AURORA: CLOUD-BASED GROUND AND MISSION CONTROL SOFTWARE AND MARKETPLACE - OPERATIONAL EXPERIENCE

Bruno Carvalho[1], Sergio Mucciarelli[2], Grazia Bibiano[3], Hugo Gomes[4]

[1] D-Orbit, Polo Tecnologico de Lisboa, Rua António Champalimaud, Lte 1, 1600-514 Lisboa, Portugal; +351 212696710; bruno.carvalho@dorbit.space

[2] D-Orbit, Viale Risorgimento, 57, 22073 Fino Mornasco (CO), Italy, +39 0237920900, sergio.mucciarelli@dorbit.space

[3] D-Orbit, Polo Tecnológico de Lisboa, Rua António Champalimaud, Lte 1, 1600-514 Lisboa, Portugal; +351 212696710; grazia.bibiano@dorbit.space

[4] D-Orbit, Polo Tecnológico de Lisboa, Rua António Champalimaud, Lte 1, 1600-514 Lisboa, Portugal; +351 212696710; hugo.gomes@dorbit.space

ABSTRACT

AURORA is a cloud-based distributed software that enables D-Orbit satellite fleet operations. The nature of the selected architecture allows the company to operate our ION spacecraft from anywhere in the world with an internet connection, providing the necessary resilience to the system. Our operators are given the tools, within AURORA, to automate the operational procedures and to quickly add functionality to the control system, needed for new subsystems as D-Orbit provides hosted payload services (used regularly for In-Orbit Demonstrations). Through AURORA it is easy to add connectivity, we currently offer significant global coverage, and keep adding more capacity with the partnerships we are forging. This paper presents an overview of AURORA from an operators perspective.

1 CONTEXT

D-Orbit launched its first satellite mission, D-Sat, in 2017, and after a thorough market analysis the company decided to develop its own ground software system. Several factors contributed to this decision, but two where deciders:

- Costs, the market had great tools and proven suppliers but most solutions were too complex to operate a simple CubeSat, most licencing models, were too expensive for a small start-up company.
- Flexibility, most solutions were monolithic, any required customisations represented additional costs, and we were not in control of the end product if we wanted to scale.

The company then progressed with its roadmap and required a solution to control multiple spacecraft, a fleet of ION Spacecraft Carrier Vehicles (SCV). We quickly realised that the software strategy needed to evolve to match the broad requirements and the envisaged scalability. Factors like infrastructure investment (CAPEX for servers, own antennae, etc.) and maintenance (OPEX for on-going maintenance, upgrades, etc.) were considered, as well as the envisaged number of operators per spacecraft, which we wanted to keep low. We turned to proven web technologies and, after validating some of the core elements (under a small ARTES contract with ESA), to guarantee that we could handle things like latency and other constraints that could apply to satellite communications, we developed what we call AURORA.

2 ARCHITECTURE

The services composing AURORA Core are deployed in Amazon Web Services and run in containers orchestrated using Kubernetes to have minimum downtime, this allows for scaling the services horizontally and facilitate the deployment process.

The services composing AURORA communicate between each other using a publish-subscribe pattern. Apache Kafka is used as an event streaming platform to allow the communication between services.

One of the main concepts in AURORA Core is the adapter, that allows each satellite to be managed independently, giving the flexibility to use any communication protocol, if implemented, and guarantees better security, by not allowing data visibility between users of different satellites if needed.

A high-level component diagram (Figure 1.) shows the main services that allow the communication between the ground segment and the satellite. The adapter services wrap the communication protocol to encode /decode send and receive packets to/from the satellite.

The files sent and received to/from the satellites are stored in Aurora Resource Manager Service (ARMS) a Data Repository and accessible via a Command Line Interface (CLI) or web interface (browser).

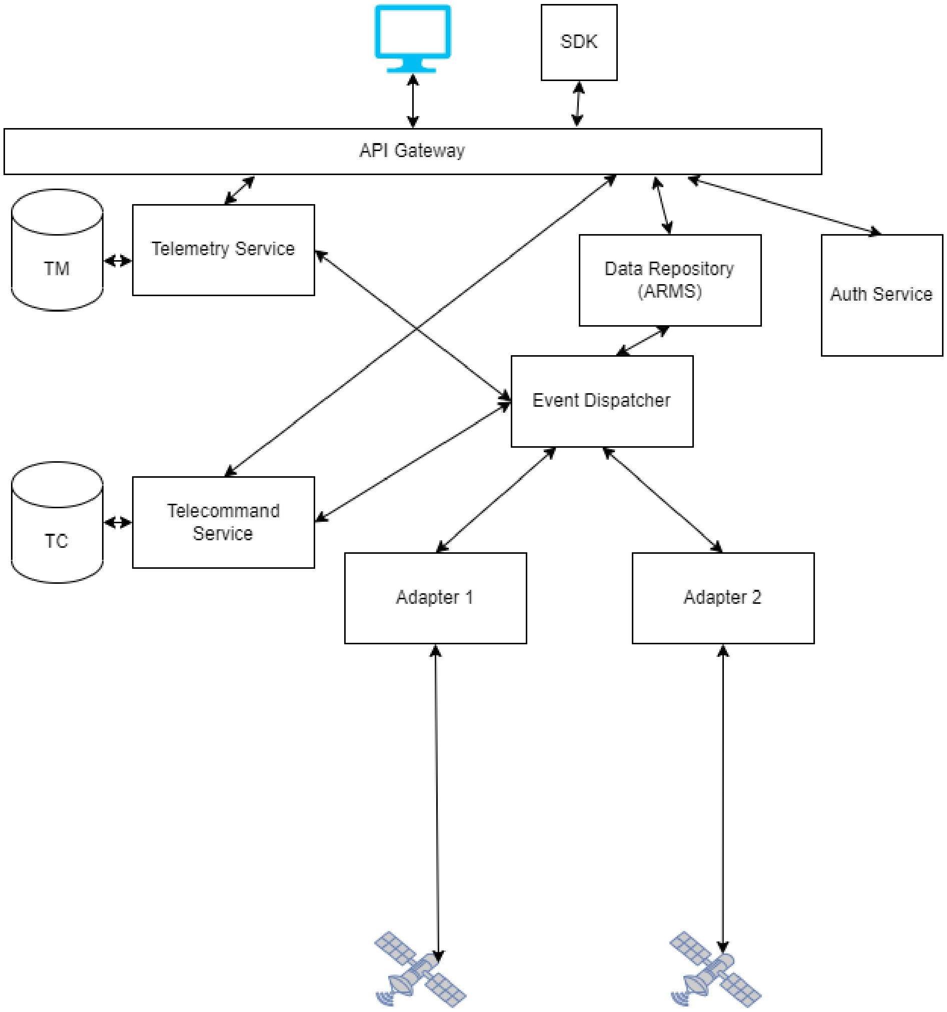


Figure 1. System Architecture

3 USABILITY

There are different options to use AURORA, and D-Orbit operators are using both the user-friendly web interface and the backend API enabled automation. The software can run in any browser and allows satellite operators to perform operations such as send telecommands to the satellite and receive satellite telemetry (TM/TC). Currently the protocol used for TM/TC is CubeSat Space Protocol. AURORA Core allows also to transfer files between ground and the satellites using protocol developed by D-Orbit, the Data Transfer Protocol (DTP).

AURORA provides UI aimed to operate fleets of satellites using a single interface. It was developed to provide the customization of the dashboards according with the user needs. Each dashboard can be fully customizable, in terms of number of components (widgets) and their positions.

The frontend provides already a set of useful widgets, that can be themselves customized according with the user needs. Furthermore, the UI also allows an easy definition of multiple satellite structures defining the subsystems composing the satellite, and the data-sources (satellite telemetry parameters) within each system. After that, the creation of a fleet of satellite instances is as easy as just giving names to them.

The same frontend also provides an interface to easily create automatic procedures that can execute complex tasks. That feature is possible due to the script execution runtime environment existing in AURORA that allows the preparation of lists of scripts written in Python that can be grouped (procedures) and executed directly from AURORA UI. If there is the need to execute the operations from external services, the integration can also be done through the AURORA API directly, or instead use the AURORA SDK, which provides all the features available in AURORA using a library implemented in Python that wraps the AURORA APIs. The exposure of the API and SDK allows for automation of all the AURORA functionalities, including new satellite configurations and the actual operations.

The next sections show the overall UI for the various configuration steps currently existing in AURORA.

3.1 Satellite and Fleet Configuration

One of the first steps to get started using AURORA is to define how a satellite is configured in terms of its sub-systems and their TM parameters (referred in AURORA as data-sources).

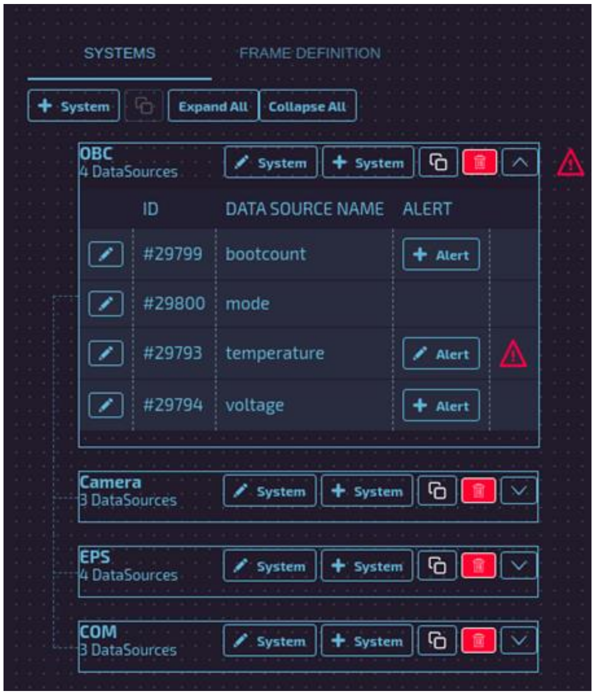
The Figure 2. is an example of a satellite configured with some systems/payloads and their TM parameters.

Within the satellite configuration window, it is also possible to define the alert severities for the different OOL (Out of Limits) levels, as show in Figure 3.

AURORA provides an easy way to configure a fleet of satellites (referred in AURORA as satellite instances) that are instances created based on a pre-configured satellite configuration, it is a simple as giving a name and an ID to a satellite and specify the configuration that will be used (see Figure 4.).

While the web forms provide an easy visual way of editing and configuring the system, all these configurations can also be done resourcing to JASON files or via direct calls to the API.

In the example below (Figure 5.), two satellites were created based on the same configuration, as soon as this is done, AURORA can interface with those satellites.



OBC
4 DataSources

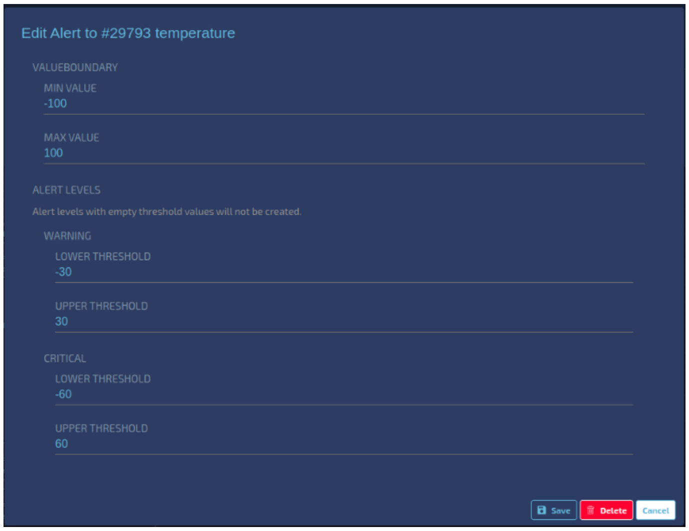
ID	DATA SOURCE NAME	ALERT
#29799	bootcount	Alert
#29800	mode	
#29793	temperature	Alert
#29794	voltage	Alert

Camera
3 DataSources

EPS
4 DataSources

COM
3 DataSources

Figure 2. Satellite Configuration



Edit Alert to #29793 temperature

VALUEBOUNDARY

MIN VALUE
-100

MAX VALUE
100

ALERT LEVELS

Alert levels with empty threshold values will not be created.

WARNING

LOWER THRESHOLD
-30

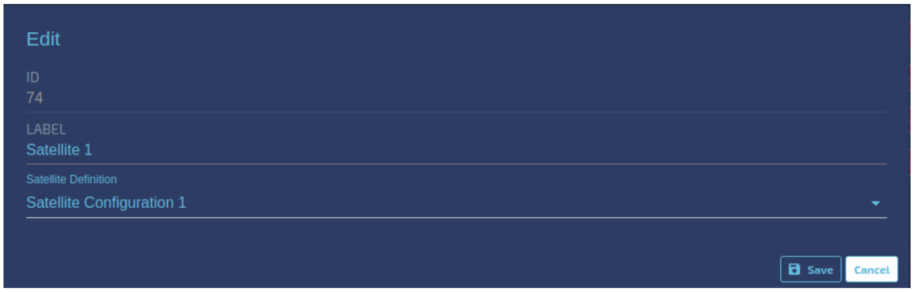
UPPER THRESHOLD
30

CRITICAL

LOWER THRESHOLD
-60

UPPER THRESHOLD
60

Figure 3. OOL Configuration



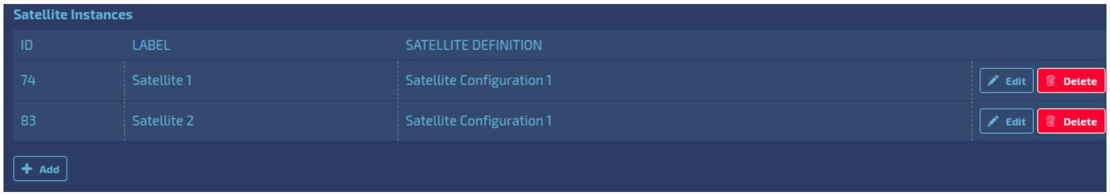
Edit

ID
74

LABEL
Satellite 1

Satellite Definition
Satellite Configuration 1

Figure 4. Fleet Configuration



Satellite Instances

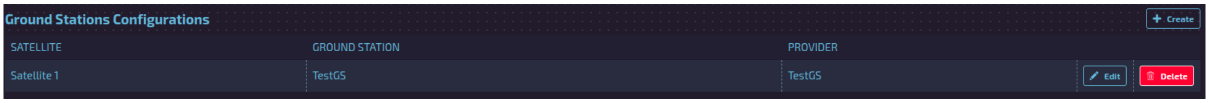
ID	LABEL	SATELLITE DEFINITION	
74	Satellite 1	Satellite Configuration 1	Edit Delete
83	Satellite 2	Satellite Configuration 1	Edit Delete

Figure 5. Satellite Instances

3.2 Ground Stations Configuration

AURORA also provides an easy way to configure how the communication between the Ground Station (GS) and AURORA is established. It provides the interface with some GS providers, already configured as part of our global network of partners, and was implemented to be extensible for other GS. Currently the communication protocol required to interface with the GS is MQTT.

The GS communication configuration requires the mapping between a GS and a satellite. That triggers AURORA to automatically handle the passages and consequently the communication between AURORA and a GS.



Ground Stations Configurations

SATELLITE	GROUND STATION	PROVIDER	
Satellite 1	TestGS	TestGS	

Figure 6. Ground Station Configuration

As per this example (Figure 6.), the TestGS is a special type of GS that was implemented in AURORA to provide 24/7 passages, that keeps the communication with the GS always available, thus providing an easy way to interface with a simulator or an EM of a satellite.

This allows for AURORA to be used across the whole life cycle of a satellite, including the early development, integration, testing and validation (AIT/AIV), enabling scripts and procedure reusability as well as training facilities for the operators.

3.3 Resources Manager (ARMS)

Satellite operations require not only to send telecommands and receive telemetry frames, but also share files with the satellite (and in both directions: uplink and downlink). For that, AURORA provides a resources storage facility, which can be used to upload files (resources) into AURORA. From there the resource be uploaded to the satellite in whichever protocol supported by the satellite. Download of resources from the satellite is done in the reverse way, where afterwards the file received from the satellite can be downloaded from AURORA, or made available to third party applications in the cloud.

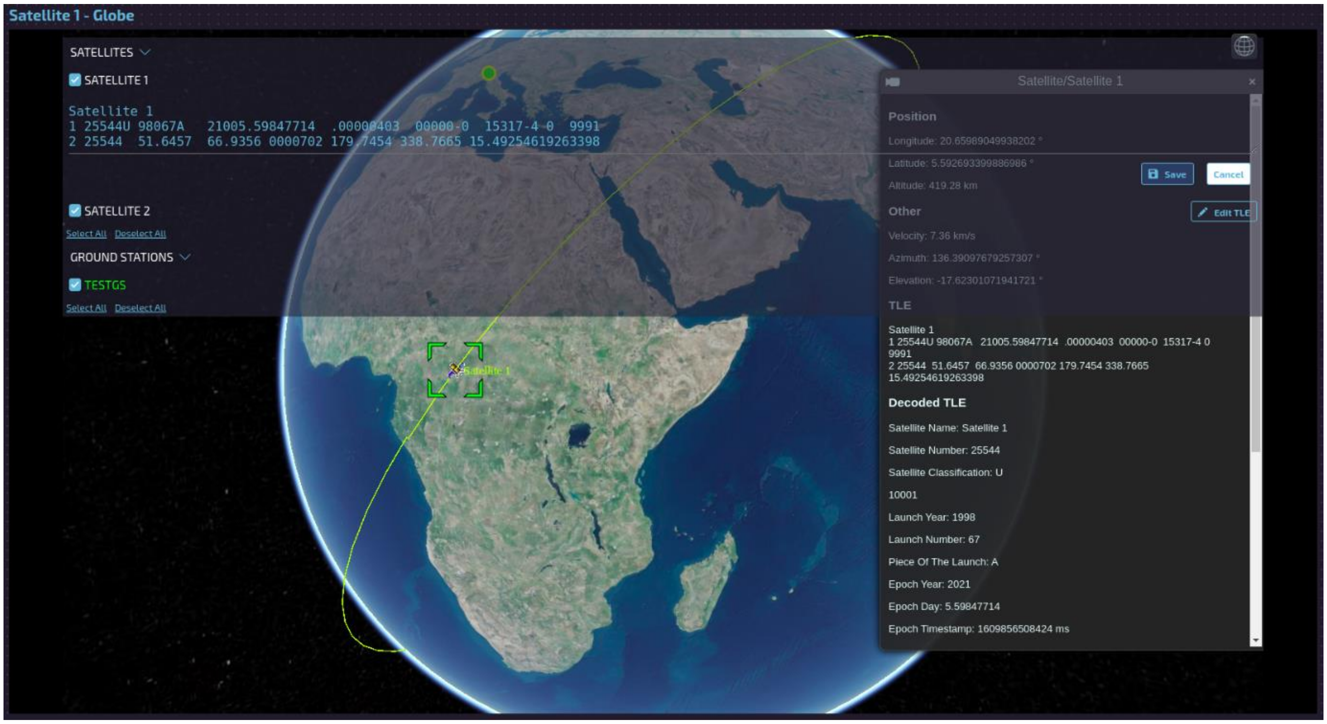
The storage provides various features such as copy, move, delete, and quarantine resources, which enables its usage as a storage service also for other resources not necessarily to be shared with the satellite.

3.4 Dashboards Configuration

AURORA presents a set of features related to the configuration of dashboards. It provides a very flexible way to organize a set of dashboards according to the mission and to the operator's roles, as each can have their own public and private sets of dashboards.

There is a standard set of displays/widgets available that can be added to any user customized dashboard, including, but not limited to: globe/map view; tables; graphs; fleet overview; historical data search and display; command line interface; telecommand (TC) execution display and history; telemetry (TC) frame display; procedure execution; script execution console display; passage planner and visibility window displays.

Most displays can be configured for various time references, and can be optionally configured with time ranges, which make the display to behave as static display (e.g. showing offline TM) or as a dynamic dysplay, showing the live data. Colour coding is done automatically on the data sources highlighting any alerts that may have been configured, with different colours for different criticality levels.



```
Satellite 1
1 25544U 98067A   21005.59847714  .00000403  00000-0  15317-4 0  9991
2 25544  51.6457  66.9356 0000702 179.7454 338.7665 15.49254619263398
```

Position

Longitude: 20.65989049938202 °

Latitude: 5.592693399886986 °

Altitude: 419.28 km

Other

Velocity: 7.36 km/s

Azimuth: 136.39097679257307 °

Elevation: -17.62301071941721 °

TLE

```
Satellite 1
1 25544U 98067A  21005.59847714  .00000403  00000-0  15317-4 0
9991
2 25544  51.6457  66.9356 0000702 179.7454 338.7665
15.49254619263398
```

Decoded TLE

Satellite Name: Satellite 1

Satellite Number: 25544

Satellite Classification: U

10001

Launch Year: 1998

Launch Number: 67

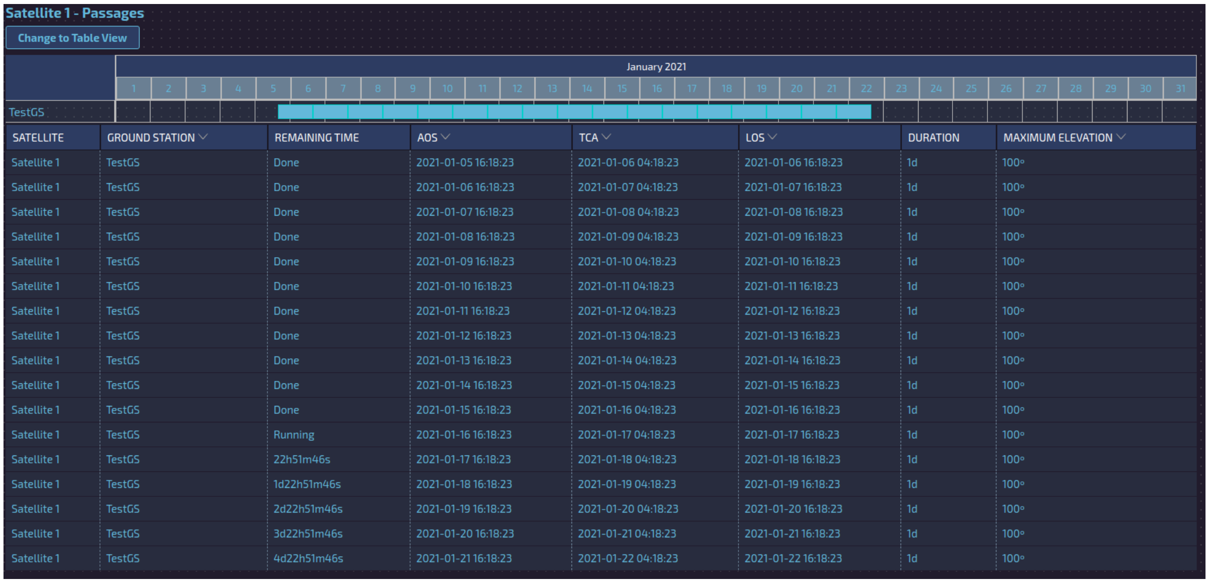
Piece Of The Launch: A

Epoch Year: 2021

Epoch Day: 5.59847714

Epoch Timestamp: 1609856508424 ms

Figure 7. Globe: This display allows the operator to see the list of GS and satellites involved in a certain mission



SATELLITE	GROUND STATION	REMAINING TIME	AOS	TCA	LOS	DURATION	MAXIMUM ELEVATION
Satellite 1	TestGS	Done	2021-01-05 16:18:23	2021-01-06 04:18:23	2021-01-06 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-06 16:18:23	2021-01-07 04:18:23	2021-01-07 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-07 16:18:23	2021-01-08 04:18:23	2021-01-08 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-08 16:18:23	2021-01-09 04:18:23	2021-01-09 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-09 16:18:23	2021-01-10 04:18:23	2021-01-10 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-10 16:18:23	2021-01-11 04:18:23	2021-01-11 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-11 16:18:23	2021-01-12 04:18:23	2021-01-12 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-12 16:18:23	2021-01-13 04:18:23	2021-01-13 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-13 16:18:23	2021-01-14 04:18:23	2021-01-14 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-14 16:18:23	2021-01-15 04:18:23	2021-01-15 16:18:23	1d	100º
Satellite 1	TestGS	Done	2021-01-15 16:18:23	2021-01-16 04:18:23	2021-01-16 16:18:23	1d	100º
Satellite 1	TestGS	Running	2021-01-16 16:18:23	2021-01-17 04:18:23	2021-01-17 16:18:23	1d	100º
Satellite 1	TestGS	22h51m46s	2021-01-17 16:18:23	2021-01-18 04:18:23	2021-01-18 16:18:23	1d	100º
Satellite 1	TestGS	1d22h51m46s	2021-01-18 16:18:23	2021-01-19 04:18:23	2021-01-19 16:18:23	1d	100º
Satellite 1	TestGS	2d22h51m46s	2021-01-19 16:18:23	2021-01-20 04:18:23	2021-01-20 16:18:23	1d	100º
Satellite 1	TestGS	3d22h51m46s	2021-01-20 16:18:23	2021-01-21 04:18:23	2021-01-21 16:18:23	1d	100º
Satellite 1	TestGS	4d22h51m46s	2021-01-21 16:18:23	2021-01-22 04:18:23	2021-01-22 16:18:23	1d	100º

Figure 8. Passages Display

3.5 Operations List

Operations List allows the operators to track the status and the progress of the files uploaded to or downloaded from the satellite. AURORA provides an API for it, and it is being used by this display to monitor the transfer of the files. The files which download is completed are stored in ARMS. Those to be uploaded to the satellite need to be already in ARMS.

4 EXPANDING THE TOOL SET

Because D-Orbit provides a wide range of services to its customers, including hosted payloads, and satellite as a service (or satellite for rent), AURORA needs to accommodate an ever expanding set of requirements. Adding new features to AURORA then became a concern but, at the same time, an opportunity to broaden the pool of partners by creating a marketplace concept within the AURORA ecosystem. We now have components developed by third parties, and several ground station providers, that are available as add-ons to the baseline subscription

4.1 Scripts and Procedures

The most common and easy way to expand AURORA is to leverage its API and SDK, and its scripting capabilities. AURORA provides a script execution environment, which allows the creation of scripts (in Python) using the SDK as the wrapper of AURORA APIs. Those scripts can be then grouped into procedures, to execute them in sequence. This expansion can be used to operate the satellite, configure ground stations and automate the system and to provide customised features.

While scripts and procedures can still be triggered and monitored by the user interface, the SDK is aimed at automating tasks and to allow third party software to interact in an automatic way with AURORA. Extra functionalities are available through the SDK such as update satellite and telemetries database, import and export dashboard in order to reuse them, update the configuration of the ground station dynamically.

Authentication for the SDK will be performed using an SDK token.

4.2 Grafana and other widgets

Grafana is an open-source monitoring and analytics solution, and it can be used to monitor the state of the AURORA services and to display information about telemetries. The Grafana plots can be integrated in AURORA UI as widgets (an example is shown in Figure 9.).

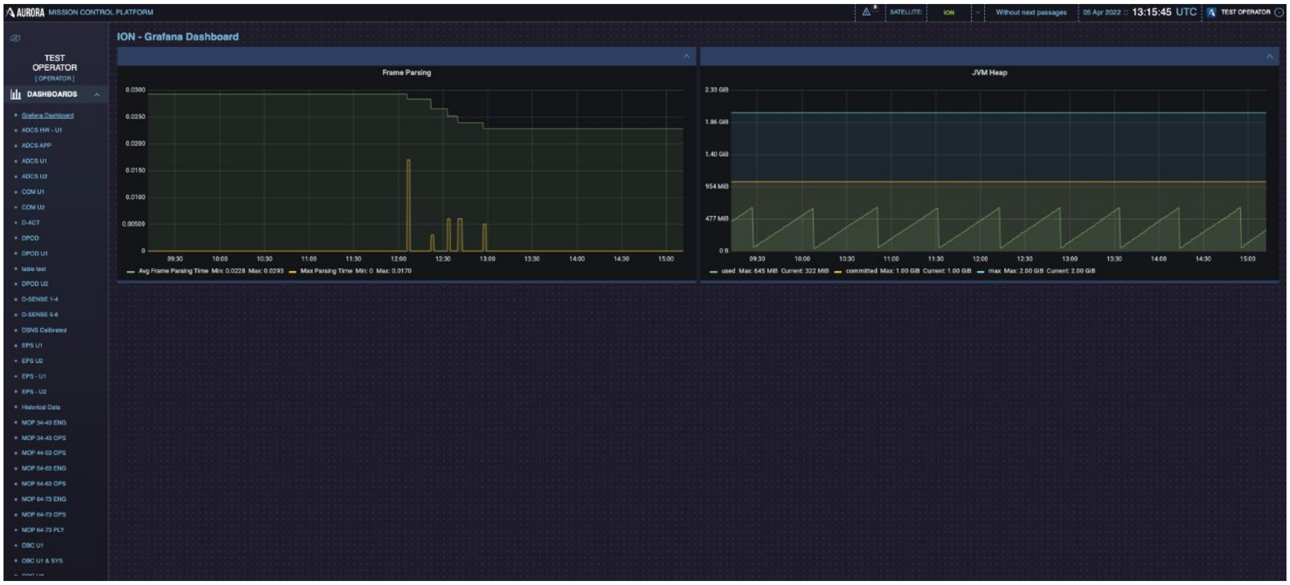


Figure 9. Grafana display embedded in AURORA

Other widgets can be developed and provided to the frontend, this allows for custom payload displays and other specific visual components to be embedded in the system and made available to the operator.

5 GLOBAL COVERAGE

Currently AURORA supports the usage of Ground Stations from various providers, which allows D-Orbit to operate its fleet of ION Spacecraft Carrier Vehicles permanently and from anywhere in the World. However, as the space to ground connectivity is still one of the bottle necks for seamless communications we support an easy the integration of other GS providers. It requires the provider follow a set of guidelines, but the system allows for the tasking of the antennae based on several factors, and for any provider to monetise their downtime by making their capacity available through AURORA to D-Orbit and to our customers.

The capability to integrate multiple GS into a single platform with a single interface, enables the customer to choose which GSs to use for each mission. This results in a more effective mission operations, as the customer is capable of increase the number of passages required, and even decide to use specific GS for strategic and commercial reasons.

Despite the interfaces with the GS being different for each GS provider, and not standardized, AURORA can support all of them, keeping the same interface for the satellite operators. This is a sort of Air B&B of antennae and provides the marketplace and the ecosystem with a transparent and competitive solution to ground communications.

Given its flexibility to interface with different GS providers, even when using different protocols and data formats, AURORA is also capable to integrate any other GS.

AURORA provides two different use cases for the integration of the GS, where one of them has some limitations that will be explained in the following sections.

In the first option, the communication between AURORA and the GS provider is done through a message broker (MQTT), which is used for data uplink and downlink. The management of passages is done through a REST API, normally provided by the GS provider.

In the second option, the communication between AURORA and GS provider is done through the message broker (MQTT), for data uplink and downlink, but the passages are predicted and generated by AURORA. This options allows for a more seamless integration into the mission operations and more flexibility and control by the operator.

6 SECURITY CONCERNS

Security is one of the major concerns these days, particularly for companies that need to protect data and critical assets. Cybersecurity therefore is a very important topic for AURORA developers and users.

Using the most advanced cybersecurity features available from the cloud infrastructure provider, as well as the best practices in the industry.

6.1 Authentication and Authorization

AURORA Core supports fine grained authorization in order to allow only certain users to perform operations such as to send specific telecommands or see telemetries for a specific satellite. Users can manage a group of satellites and each user can have different permissions depending on their role.
One classical example commonly used is the organization of teams involved in operations of satellites as following:
- Fleet Lead: Users that have control over the configuration of the mission control, as well as planning the mission activities. Those users are also responsible to assign the permissions levels to the other users according to their roles in the mission.
- Operators: Users responsible to control the satellite, executing tasks such as sending telecommands and monitor the heath of the satellites. Those users are only allowed to send commands and visualize certain telemetry parameters related to the satellite health.
- Data Analysts: Users that cannot execute any operations. Instead, they are just able to consume telemetry data and perform analyses to it.
- Payload Data Consumers: Usually are clients that have interest in data of their payloads, and thus cannot perform any spacecraft related operations nor monitor its data by default.

As seen above, the permission rules can be configured according to the organization needs and as per mission basis.

6.2 Public, private and hybrid cloud deployments

Although currently deployed in AWS, AURORA can be deployed in other cloud environments, including secure private clouds, as well as hybrid setups. D-Orbit is working with specialised partners that provide specific cybersecurity features that are required by several of our customers, this ability is relevant for customers, mainly institutional, that may have constraints on data policies and data protection as well as other cybersecurity constraints.

7 HELPDESK

The software is developed and maintain by a dedicated team, not by the spacecraft operators. AURORA is still evolving and as all software, particularly mission critical software, requires regular support. This support is more critical during initial deployments, for new spacecraft configurations or for new operators. The provision of the Software as a Service comes with defined service level agreements (SLA) and helpdesk is available not just for bug reporting, but also for operator support.
One of the advantages of the cloud deployment is the possibility of having regular updates that may correct issues, patch security issues, or simply add more features to the system.

8 CONCLUSIONS AND FUTURE WORK

The concept of Software as a Service (SaaS) is not new but, with AURORA, D-Orbit applies the concept to space operations – **Space Operations as a Service (SOaaS)**. The nature of the system allows for subscription models to be applied for commercialization, so it can be used by third parties. The same architecture allows for ground to satellite connectivity to be added easily, from different third party suppliers, providing effective global coverage, automating procedures during the visibility windows optimising usage costs and minimising operator workload for multiple satellite operations.

AURORA minimises CAPEX required by new satellite operators and optimises the OPEX of the actual operations.

D-Orbit is operating five ION spacecrafts, as of April 1st 2022, all with different configurations and missions (i.e. different instruments, ranging from imagery EO payloads to cybersecurity) and AURORA is proving to be the backbone of the services we are providing.

AURORA is evolving, with new features such as support for CCSDS protocols coming online soon, and via its API, people are adding functionality, partners and users with specific needs and constraints are developing components for their specific usage, and some of these components are available, via the marketplace, to third-party operators.

Finally, because AURORA is cloud-based, it enables cloud-services to be used in real time when data is downloaded from the spacecraft (e.g. data processing and correlation with other data sets), speeding up the delivery to the end-user. Moreover, it also enables the same services on the spacecraft itself, D-Orbit is now also offering its NUVOLA space cloud service, enabled by AURORA.

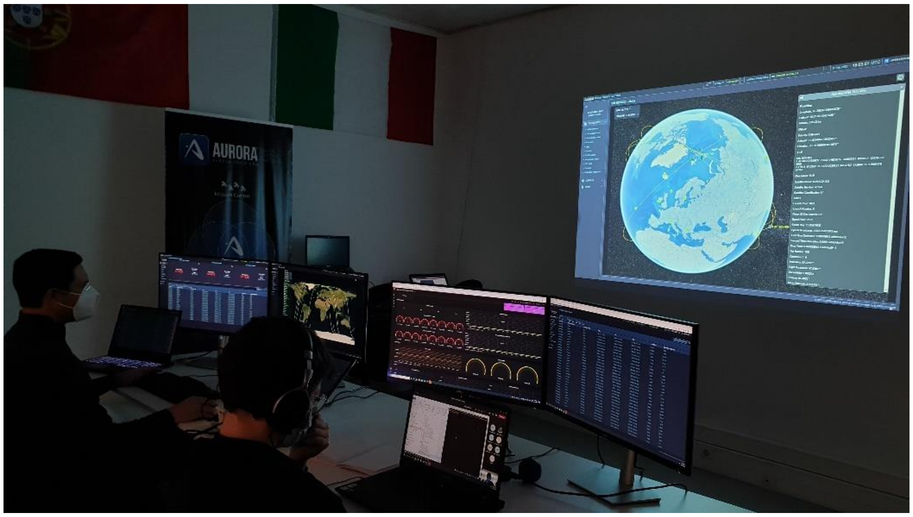


Figure 10. AURORA in D-Orbit’s control room in Lisbon, Portugal

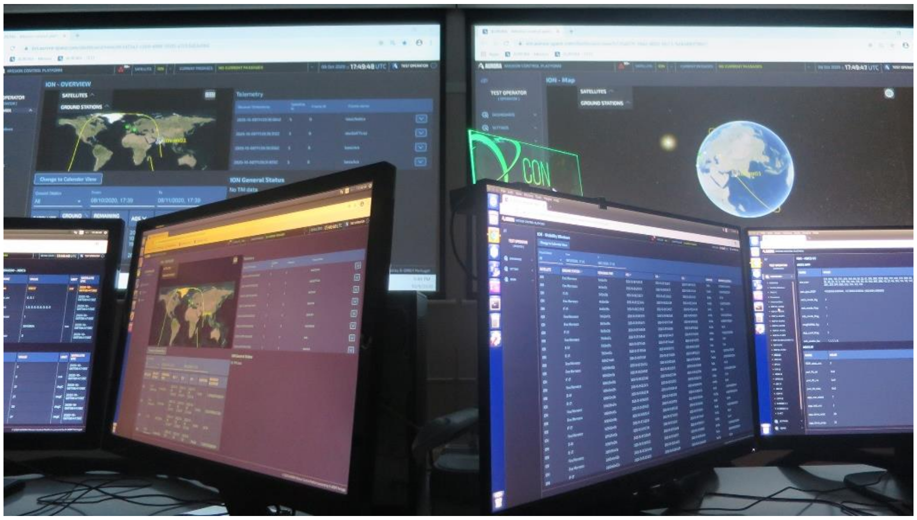


Figure 11. AURORA in D-Orbit’s control room in Fino Mornasco, Italy