, with a selected time step, position (latitude, longitude, altitude, x, y, z), velocity, airspeed, and data related to the companion computer, such as CPU, GPU, RAM consumption, energy and more. Telemetry is published to Kafka topics with the name convention of <entity_name>-telemetry.

## 3 Using SMaaS: Discussion and Further Work

Observing any space vehicle, understanding this spaceship as a device traditionally built with specific-purpose hardware and software is rethought as a general-purpose IoT device, embedding a companion computer. One of the main challenges is how to reuse the device, obtaining the best performance at a low cost involving sustainable use. For example, in 2018, the Colombian Air Force research center: "Centro de Investigación en Tecnologías Aeroespaciales (CITE)" and three Colombian universities, Universidad Industrial de Santander, Universidad del Valle y Universidad Sergio Arboleda, agreed upon a joint space engineering project, named MISC-3 based on a 3-unit CubeSat mission, as cooperation between research institutions in Colombia and funded by the national research ministry: MINCIENCIAS. The MISC-3 project, recently named LEOpar mission, will develop the first Colombian satellite built by multiple universities, a CubeSat 3U model with a hyperspectral camera in VIS-NIR spectral bands, as a payload.

This satellite was proposed for multiple uses and required the use of general-purpose computing for both budgetary reasons and a diversity of applications. Then, one of the main components, the Processing Formatting Control Unit (PFCU), is the central computing unit that will use a XILINX VIRTEX 5-QB and a NVIDIA JETSON TX2 to perform on-board-on-board-processing, both general-purpose computers. Using SMaaS concepts, data analysis will be carried out on board the space entity, facilitating the downloading of only valuable data.
Taking the elements of the Table 1, we show the Table 2. For this case:

Table 2. SMaaS concepts for LEOpar Mission.
LEOpar's mission supports the proof of concept of the SMaaS approach. Each identified element provides concepts addressing the definition of the service and the description of the different components.

| Concept | Description |
|---|---|
| **Entity** | MISC-3 CubeSat |
| **Node** | NVIDIA JETSON TX2 |
| **Resources** | GPU: 256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores. CPU: Quad-Core ARM® Cortex®-A57 MPCore. MEMORY: 8GB 128-bit LPDDR4 Memory 1866 MHx - 59.7 GB/s. STORAGE: 32GB eMMC 5.1 |
| **Task** | Camera Control and Camera Applications with an Energy Consumption Under 7.5 W. (Total of consumption 15 W) |
| **Queue** | A set of multi-spectral camera tasks, scheduled by a general application based in application rules. |
| **Users** | Only MISC-3 researchers and operators. |
| **Service level** | Associated with the multi-spectral camera tasks and energy consumption rules. |
| **Space Mission Workflow** | The workflow of a space mission involves a series of intricate steps and requirements to ensure its success. |
| **Reservation** | Resources allocation is managed by SC3UIS HPC resources allocation based in Slurm. |

On the other hand, following SMaaS concepts, using cloud-based services such as AWS ground station to increase the number of contacts, coupled with access to data-processing services, would make it possible to create an integrated platform for space and earth applications, following the exposed in Figure 5. In this case, the definition of the objectives and requirements of a space mission is crucial for its success. This involves planning action, deployment of the task, and job execution based on requirements and infrastructure support. SMAAS concepts allow us to understand a specific space mission's complex and chaotic nature; agreeing to and comparing the planning process to the realistic effects written in after-action reviews can significantly increase the chances of developing a successful space mission that meets SLAs and requirements.

SMaaS can provide sustainability and a contribution to space vehicle reutilization, including when the vehicle can't simply be used, given its location. Therefore, users on Earth can access its capabilities in space during selected time windows. From elsewhere, SMaaS aims to set the basis for future space research and business infrastructure, powered by edge-cloud general-purpose computing and open to anybody on Earth.

Further work is planned to optimize and improve the simulator to include more capacities and the possibility of customizing the modules following the SMaaS concept to grow and democratize space exploitation. The emergence of space mission services reflects a growing trend towards democratizing access to space and making space missions more affordable and accessible for a broader range of organizations and industries:

- Innovations in Mission Control Software and AI (Mission Control, 2024).
- Commercial Space-as-a-Service Missions.
- Mission Control as a Service, companies like Spaceit (2017) offer Mission Control as a Service (MCAAS), providing a one-stop solution for satellite ground operations, including cloud-based Mission Control System, worldwide radio communication, satellite control, and 24/7 support. This service model removes the need for custom software development, hardware investments, ground station licenses, and personnel for upkeep and satellite management, making access to space more affordable.

The Space-as-a-Service Model has been extended to the space industry, aims to make space missions more accessible and cost-effective (Ellipsis Drive, 2022). Small satellite operators are leveraging specialized technology and miniaturized sensors to gain a significant advantage in monitoring space, facilitating the Satellite-as-a-Service model (Financial News Media, 2022).

## Acknowledgments

The authors acknowledge the support from Universidad Industrial de Santander (UIS) and Vicerrectoría de Investigación y Extensión UIS-VIE. This contribution was part of the contract RC N° 010-1583/ 2020000518 Space Mission as a Service Model Development into the 2020 support program for postdoctoral stays at Universidad Industrial de Santander.

## ORCID ID

Leonardo Camargo-Forero https://orcid.org/0000-0003-3440-9325
Julián Gustavo Rodríguez Ferreira https://orcid.org/0000-0003-1373-6044
Carlos Jaime Barrios Hernández https://orcid.org/0000-0002-3227-8651

## References

Fuerza Aeroespacial Colombiana. (2024). *Fuerza Aeroespacial Colombiana*. Retrieved from www.fac.mil.co: https://www.fac.mil.co/es

AbdelBaky, M., Parashar, M., Kim, H., Jordan, K. E., Sachdeva, V., Sexton, J., . . . Wheeler, M. F. (2012, October). Enabling High-Performance Computing as a Service. *Computer, 45*(10), 72-80. https://doi.org/10.1109/MC.2012.293

Altaf, N. (2021, February 19). *The New Space Age: IBM Develops A Unique, Custom Edge Computing Solution in Space.* Retrieved from IBM Website: https://www.ibm.com/blog/ibm-develops-a-unique-custom-edge-computing-solution-in-space/

Camargo Forero, L. (2019, September). *Ubiquitous supercomputing: design and development of enabling technologies for multi-robot systems rethinking supercomputing.* Universitat Politècnica de Catalunya (UPC), Departament de Física. Universitat Politècnica de Catalunya (UPC). https://doi.org/10.5821/dissertation-2117-171404

Camargo-Forero, L., Royo, P., & Prats, X. (2018, September 26). Towards high performance robotic computing. *Robotics and Autonomous Systems, 107*, 167-181. https://doi.org/10.1016/j.robot.2018.05.011

Camargo-Forero, L., Royo, P., & Prats, X. (2019, April). The ARCHADE: Ubiquitous Supercomputing for robotics. Part I: Philosophy. *Robotics and Autonomous Systems, 114*, 187-198. https://doi.org/10.1016/j.robot.2019.01.006

Cocchioni, F., Frontoni, E., Ippoliti, G., Longhi, S., Mancini, A., & Zingaretti, P. (2016, December). Visual Based Landing for an Unmanned Quadrotor. *Journal of Intelligent & Robotic Systems, 84*, 511–528. https://doi.org/10.1007/s10846-015-0271-6

Ellipsis Drive. (2022, June 22). *Exploring the New Space ecosystem | Space-as-a-Service*. Retrieved from Ellipsis Drive website: https://ellipsis-drive.com/blog/new-space-ecosystem-space-as-a-service/

Financial News Media. (2022, May 13). Why the Space-as-a-Service Business Models are Taking the Space Sector by Storm. *Financial News Media website*. Retrieved from https://www.financialnewsmedia.com/why-the-space-as-a-service-business-models-are-taking-the-space-sector-by-storm/

Flückiger, L., & Coltin, B. (2019, May 10). Astrobee Robot Software: Enabling Mobile Autonomy on the ISS. *NTRS - NASA Technical Reports Server*, 52. Monterey, CA, United States: Naval Academy. Retrieved from https://ntrs.nasa.gov/citations/20190027521

HPE. (2021). *HPE Spaceborne Computer*. Retrieved from Hewlett Packard Enterprise Website: https://www.hpe.com/us/en/compute/hpc/supercomputing/spaceborne.html

Jeon, D., Kim, D.-H., Ha, Y.-G., & Tyan, V. (2016, May). Image processing acceleration for intelligent unmanned aerial vehicle on mobile GPU. *Soft Computing, 20*(5), 1713-1720. https://doi.org/10.1007/s00500-015-1656-y

Mapbox. (2011, November 8). *Mapbox: Location intelligence*. Retrieved from Mapbox website: https://www.mapbox.com/

Ministerio de Ciencia Tecnología e Innovación. (2024). *Minciencias*. Retrieved from minciencias.gov.co: https://minciencias.gov.co/

Mission Control. (2024). *Mission Control: Software for Earth, Moon, and Mars*. Retrieved from Mision Control Space website: https://missioncontrolspace.com/

NVIDIA. (2019, March 18). *Get Started With Jetson Nano Developer Kit*. Retrieved from NVIDIA Developer website: https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit

NVIDIA. (2024a). *NVIDIA*. Obtenido de https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/

NVIDIA. (2024b). *NVIDIA Jetson Nano*. Retrieved from www.nvidia.com: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/

Pallets. (2019, July 2). *Flask*. Retrieved from Flask website: https://flask.palletsprojects.com/en/stable/

Powers, D., Arthur, D., Couture-Beil, A., Saifee, A.-A., Duquesne, C.-M., Dimson, T., . . . Hichri, M. H. (2016, April 28). *Kafka-python*. Retrieved from Kafka-python website: https://kafka-python.readthedocs.io/en/master/

PyA group. (2013, August 16). *Welcome to PyAstronomy*. Retrieved from PyAstronomy website: https://pyastronomy.readthedocs.io/en/latest/index.html#about-pyastronomy

Raspberry Pi Foundation. (2023). *Astro PI*. Retrieved from astro-pi.org: https://astro-pi.org/

Raspberry Pi Foundation. (2024). *Raspberry Pi*. Retrieved from www.raspberrypi.org: https://www.raspberrypi.org/

Seng, K. P., Ang, L. M., & Ngharamike, E. (2022). Artificial intelligence Internet of Things: A new paradigm of distributed sensor networks. (P. Lopez Iturri, Ed.) *International Journal of Distributed Sensor Networks, 18*(3), 1-27. https://doi.org/10.1177/15501477211062835

Sharma, S., Jain, G., D, P., & Bhardwaj, S. (2023, July 11). High Performance Computing (Hpc) in the Cloud: A Proactive Fault Tolerance (Pft) Strategy. *International Journal of Intelligent Systems and Applications in Engineering, 11*(8s), 71-78. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/3023

Snowflake Inc. (2024, December 13). *Streamlit: A faster way to build and share data apps*. Retrieved from Streamlit website: https://streamlit.io/

Spaceit. (2017, January 14). *Spaceit*. Retrieved from https://spaceit.eu/

Universidad Industrial de Santander. (2024). *SC3UIS: SC3 Super Computación y Calculo Cientifico UIS Repository*. Retrieved from SC3UIS Github Repository: https://github.com/SC3UIS

YAML Language Development Team. (2021, October 1). *YAML Ain't Markup Language (YAML™) version 1.2*, Revision 1.2.2. Retrieved from YAML Website: https://yaml.org/spec/1.2.2/#fn:team