17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 228

Galileo GCS: Monitoring Services with Middleware Applications

Guillermo Ramos[*], Mónica Jiménez[†], Víctor Pozo[‡]
GMV, Madrid, 28760, Spain

Pedro Rodríguez de Andrés[§]
European Space Agency (ESA) / European Space Technology Centre (ESTEC), Noordwijk, 2201, The Netherlands

Abstract

Satellite ground segment subsystems development has been traditionally linked to the use of consolidated products and technologies, usually developed ad-hoc for these specific purposes and then maintained for many years to avoid costly revalidation campaigns. This approach provides a safe and reliable basis for operations, but imposes some limits and delays on the subsystem evolutions and prevents the use of state-of-art technologies that could radically improve their efficiency. Initial steps to address these issues in the Galileo mission were completed in 2021, with the operational deployment of a constellation performance and telemetry analysis service based on open-source software components which is currently supporting the Galileo GCS Operations.

Following a similar approach, upgrading the ground assets monitoring and control approach was the next step; it was indeed a challenging task, as the solution to be deployed must be not only independent and resilient, but also flexible enough to accommodate a wide variety of ground assets to be managed (some of them probably outdated) and should include powerful analysis mechanisms to make the most of the metrics gathered. While an ad-hoc approach could perfectly fit these requirements, the time and resources required to develop such solution may have escalated rapidly.

On the other hand, nowadays most of the hardware and software infrastructure supporting a satellite ground segment does not radically differ from generic IT datacenters, which usually include monitoring systems with similar constraints; these solutions are commonly based on available open-source COTS to allow fast evolutions and deployments, focusing the efforts on the final customization rather than in the development process, and with a greater flexibility to deal with future obsolescence resolution issues.

Inspired by this context, the new Galileo ground control segment monitoring system continues with the transition from monolithic subsystems to a collection of more generic services, carefully chosen, integrated and configured to provide specialized functions; in this case the services deployed cover the data gathering, storage, normalization, exploitation and display functions for the monitoring of ground assets, following a data-centric approach to ease future expansions, and without requiring costly adaptations of the current infrastructure. In addition, the introduction of common APIs and standard data structures allows as well including constellation status data in this monitoring framework, to provide a complete situational awareness of the status of the space and ground assets.

As result of this upgrade, monitoring information is now aggregated and displayed in a more synthesized approach through flexible dashboards, with a direct benefit on how routine operations and KPIs are managed: alerts and monitoring metrics are more clear, automated processes can be supervised more efficiently, and problem detection and reaction times are improved.

Keywords: Ground Segment, Galileo, COTS, Technology, Telemetry, Performance, KPI

---

[*] Galileo Ground Control Segment Engineering Manager, GMV, Madrid, 28760, Spain; gramos@gmv.com
[†] Galileo Ground Control Segment Services Manager, GMV, Madrid, 28760, Spain; mjanton@gmv.com
[‡] Director of Navigation Systems Ground Control Segment Business Unit, GMV, Madrid, 28760, Spain; vpozo@gmv.com
[§] Head of Galileo G1 Ground Control Segment Management Service, ESA/ESTEC, Noordwijk, 2201, The Netherlands; pedro.rodriguez.de.andres@esa.int

Acronyms/Abbreviations

AI = Artificial Intelligence
API = Application Programming Interface
COTS = Commercial-off-the-shelf
ESA = European Space Agency
FOC = Full Operational Capability (Galileo mission phase)
GCC = Ground Control Centre
GCS = Ground Control Segment
HMI = Human Machine Interface
IT = Information Technology
IOV = In Orbit Validation (Galileo mission phase)
KPI = Key Performance Indicator
SOA = Service-oriented architecture
SNMP = Simple Network Management Protocol

1. Introduction

The Galileo Ground Control Segment (GCS) is responsible for the control and monitoring of the Galileo satellites, planning and automation functions; its infrastructure comprises two control centres (GCCs) and several Telemetry Tracking and Control facilities disseminated around the globe. The ground assets associated to this infrastructure include a heterogeneous collection of hardware and software items, which are managed by the operators from the GCCs.

The monitoring and control infrastructure associated to these ground assets has been gradually developed and deployed in parallel to the GCS growth, based on an ad-hoc system which has been expanded as required during the last decade. This approach has allowed a successful setup and operations of the Galileo constellation since 2011, however, technology has evolved rapidly during the last years and once the GCS infrastructure renewal activities started a few years ago it was the right time to evolve the paradigm of the ground assets monitoring and control system as well. The solution to be deployed had to be not only independent and resilient, but also flexible enough to accommodate the different manufacturers and functions included in the ground assets to be managed while on the other hand it is necessary to include, as well, powerful analysis mechanisms to make the most of the metrics gathered and improve the operator's experience.

Nowadays, a direct replacement of the current ad-hoc system would require many years of design, development and validation activities to comply with the strict requirements and standards needed for the Galileo Project. On the other hand, the tight renewal schedule associated to the cybersecurity requirements and the obsolescence resolution strategy for the ground assets demanded a quicker approach. Inspired by the latest solutions designed for modern IT datacenters, the roadmap stablished was based on available open-source COTS to allow fast evolutions and deployments, focusing the efforts on the final customization and securitization rather than in the development process, and with a greater flexibility to deal with future obsolescence resolution issues.

This paper provides an overview of the new ground assets monitoring and control approach based on middleware applications; first, the logical architecture of the GCS monitoring and control approach is presented, identifying the functions that need to be covered by the new middleware layer, and the convenience of combining ground and space assets monitoring information to improve data analysis. The migration strategy from the current monitoring system is described in sections 3 and 4, which provide the rationale behind the objective of evolving towards a service-oriented architecture with an event-centric approach. Finally, an overview of the final GCS situational awareness function is provided, followed by some conclusions regarding this whole process and lessons learned.

2. GCS monitoring and control approach

The high-level monitoring and control function encompasses all the operational activities required for the operator of the Galileo system to visualize the status of ground and flight segments, the commanding of their assets, and the evaluation of the performance and availability of the system. In the GCS context, the architecture depicted in Fig.1 identifies the required functions and defines the desired logical relationship between the different functional blocks:

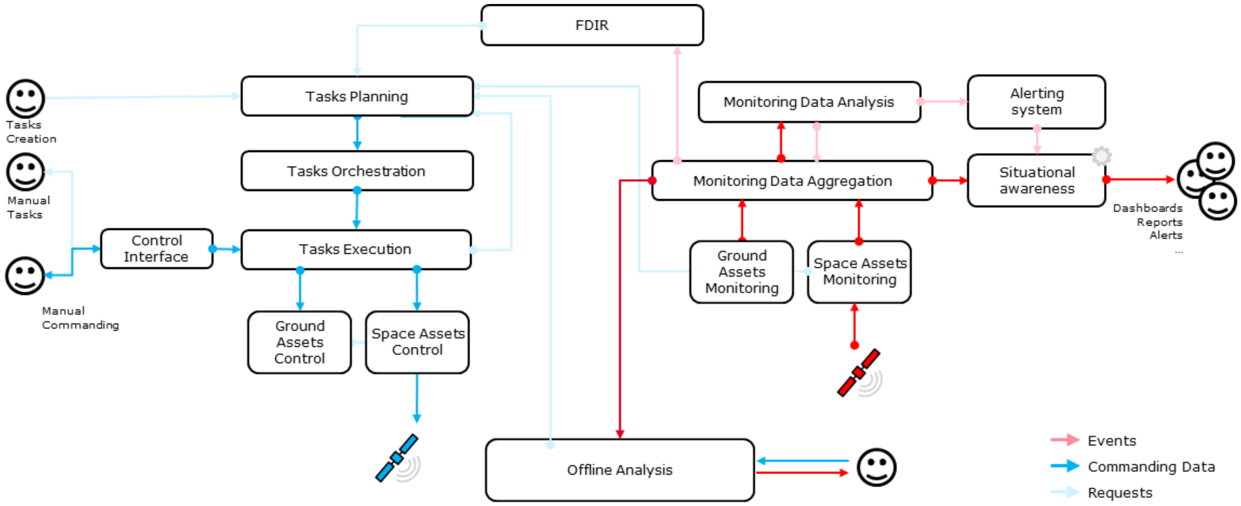


Fig 1 – Monitoring and Control logical architecture

Monitoring data is generated by the ground and space assets and aggregated by a single function, which implements data normalization processes to ease the execution of correlation analysis by the monitoring data analysis function; this component provides event information to be added to the aggregated data, as well as relevant events to be handled by the alerting system. Operators have access to the metrics and alerts through dedicated dashboards and reports generated by the situational awareness function, that can be customized according to their role and required KPIs, as well as through offline analysis tools which allow exploring the data in depth. On the control side, monitoring data is used to generate control tasks either in automatic or manual processes in order to command both ground and space assets.

The following table lists the detailed purpose of each service identified:

Table 1. Monitoring and control functions decomposition

Function	Description
Space assets data gathering	Functionality in charge of gathering the different Monitoring data from the space assets - i.e., metrics (e.g. satellite telemetry, satellite orbit predictions) and low-level events (e.g. on-board events, …). The different assets to be monitored provide monitoring data either in a passive (e.g., satellite telemetry retrieval, …) or active way (e.g. determination of satellite orbits based on ranging measurements, …).
Ground assets data gathering	Functionality associated to the gathering of Monitoring data from the ground assets (all levels) - i.e., metrics and low-level events (e.g. GCS network components and status, servers, workstations and virtual machines metrics, etc.). The different assets to be monitored provide monitoring data either in a passive (logs) or active way (e.g. agents).
Monitoring data aggregation	A common storage layer, which allows querying, summarizing, exploring, and analysing the data by following a systematic and unified approach. The storage layer is designed following a data-centric approach. The description can be found in section 4.
Monitoring data analysis	Function that analyses the collected Monitoring Data and look for a correlation of metrics and events from different sources that could be associated with catalogued issues and triggering the corresponding Integrated Metrics and Events (derived parameters, alarms, info, …). This function could also include behaviour patterns identification to support the Operator finding incidents not tracked by specific alarms.

Published by the Mohammed Bin Rashid Space Centre (MBRSC)
on behalf of SpaceOps, with permission and released to the MBRSC to publish in all forms.

Situational awareness	Provision of monitoring information by displaying visual elements linked to aggregated data that give the Operator a perception of environmental Metrics and Events with respect to time or space, the comprehension of their meaning, and the projection of their future status. The display includes a set of dashboards and capabilities which allow displaying constellation or infrastructure-related metrics. Additionally, it interfaces with the Ground Assets Control interface to provide a centralized endpoint for the GCS M&C operations
Offline analysis	Function which provides the Operator the capability of running all the available system functions under a controlled environment for analysis of a given solution, not interfering with the online workflow
FDIR	Function is in charge of implementing the Fault Detection Isolation and Recovery (FDIR) for a detected issue by checking if the integrated monitoring data is reporting something known and requesting isolation and recovery tasks to be run for the known issues.
Tasks planning	Function which provides the Operator with the means to defining the Tasks to be run within the GCS.
Tasks orchestration	Function in charge of triggering the Task Execution function for the execution of a given task at a certain point in time (or based on a given event). Tasks could be defined as Graph or Code Tasks and can be composed of other Graph and Code Task.
Tasks execution	Function in charge of running the different kind of Tasks that could be triggered by the Task Orchestrator.
Ground assets control	Functionality associated to each Ground Asset offering a commanding interface for the reception of commands to be executed.
Space assets control	Function covering the functionality offering the capabilities to send Command Messages to the space segment and process the related Acknowledge Messages.

From an operator's perspective, the monitoring data aggregation function is the key driver that shall allow the provision of all the metrics in a centralized and normalized way, enabling them to query, summarize, explore and analyse the data by following a systematic and unified approach. The design of this function admits several sources of monitoring data, not being limited to ground assets metrics only; by combining under a common layer the space assets monitoring data as well (and potentially other mission parameters), the monitoring data analysis functions can perform more powerful correlations and the operator shall be able to stablish KPIs based on all these parameters.

Moreover, due to the high amount of monitoring data expected to be processed, support from AI algorithms is required at some point to make a better use of the information. AI analytics may help to discover new correlation and predict the system behaviour in order to enhance the system monitoring capabilities and the ability of the FDIR functions to take better decisions.

Once the logical functions have been defined, the next step is the identification of the physical components that shall implement them.

3. Transitioning from the monolith

At the earliest stages of the Galileo mission, the different sub-systems that implement the GCS functions were designed as monolithic applications. Due to the industrial organization of the project and the technical means back then, having all the software and hardware parts related to big functional blocks unified under sub-systems developed and deployed by different companies was easier to develop, deploy and manage.

However, with the evolution of the project, the introduction of virtualization and the arrival of new architectural solutions (such as containers), this approach is becoming obsolete and has demonstrated having several drawbacks:

- Original subsystems have grown beyond their initial functions, implementing supporting capabilities that are duplicated in many other places withing the ground segment (e.g., common libraries, databases, monitors, data processors, etc.)
- GCS Technologies are very heterogeneous, as each sub-system has evolved independently. This affects MMIs, databases, interfaces, etc., as different approaches for similar problems are applied, ending up in complex architectures difficult to maintain and operate.
- Software monoliths are difficult to evolve to adopt new technologies, as they usually require complete and costly re-coding processes.
- Monolithic applications are usually difficult to scale.

- Sub-systems are hardly customizable, and any minor improvement or fix usually requires the development and deployment of a new version of the software.

Considering the monitoring and control system functional decomposition described previously, the transition towards a service-oriented architecture (SOA) seemed a logical step [3]; the expected benefits of this approach would be the reusability of services, maintainability, and the possibility of performing parallel development and deployment activities. Under this schema, each independent function would be provided by a single service (or a limited set of services) covering the required function, enabling the use of specialized COTS with a single purpose within the GCS application environment. And, going one step beyond, the potential advantages of a microservices architecture were also available, aiming to reduce the validation and deployment efforts and boost the SecDevOps pipelines and related activities.

Nevertheless, direct transitioning to any of these schemas from the legacy system is not simple nor straightforward. The main drawback of SOA and microservices architectures is their complexity (from development to management), and in already complex designs such as the GCS, this solution cannot be applied directly to all components without taking unnecessary risks.

The final approach chosen foresees a gradual decomposition of the current monolithic components into smaller services, mostly based on SOA and applying microservices in specific cases with clear expected benefits. In order to decouple the IT infrastructure from the monitoring and control services, the new physical architecture makes use of containerization and Kubernetes as orchestration solution. The different functions have been implemented using open-source COTS to provide the required services, combined with minor ad-hoc developments which help to better integrate the different solutions.

Using open-source COTS has demonstrated to have many benefits. These tools are developed by experts in their business areas, ending up in highly optimized applications with a very good user experience. Then, the applications can be customized for GCS use cases, as needed, by a small team of people which is expert on the Galileo business logic. They are usually supported by large communities that provide frequent patches solving bugs and vulnerabilities and implementing new features while providing documentation and training resources that simplify the learning period. On the contrary, the development of an ad-hoc solution would require a bigger team and it has demonstrated to require a larger development period and to be subject to a bigger number of anomalies. Additionally, in the case of HMIs, associated technologies becomes quickly outdated, with a complicated maintainability and lack of validation tools.

It should be noted as well that the use of these components comes with some drawbacks. The most relevant is that COTS may not fully comply with the project requirements, so therefore choosing highly configurable COTS it’s critical to reduce this risk. Performing some customizations on open-source COTS is also a useful approach in some cases (to develop plugins implementing missing functionalities), but assuming that this approach may complicate future software upgrades as it requires to migrate the changes on every product release. Another potential disadvantage is that the response time to the anomalies found on the application is linked to the development cycle of the COTS community; although this usually does not suppose an issue in highly active communities, as new releases of the COTS are being published almost every month, monitoring their health periodically (e.g. looking for potential key personnel leaving the project or market replacement by other alternatives) becomes a new relevant task. The effort associated to cybersecurity watching is not negligible either, but in this case it’s compensated by the global community strength to keep their solutions safe.

In any case, despite these potential drawbacks, the use of open-source COTS has been proven to be a solid solution that allows modernizing the Galileo GCS in terms of security and functionality, thanks to the support of consolidated selection and maintenance procedures to ensure risk mitigation.

4. The event-centric approach

An additional evolution of the GCS infrastructure that contributes to improve the monitoring approach is the transition towards a data-centric and event-driven architecture, which shall allow a better internal interface monitoring strategy in real time. The legacy data exchange model was oriented to communicate isolated and independent software components (monoliths) using file-based interfaces and SNMP monitoring, this approach eases the development and testing of this components in isolation, but the status information that can be reported to the system operator is quite basic and the control options are limited.

In opposition to this model, in a data-centric architecture, data is stored independently of software applications [1], and it’s available to all the GCS components once generated, reducing the dependency on transfer mechanisms and simplifying the interfaces monitoring and reporting to the operator. The implementation of this architecture is based on the provision of different shared data storage techniques (e.g. databases, object storage, indexed storage, etc.) that can be used by the GCS components to store their data.

Then, the event-driven architecture is used to trigger and communicate between decoupled services; this deviates slightly from a service-oriented architecture, as instead of accumulating data in data lakes, data is in flight and tracked while moving place to place. Under this schema, data is still important, but events become the most relevant component. Information is emitted by a producer as discrete events, and it is received asynchronously by the consumer at a later point. The implementation of this architecture is usually based on COTS providing asynchronous message queue functions based on producer/consumer paradigms.

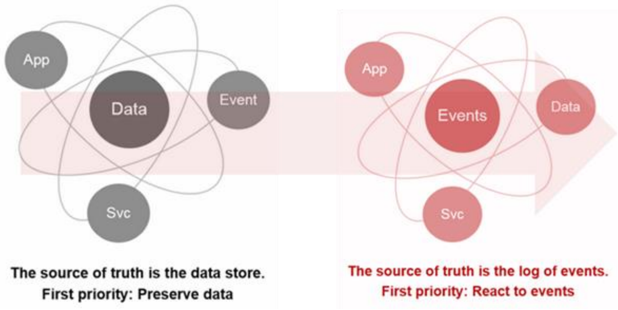


Fig 2 – Transition from data-driven to event-centric model

Monitoring the status of the different services conforming the GCS is then simplified with these architectures, as the COTS supporting them already include status reporting functions by default; these functions can then be easily accessed by the GCS data gathering functions to be incorporated to the aggregated metrics made available to the system operator.

5. GCS situational awareness

The last component of the GCS monitoring architecture is the situational awareness function, aimed to provide to the Galileo operator a complete overview of the status of the ground and space assets, at logical and physical levels, allowing to monitor the system automation and providing quick access to the control interface. The information is retrieved from the data aggregation function and made available to the operator as a set of hierarchical dashboards, from the high-level system status overview to the lower monitoring level dashboards associated to a single monitored asset (e.g. a virtual machine or interface). As mentioned previously, once all the data provided by the ground and space assets is aggregated, it's possible to display both constellation and infrastructure information either independently or combined in order to ease operational KPIs monitoring in both areas. Dashboard navigation is designed to allow an intuitive browsing, allowing the user to access detailed component information easily.

3A5	3A6	3A7	316	261	262	263	264	265	266	267	268	269	26A	26B	26C	26D	26E	2C5
2C6	2C7	2C8	2C9	2C0	2C1	2C2	109	10B	10C	10D	10E	10F	112	113	136	137	138	139

Ongoing Contacts

Station	Responsibility	Domain	Family	Status	LOS
TTCF-1	GCC-D	262	FOC	OOL detected	58 minutes
TTCF-3	GCC-I	265	FOC	ON GOING	1 hour
TTCF-4	GCC-D	10E	FOC	ON GOING	9 minutes
TTCF-5	GCC-I	3A7	IOV	ON GOING	15 minutes
TTCF-6	GCC-D	109	FOC	ON GOING	36 minutes

Incoming Contacts

Station	Responsibility	Domain	Family	AOS
TTCF-1	GCC-I	263	FOC	1 hour
TTCF-3	GCC-D	3A5	IOV	58 minutes
TTCF-4	GCC-D	138	FOC	8 minutes
TTCF-5	GCC-D	2C5	FOC	18 minutes
TTCF-6	GCC-I	112	FOC	45 minutes

Fig 3 – High level constellation status dashboard

Published by the Mohammed Bin Rashid Space Centre (MBRSC)
on behalf of SpaceOps, with permission and released to the MBRSC to publish in all forms.

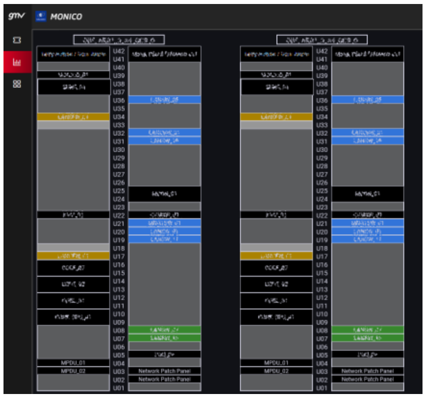
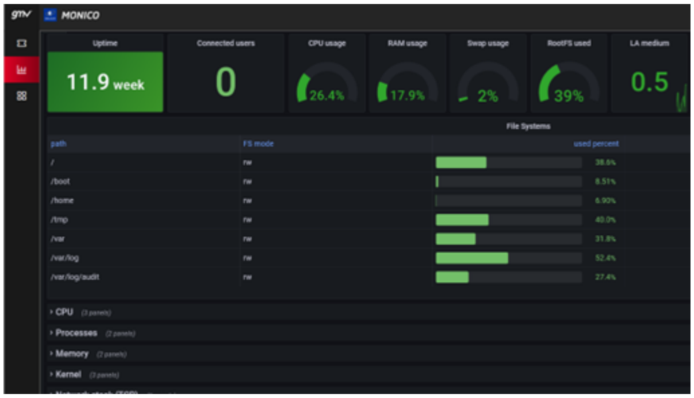


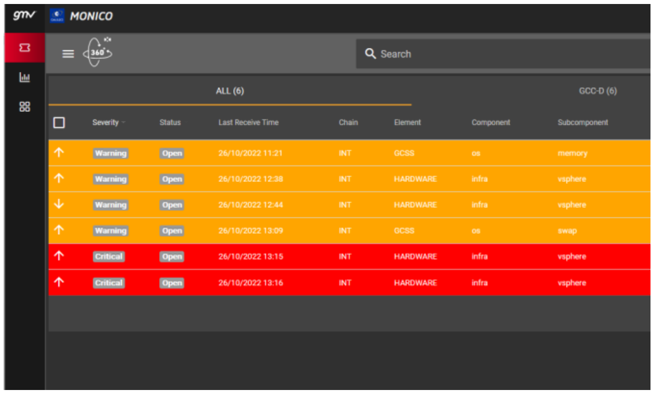
Fig 4 – Infrastructure (rack) status dashboard



path	FS mode	used percent
/	rw	38.6%
/boot	rw	8.51%
/home	rw	6.90%
/tmp	rw	40.0%
/var	rw	31.8%
/var/log	rw	52.4%
/var/log/audit	rw	27.4%

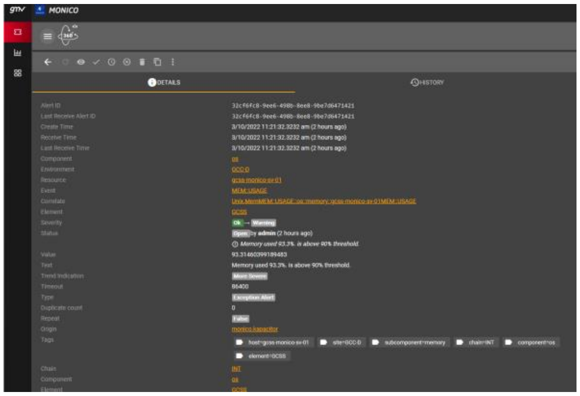
Fig 5 – Infrastructure component status dashboard

Metrics information are complemented with a flexible alerting system that allows the definition of custom thresholds for any monitored parameter and KPI. Relevant alerts per component are aggregated in the corresponding dashboards and can be explored individually in a dedicated tool to be analysed, tracked and managed.



☐	Severity	Status	Last Receive Time	Chain	Element	Component	Subcomponent
↑	Warning	Open	26/10/2022 11:21	INT	GCSS	os	memory
↑	Warning	Open	26/10/2022 12:38	INT	HARDWARE	infra	vsphere
↓	Warning	Open	26/10/2022 12:44	INT	HARDWARE	infra	vsphere
↑	Warning	Open	26/10/2022 13:09	INT	GCSS	os	swap
↑	Critical	Open	26/10/2022 13:15	INT	HARDWARE	infra	vsphere
↑	Critical	Open	26/10/2022 13:16	INT	HARDWARE	infra	vsphere

Fig 6 – Global alerting component



Alert ID	32cf6fc8-9ee6-498b-8ee8-9be7d6471421
Last Receive Alert ID	32cf6fc8-9ee6-498b-8ee8-9be7d6471421
Create Time	3/10/2022 11:21:32.3232 am (2 hours ago)
Receive Time	3/10/2022 11:21:32.3232 am (2 hours ago)
Last Receive Time	3/10/2022 11:21:32.3232 am (2 hours ago)
Component	os
Environment	GCC-D
Resource	gcss-monico-sv-01
Event	MEM_USAGE
Correlate	Unix.MemMEM_USAGE:os:memory:gcss-monico-sv-01MEM_USAGE
Element	GCSS
Severity	Ok → Warning
Status	Open by admin (2 hours ago) Memory used 93.3%. is above 90% threshold.
Value	93.31460399189483
Text	Memory used 93.3%. is above 90% threshold.
Trend Indication	More Severe
Timeout	86400
Type	Exception Alert
Duplicate count	0
Repeat	False
Origin	monico.kapacitor
Tags	host=gcss-monico-sv-01 site=GCC-D subcomponent=memory chain=INT component=os element=GCSS
Chain	INT
Component	os
Element	GCSS

Fig 7 – Specific alert details

Ground assets control interface can be accessed directly from the situational awareness dashboards, in case the operator needs to command an operation manually; having a dedicated control interface allows providing more powerful tools to track properly the execution status of a manual command, while keeping a centralized operational position for both monitoring and control tasks.

6. Conclusions

As result of the GCS monitoring and control system evolution described previously, GCS assets monitoring information is now aggregated and displayed in a more synthesized approach through flexible dashboards, with a direct benefit on how routine operations and KPIs are managed: alerts and monitoring metrics are more clear, automated processes can be supervised more efficiently, and problem detection and reaction times are improved as the operator has quick access to the ground assets control interface directly from the monitoring interface.

By combining both infrastructure and constellation metrics under the same data aggregation layer, more powerful data correlation techniques can be applied to improve the system automatic FDIR capabilities, enabling also the use of AI mechanisms to exploit this data in depth. Operators may benefit from this approach to improve the system automation and maintainability, as the system will be capable to predict its future behaviour based on historical records.

The use of a service-oriented architecture has demonstrated to have clear advantages in terms of reusability, maintainability, development and deployment, and has enabled the implementation of the different functions required with specific open-source COTS; as result of this approach, the new system required less time to be designed and built, and the operator feedback could be considered in earlier stages of the development in order to adapt better the final solution to their needs.

Main challenges faced during this process include the proper hardening, isolation and strict patching policy of the selected COTS to comply with the cybersecurity guidelines, a careful analysis of the COTS sources and license conditions, and thorough validation campaigns to confirm the suitability of the solution in line with Galileo standards.

References

[1] E. Yuan, P. Hwu, V. Rohr, *Modern data engineering for a data-centric ground enterprise*, Ground System Architectures Workshop (GSAW) 2022, The Aerospace Corporation.

[2] Sonia Toribio, Mario Merri, Sandro Matussi, Alejandro Pena and Chris Payne *Galileo IOV Phase: Design and Implementation Challenges of the Galileo Ground Control Segment*, SpaceOps 2006, AIAA 2006-5750 https://arc.aiaa.org/doi/pdf/10.2514/6.2006-5750

[3] Mission Operations Service Concept, Informational report, CCSDS 520.0-G-3, Green Book, December 2010

[4] H. Knoche, and W. Hasselbring. Using microservices for legacy software modernization. IEEE Software, 35(3), pp. 44–49. 2018.