INTEGRATING SMALL SATELLITE MISSIONS INTO A MULTI-MISSION OPERATIONS ENVIRONMENT: A CONSOLIDATED APPROACH WITH EGS-CC

Tobias Brügge [(1)], Gary Morfill [(2)], Markus Hobsch [(1)], Christian Stangl [(1)], Pierre-Alexis Lagadrilliere [(3)]

[(1)] Department for Mission Control Systems, German Space Operations Center (DLR), Münchener Straße 20, 82234 Weßling, Germany

[(2)] Department for Mission Operations, German Space Operations Center (DLR), Münchener Straße 20, 82234 Weßling, Germany

[(3)] Department for Communication and Ground Stations, German Space Operations Center (DLR), Münchener Straße 20, 82234 Weßling, Germany

ABSTRACT

The German Space Operations Center (GSOC) has a long history of operating large-scale and prestigious space missions. The key to such multi-mission operations is a reliable and interoperable multi-mission operations environment, based on highly standardized TM/TC communications. For this reason, the core mission operations systems and ground station equipment used by GSOC rely on established standards like CCSDS 132.0-B-3, CCSDS 232.0-B-4, CCSDS 133.0-B-2, and ECSS E-ST-70-41C.

With the advent of New Space, recent years have seen an increasing prevalence of small satellite missions. Such missions follow a very different mindset, being much cheaper and more open to risk, challenging traditional mission operations concepts and standards.The first mission of this kind to be operated at GSOC is CubeL, a CubeSat launched in early 2021, which relies on the CubeSat Space Protocol (CSP). CubeL has provided GSOC with valuable insights into the integration and operations of a small satellite mission within the existing multimission environment. However, it also laid bare the challenges of bringing two very different worlds together: established standards like CCSDS and ECSS on the one side, and New Space concepts like CSP on the other. As a result, the integration of CubeL required the implementation of a number of custom solutions, which do not promote GSOC's multi-mission strategy.

In this paper, we propose a new concept for integrating small satellite missions, such as CubeL, into the GSOC multi-mission environment. This strategy is based on the European Ground System - Common Core (EGS-CC), which is the core building block of GSOC's future MCS. Using CubeL as a reference, we review the challenges of integrating a small satellite mission into the established multi-mission environment. Building upon this analysis, we describe a more consolidated solution based on EGS-CC, which aims to extend the GSOC multi-mission strategy by providing a standardized and flexible framework for small satellite missions. Finally, we identify some key considerations for small satellite manufacturers looking to align with a more standardized strategy for operating small satellite missions.

Keywords – GSOC, EGS-CC, Multi-Mission, Operations, Small Satellites, Standards, CSP

1 INTRODUCTION

With a track record spanning over five decades, GSOC has a long history in space operations. Over the years, notable missions operated by GSOC include the Spacelab missions D1 (1983) and D2 (1993), operations of the Columbus module of the International Space Station (ISS) since 2008, and satellites such as GRACE (2002), TerraSAR-X (2007), TanDEM-X (2010), and EnMAP (2022) [1]. These successes were made possible by the multi-mission environment at GSOC, which consists of a core set of tools based on high standardization and reusability. This infrastructure enables GSOC to support a large number of missions and to provide reliable, around-the-clock operations for customers. However, the consequence of this approach also is that missions not compliant with the established standards and protocols may face barriers to integration and support.

Driven by decreasing launch costs, recent years have seen the landscape of space missions rapidly evolving, with an increasing number of manufacturers entering the market, in particular, of small satellites. These new manufacturers, often associated with the New Space [2] movement, have a more agile and entrepreneurial mindset. Often, they opt to follow novel approaches to operating space missions, using new concepts, new protocols, and new tools. The appeal of this to customers is clear, promising dramatically lower costs for both launch and operations.

Unfortunately, there are also some drawbacks. Typically, the involvement of the manufacturer in the space mission is limited to the Launch and Early Orbit Phase (LEOP), after which control of the mission is transferred to the mission control center. If the mission does not conform to the established standards the control center's infrastructure is based on, the control center is either forced to use tools provided by the manufacturer, or to integrate the non-compliant spacecraft by extensive modification and adaptation of the ground segment and potentially also the space segment. Not only does this diminish the control center's capability to leverage its own infrastructure for mission operations, but also potentially introduces design changes in the mission for which there is no experience on either side. This leads to higher costs not foreseen in the budget of a small satellite mission, resulting in low-budget solutions with increased risk to mission success.

In this paper, we describe the challenges of integrating a non-standard mission into the GSOC multi-mission environment at the example of CubeL, a CubeSat currently operated at GSOC. An alternative approach is proposed for integrating small satellites like CubeL, so that such challenges could be mitigated in the future. This approach is based on EGS-CC, which is poised to become the next generation MCS at the heart of GSOC's multi-mission environment. Finally, some considerations are made for small satellite manufacturers looking to align with such a strategy.

2 GSOC MULTI-MISSION VS. SMALL SATELLITES

This section gives an overview of the GSOC multi-mission environment, as it is typically used for unmanned missions operated by GSOC. This is contrasted against the design of a typical small satellite mission that might have to be integrated. The challenges of integrating a small satellite mission into the GSOC multi-mission environment are illustrated at the example of CubeL, a CubeSat mission that is being operated by GSOC since early 2021.

2.1 ESTABLISHED SYSTEMS AND STANDARDIZATION

The GSOC multi-mission environment consists of a set of core tools, which can easily be adapted and reconfigured to support different space missions, provided that the mission supports CCSDS and ECSS standards. Figure 1 [3] shows a missions operations segment, as it is typically used to operate unmanned missions. At the heart of the multi-mission environment is GECCOS, a SCOS-2000-based MCS, which is used for TM/TC communications with the spacecraft.

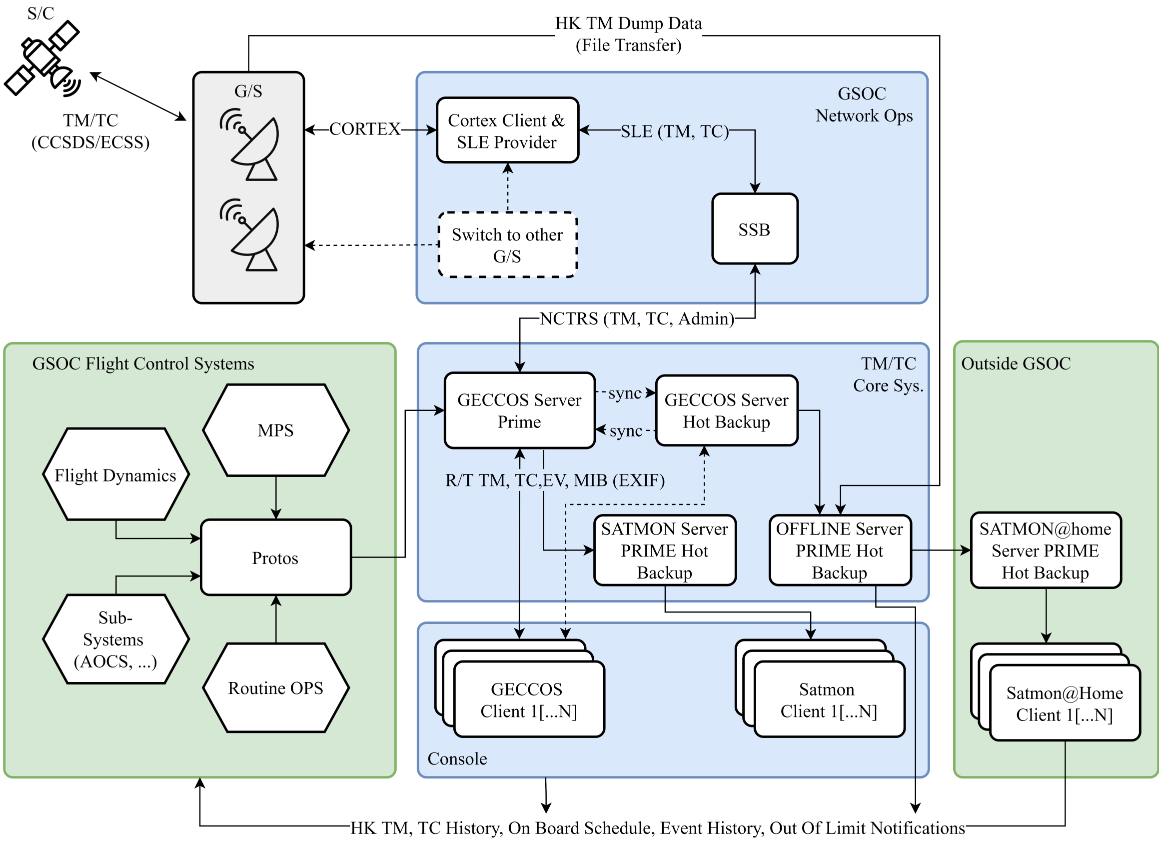


Figure 1. Mission operations segment in the GSOC multi-mission environment [3]

Using the NCTRS protocol, GECCOS connects to an SLE Switch Board (SSB). This allows access to a worldwide ground station network via the Space Link Extension (SLE) protocol, so that there are multiple contacts during the orbit of the spacecraft where TM/TC can be transmitted. For this to be possible, the TM/TC structure must comply with the CCSDS protocol. In addition, GECCOS is designed to process TM/TC packets by use of a Packet Utilization Standard (PUS) according to ECSS. If the spacecraft complies with the employed PUS, this enables an evaluation of the transmission status, reception and on-board processing of telecommands sent from the ground, and a precise definition of the expected response from the spacecraft according to the implemented PUS services. This enables highly reliable TM/TC communications and gives a clear picture of the on-board state of the spacecraft at every moment in time. Moreover, it enables interoperability with other organizations, like control centers and ground stations.

Two elements are central to enabling the TM/TC communications. The first element is the Mission Information Base (MIB), in which each single TM/TC that can be sent between ground and space segment is precisely defined. The second element is the procedures that specify the exact order in which TC must be sent to the spacecraft in order to achieve a given task.

The TCs to be sent to the spacecraft are typically prepared ahead of a contact by subsystem engineers and mission planning according to predefined procedures. These TC are assembled into a TC stack, which is executed during a contact. This ensures that the TC are uplinked to the spacecraft in the precise sequence defined in the procedures. The tool that facilitates this is ProToS, which is a procedural tool suite developed and maintained by GSOC. Not only does ProToS allow subsystem engineers to define, edit, validate, and version procedures, which can be executed directly from ProToS by use of the GECCOS API, but it also provides a framework that can be used to automate operational activities.

For monitoring TM received from the spacecraft, GSOC develops and maintains the SATMON system. SATMON gives subsystem engineers and stakeholders access to real-time and historical TM from the spacecraft, both in the control room, and also from outside the control center via SATMON@home.

2.2 THE PROBLEM OF CUSTOM PROTOCOLS IN SMALL SATELLITES

The development of the multi-mission infrastructure described in Section 2.1 is the result of multiple decades of experience in space operations. Traditionally, the missions operated by GSOC have been high-profile, international missions with budgets exceeding millions of euros, where customers expect the highest level of safety and reliability. Especially in international projects, with stakeholders from around the world involved in development and operations, such a high level of quality can best be achieved through adherence to a set of common standards. This is the reason why standardization efforts like CCSDS/ECSS exist and why GECCOS, as the centerpiece of the GSOC multi-mission infrastructure, is designed to comply with protocols such as CCSDS 132.0-B-3, CCSDS 232.0-B-4, CCSDS 133.0-B-2, and ECSS E-ST-70-41C.

This design choice has consequences for missions that GSOC can support and for the requirements placed on spacecraft manufacturers. Adherence to CCSDS and ECSS standards ensures reliable and high- quality operations, but it also means that GSOC can only integrate missions from compliant manufacturers, lest it modifies and adapts its multi-mission infrastructure. Certain formats and deliverables are also expected to be provided by the manufacturer. For example, it has become an ”industry standard” that the manufacturer provides a MIB and the required procedures so that SCOS-2000-based systems like GECCOS can be used to operate the spacecraft.

Unfortunately, this is not always the case. Especially with the emergence of the New Space movement, there is an increasing prevalence of small satellite manufacturers following new approaches to space operations. These manufacturers cater to a very different type of customer from the large organizations or agencies that can afford the large budget missions traditionally operated at GSOC. Instead, with falling launch costs, more and more smaller entities with smaller budgets, like universities or research institutes, are eager to deploy payloads and experiments into orbit. Additionally, commercial interest is growing to deploy large satellite constellations. Small satellite manufacturers are able to significantly bring down the cost of deploying payloads in orbit, thereby giving this market increased access to space.

They achieve this by reducing the size and mass of the spacecraft, adhering to standard size and form factors [4], and utilizing commercially available components for modular design and scalability. They also choose a more simplified approach to system testing and validation, foregoing strict qualifications and redundancy concepts often found in traditional missions. Costs are further reduced by simplifying the approach for operating the spacecraft. For example, the manufacturer might choose to deliver a complete plug-and-play solution to the customer, with access to one or more ground stations and a set of tools that enable monitoring and control of the spacecraft. These custom solutions do not necessarily comply with CCSDS and ECSS standards. Instead, they can be based on simpler solutions, either proprietary to the satellite manufacturer or reusing existing open-source functionality like, for example, the SatNOGS infrastructure [5] or the CSP protocol [6].

Of course, this approach can also have drawbacks. Particularly in the challenging environment of space, the lack of redundancy and safety measures increases the risk of mission failure. The tools provided by a manufacturer to operate the spacecraft may not have the sophistication and features of the services provided by a control center, and a customer may not have the resources to conduct reliable operations a control center can guarantee. Additionally, support for open-source and third-party solutions may not always be available. Relying on proprietary or non-standard solutions may limit interoperability and compatibility with other systems, particularly when an organization does not adhere to the same standards as the manufacturer. This can restrict the potential market for the small satellite manufacturer and impede collaboration with other organizations.

2.3 CASE STUDY: INTEGRATION OF CUBEL INTO GSOC MUM

The following case study illustrates the difficulties that arise when operational practices of a traditional control center such as GSOC converge with those of a typical small satellite mission. The spacecraft under examination is CubeL, launched in early 2021, which is the first small satellite to be operated at GSOC. CubeL is a CubeSat based on a 3U platform and was designed as a low-budget mission for technology demonstration.

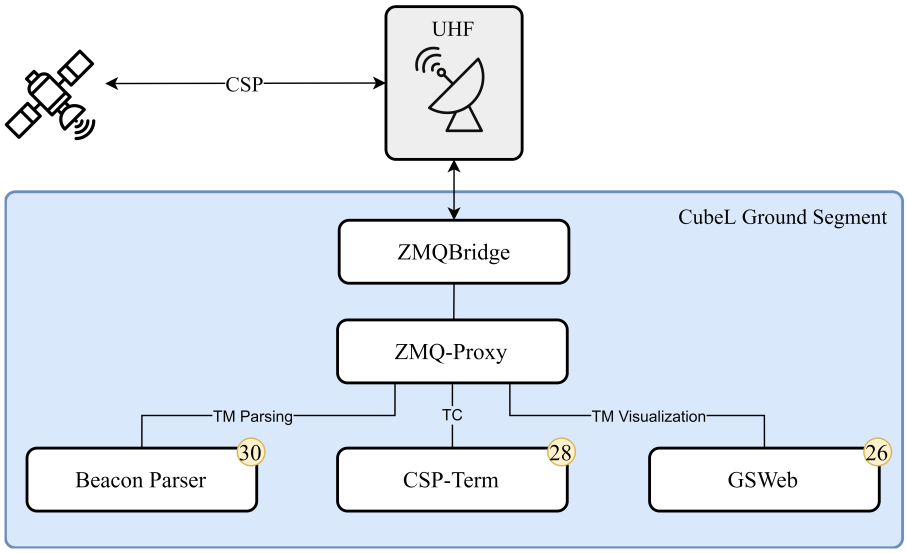


Figure 2. CubeL Ground Segment provided by the Manufacturer (simplified)

To operate the spacecraft, the manufacturer delivered a software toolkit to be used for monitoring and control and a Ultra High Frequency (UHF) ground station. The ground segment is shown in simplified form in Figure 2. It includes a number of software modules (referred to as nodes) that communicate with each other using the ZMQ messaging framework. The core component is CSP-Term (node 28), which an operator can use to send commands. As shown in Figure 3, CSP-Term is a simple command-line application. It takes commands with parameters as text input and uses the CSP API to generate the CSP packets that are transmitted to the spacecraft. Responses from the spacecraft are then processed in the implementation of the CSP API.

The manufacturer's toolkit is intended to enable plug-and-play functionality. However, this approach is not in line with the multi-mission strategy at GSOC, which seeks the reuse and maintenance of its multi-mission core tools to operate all missions (see Section 2.1). Moreover, the functionality provided by the manufacturer's toolkit could equally be accomplished and potentially exceeded by using existing GSOC multi-mission tools. For example, the functionality of the Beacon Parser (node

30) and GSWeb (node 26) modules in Figure 2, which take care of parsing and visualizing telemetry, can also be achieved with SATMON. The manufacturer's toolkit also does not allow for the automation of typical operational activities, as would be possible with ProToS.

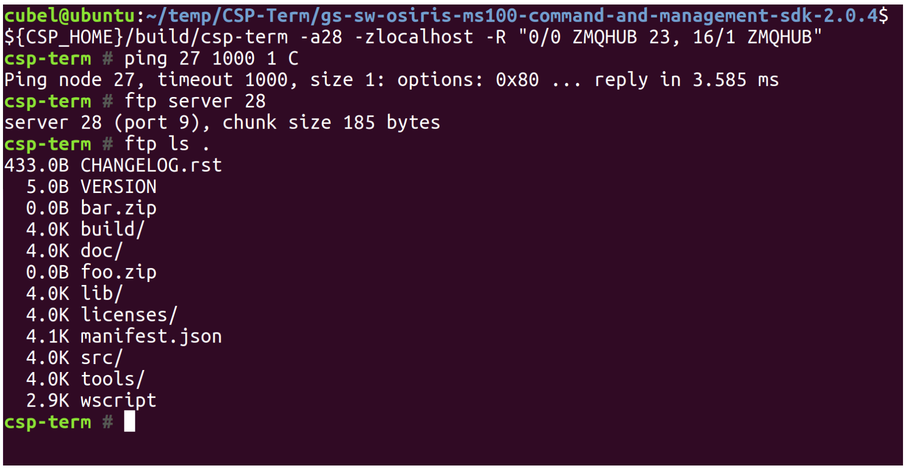


Figure 3. Executing commands in the CSP-Term tool

Unfortunately, due to the different standards and design philosophies, it was difficult for GSOC to leverage its own systems and capabilities.

One issue was the mission design based on the CSP protocol. Commands and telemetry are transmitted as CSP packets, which can only be understood by CSP-capable ground stations and tools. It is not possible to use the CCSDS-based space links of the antenna network accessible to GSOC without modification. Another issue was the non-existence of a MIB. Traditional operations with a SCOS-2000 based system like GECCOS rely on the assumption that each single TM/TC element is defined in a MIB. Telecommands sent to the spacecraft according to defined procedures are transported in CCSDS/ECSS packets and are subject to verification of PUS services according to ECSS standards. Telemetry from the spacecraft is similarly sent in PUS packets that identify the type and purpose of the packet. Because the information for handling TM/TC is encoded in the MIB, this makes the MIB a crucial deliverable for GSOC to operate a satellite mission.

The CSP protocol, in contrast, knows nothing of ECSS and PUS. A concept such as the MIB simply does not exist. Instead, telecommands are part of the CSP API (written in C), which can be extended and modified. Thus, sending a command in CSP essentially means calling a C function in the implementation of the CSP API, giving a developer substantially more freedom to define interactions with the spacecraft. For example, it is possible that a single input by the operator into the MCS may trigger the exchange of multiple CSP packets between the space and ground segment. Similarly, there is more freedom in structuring the telemetry sent from the spacecraft. For example, the TM structure of CubeL is defined in JSON format, so that the spacecraft can be configured to transmit parameters of interest at regular intervals or collections of parameters can be requested by an operator.

In summary, handling of TM/TC in the CubeL mission not only presents a different design philosophy, but made integration with GSOC systems exceedingly difficult. To enable the GSOC multi-mission infrastructure to be used for routine tasks such as commanding, monitoring, offline processing, and mission planning, extensive modifications and adaptations of both the space and the ground segment were necessary [7]. For example, this included patching the firmware of the on-board modem, the development of a new tool to enable commanding via the GSOC S-Band link, the definition of an auxiliary MIB for parsing and visualizing telemetry, a new tool for offline processing, and modifications of the mission planning system. Additionally, a custom solution was implemented to enable automated commanding of the spacecraft. This led to increased costs and complexity for what was initially envisioned as a low-budget CubeSat mission, exacerbated due to contingencies related to on-board problems and issues with the UHF ground station over the course of the mission.

3 ALTERNATIVE APPROACH: INTEGRATION WITH EGS-CC

This section proposes an alternative approach to integrate a small satellite mission like CubeL into the multi-mission infrastructure at GSOC. This approach is based on EGS-CC, which is set to become the core of the next generation MCS at GSOC, replacing GECCOS as the central component in the multi-mission infrastructure. The assumption in the described approach is that the satellite is based on the CSP protocol. However, the described methodology could also be applicable to other types of small satellite missions and protocols.

3.1 AN OVERVIEW OF EGS-CC

EGS-CC is an initiative of European stakeholders in the space industry to establish a common infrastructure that supports of all types of space missions [8]. It offers a comprehensive range of services typically required of a heritage system to operate space missions. This includes critical functions like Monitoring and Control (M&C), display systems, automation, user management and authentification, and also interfaces for subsystems like mission planning and flight dynamics.

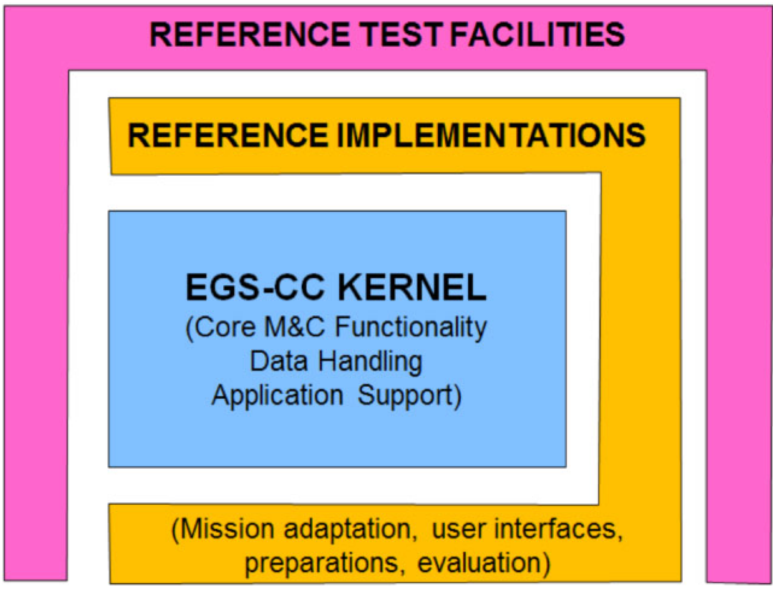


Figure 4. EGS-CC Architecture [8]

EGS-CC has a layered architecture, which is shown in Figure 4. The system has a component-based, service-oriented design, which means that each layer is constituted by a set of components that allows the definition of modular systems. The core layer is the EGS-CC kernel, which enables core monitoring and control functionality, data handling and application support. The kernel is designed to be generic, so that it is agnostic to the application domain. This means that the kernel can support operations of a space mission, but may also be used for M&C in other use cases. The adaptation of an EGS-CC based system to the specific application occurs in the Reference Implementation (RI). The RI may consist of a set of components that enables the operation of a CCSDS/PUS based space mission with EGS-CC. However, the RI may also be modified or even completely replaced to support a different use case.

Central to an EGS-CC based system is the Tailoring Data Model (TDM). The TDM defines the data model of the system managed by EGS-CC, but also of the EGS-CC system itself. It also allows customization of the packetization layer and contains consistency rules related to the semantics of the data model. The idea is to use the TDM to tailor the abstract Monitoring and Control Model (MCM) component in the EGS-CC kernel, which provides the core functionality for M&C operations of a controlled system. The tailoring involves decomposing the controlled system into a hierarchy of monitoring and control elements (MCE) containing the required monitoring and control information. This information is abstracted into basic aspects like parameters of the controlled system, events that can occur, or activities that can be performed. The abstract aspects are mapped to the implementation aspects of the actual controlled system. For example, a parameter can be mapped to a packet, or an activity can be mapped to the implementation of a process. This abstraction makes TDM agnostic to the application domain (just like the EGS-CC kernel) and allows the tailoring of data models for various kinds of use cases.

3.2 DEFINING A REFERENCE IMPLEMENTATION FOR CSP

The case study in Section 2.3 identified a number of issues for integrating a small satellite mission into the GSOC multi-mission infrastructure. The key problems were caused by a non CCSDS/ECSS packet format and different approach for handling TM/TC. As a result, it was only possible to integrate the mission by modification and customization, with reduced access to the full functionality of the multi-mission infrastructure and decreased reliability. With EGS-CC set to replace GECCOS as GSOC's MCS, this presents an opportunity to solve these kinds of challenges. This could be facilitated by EGS-CC's generic and modular architecture and by the ability to adapt the system to different use cases in the RI. Additionally, with abstraction of controlled systems into M&C aspects, the TDM gives great freedom to create data models for different types of controlled systems.

The EGS-CC system is designed as a more flexible framework to support various types of uses cases and missions. As described in Section 3.1, EGS-CC consists of a generic kernel and a RI, which adapts an EGS-CC based system to a specific application. Since the RI is completely exchangeable, it is possible to define a new RI that supports a CSP based space mission like CubeL. This is illustrated in Figure 5, which shows as an example the TC chain of an EGS-CC based system with a CSP based RI. As can be seen, an operator can command the spacecraft by invoking an activity in the EGS-CC user interface. The activity is routed through the MCA and MCM components of the EGS-CC kernel to an activity processor (AP) in the RI[1]. This activity processor is simply another EGS-CC component called CSP, which instantiates the abstract activity into an actual CSP packet. It achieves this by calling a function from the CSP API provided by the manufacturer, just like it is done the manufacturer's own MCS (see the CSP-Term tool described in Section 2.3). From the CSP component, the CSP packet can then be routed via a second component containing the implementation of the CSP interface (for example ZMQ) to a ground station that transmits the packet to the spacecraft. Because the CSP component uses the manufacturer's API, the response from the spacecraft will directly be available as the return value of the called API function, which can be mapped to the result of invoked activity. By tailoring the TDM, the CSP RI can also be extended to support the CSP packet format, which would allow processing of TM data with the components in the EGS-CC kernel.

[1]The RI can contain multiple activity processors. For example, it would be possible to define an individual activity processor for each command type or even for different protocols.

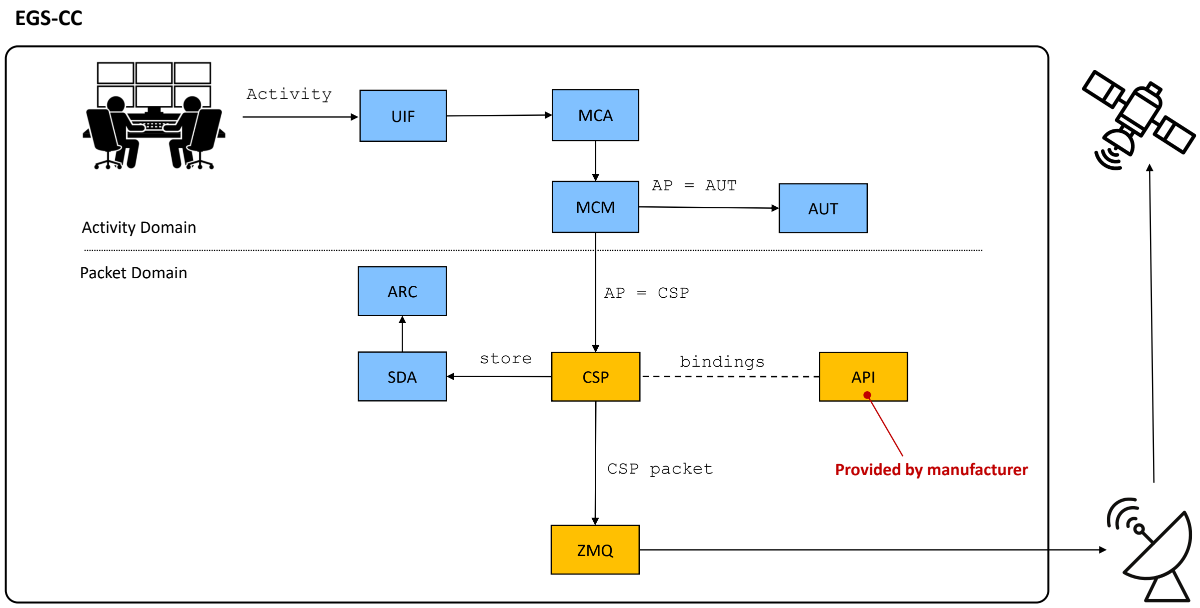


Figure 5. EGS-CC with a CSP Reference Implementation

The benefits of this approach are manifold. Firstly, it enables design philosophies and protocols like CSP to be accommodated next to traditional protocols like CCSDS/ECSS. Secondly, the abstraction of implementation aspects, such as calling a function in the CSP API or parsing a CSP packet, facilitates interfacing with other GSOC multi-mission tools. This enables GSOC to leverage its full multi-mission infrastructure to support the mission effectively. The ability of GSOC to use its own tools reduces reliance on tools and support of the manufacturer for mission operations. Thirdly, it allows requirements placed on a manufacturer to be more aligned with the manufacturer's capabilities. For example, the current requirement to integrate a mission at GSOC is that the manufacturer provides a MIB, which a manufacturer using a protocol like CSP inherently cannot comply to. With the EGS-CC based approach, however, the requirement shifts to exposing the manufacturer's CSP API, so that it can be mapped to EGS-CC activities, and to defining TM structures in the TDM.

Of course, an open question is the interface between EGS-CC and the ground station. If the CSP packet format is used, this has implications on how the packet will be transmitted. A possible scenario with CCSDS compatibility has been described in [7], another approach could foresee the use of a project-specific software-modem, as described in Section 4.4.

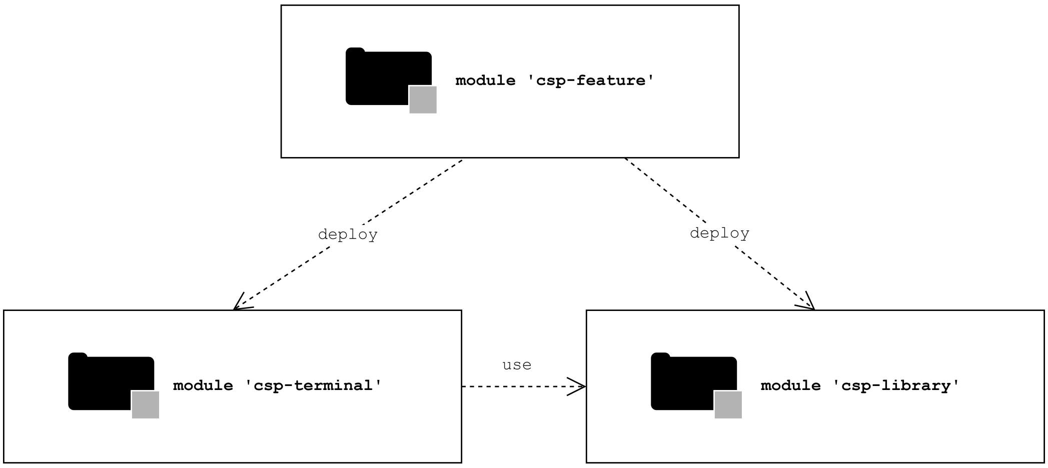
3.3 CSP COMPONENT DESIGN

Key to enabling the CSP based reference implementation of EGS-CC is the implementation of an EGS-CC component that enables the instantiation of activities into CSP packets. This component is referred to here as the CSP component. Figure 6 shows a design of the CSP component and the toolchain that is used. The component is developed in an Apache Maven project, consisting of three modules.

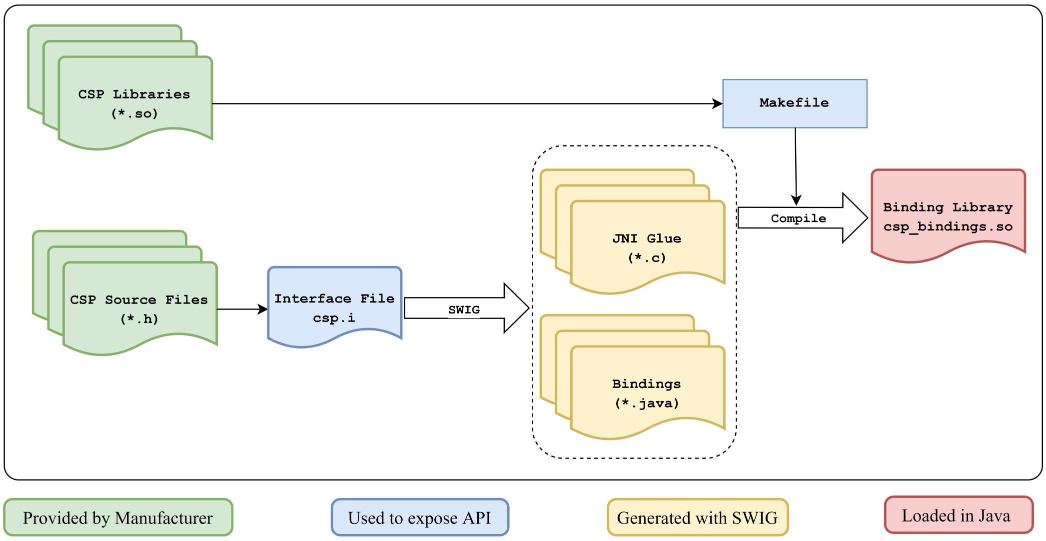
The first module is *csp-library*. It exposes the CSP API provided by the manufacturer to be used in EGS-CC. According to the CSP API exposed in an interface file, SWIG[2] is used to generate a set of

[2]SWIG (Simplified Wrapper and Interface Generator) is an open-source tool available at *https://www.swig.org*

Java bindings for the CSP source files (written in C) and some Java Native Interface (JNI) glue code. The JNI glue code is then compiled into a binding library that can be loaded in Java. This allows calls of the generated bindings in Java to access the implementation contained in the CSP libraries. In the Karaf[3] runtime of EGS-CC, this compiled library is then packaged in the deployed *csp-library* bundle as native code.



(a) CSP Component Modules



(b) Toolchain used to generate CSP Bindings

Figure 6. CSP Component Design

The second module is *csp-terminal*, which loads the compiled library provided by the *csp-library* module and calls the CSP API via the generated Java bindings. It does this by the use of the *CspTerminal* class, as shown in Figure 7.

The *CspTerminal* class acts as a CSP node, with dedicated handlers for CSP services and a routing configuration. It can be triggered to call functions from the CSP API via the *CspActivityProcessingService*, which is an Activity Processor implementing an activity defined in the TDM. The error code returned by the implementation in the CSP libraries is then mapped back to the state of the activity by the use of EGS-CC's *IActivityReporter* interface. The current prototype implementation only

[3]EGS-CC is implemented as a distributed OSGi application in the Apache Karaf runtime (*https://karaf.apache.org/*)

allows the processing of String-based inputs with the *sendCommand* function, similar to the text-based commands that can be executed in CSP-Term tool (see Figure 3). However, the functionality in the CSP API could be exposed on a more granular level in future work, which would enable the definition of a more sophisticated TDM, with multiple activities to perform different commands, to which various types of arguments can be passed.

The final module in the CSP component is *csp-feature*. It used to deploy the compiled *csp-terminal* and *csp-library* bundles into the EGS-CC runtime as a Karaf feature.

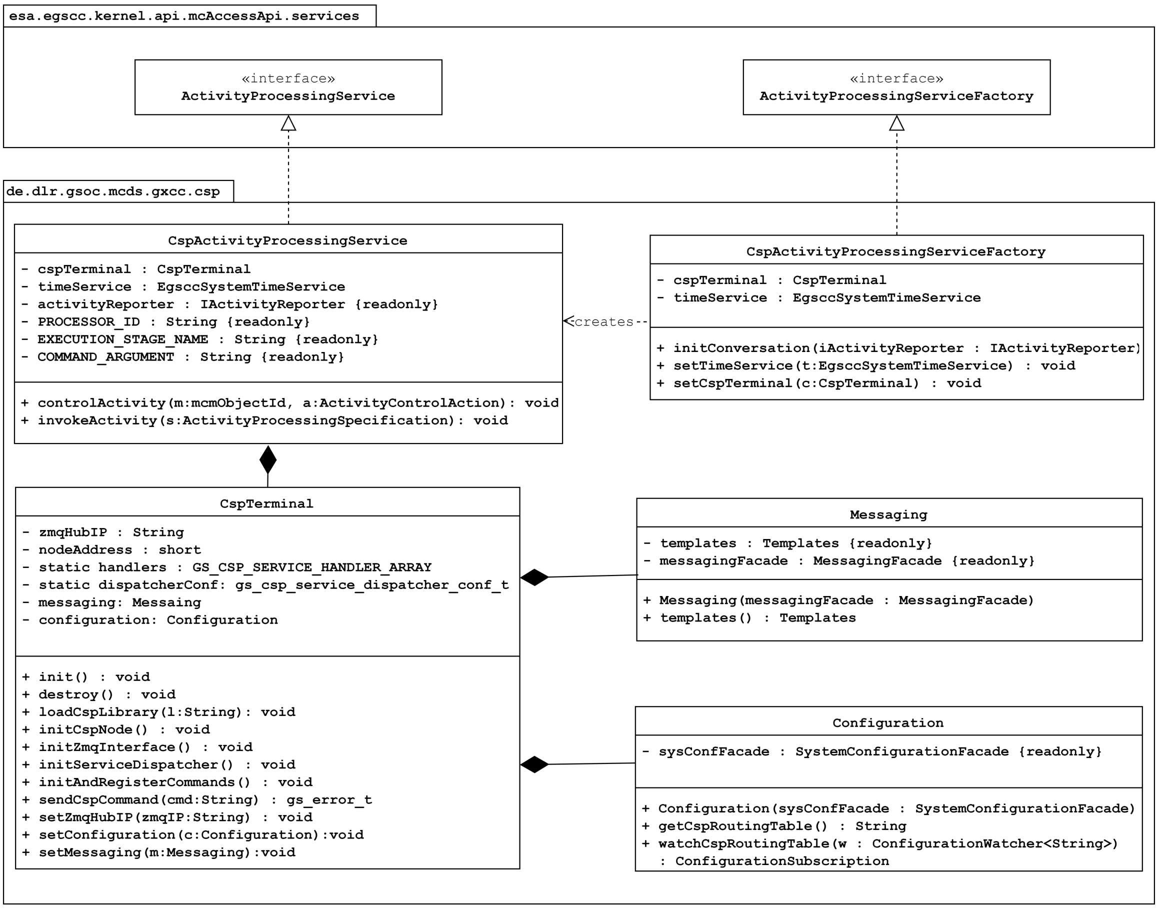


Figure 7. Implementation of the *csp-terminal* Module (UML class diagram)

4 CURRENT STATUS AND OUTLOOK

This section describes the current status of the EGS-CC based approach for integrating small satellites into the GSOC multi-mission infrastructure and provides an outlook for ongoing activities and future work.

4.1 COMMANDING CUBEL WITH EGS-CC

A prototypical EGS-CC based system with a CSP based RI, as described in Section 3, has been deployed in the GSOC multi-mission environment. Using this system, GSOC was able to successfully command the CubeL spacecraft in an experimental pass performed in January 2024. During the pass, pings were sent to the on-board computer and TM was requested from on-board subsystems. Figure 8 shows the successful execution of an activity in the EGS-CC user interface, which triggered the downlink of house keeping data from CubeL.

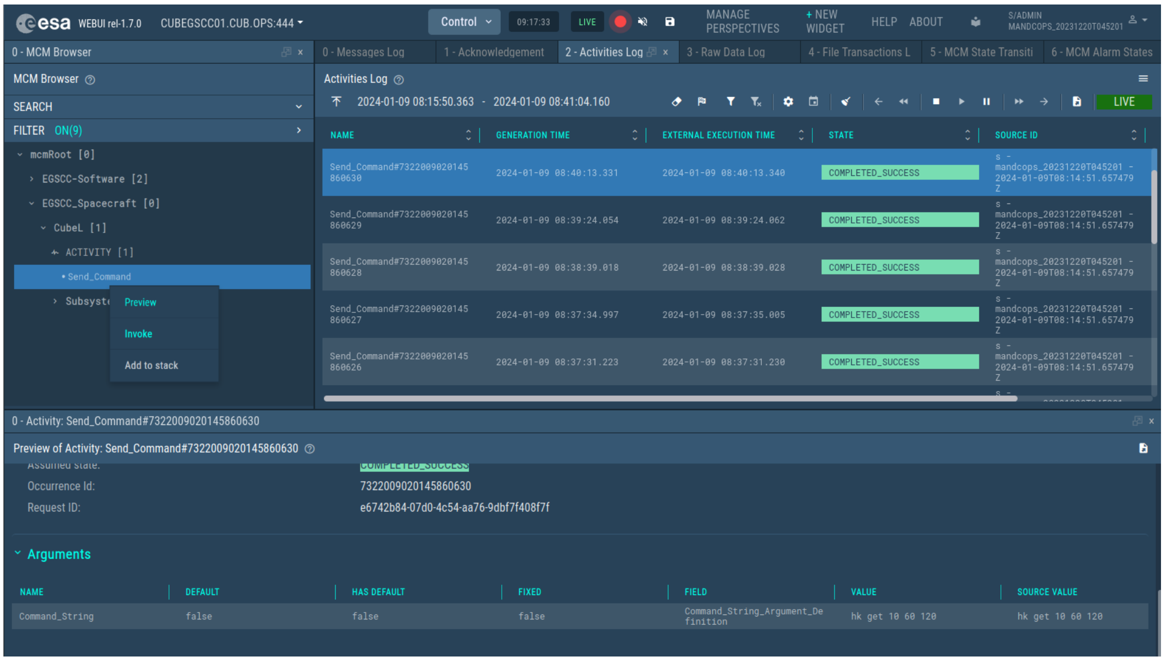


Figure 8. Activities Log in the EGS-CC User Interface

4.2 CONTINUED DEVELOPMENT OF THE CSP RI

With future CubeSat missions in the pipeline, GSOC has an interest in advancing the initiative to integrate small satellites with EGS-CC. Ongoing development efforts are focussed at improving and expanding the functionality of the CSP RI on the basis of the CubeL mission. This includes exposing the CSP API provided in the manufacturer's SDK on a more granular level and defining the structure of the house keeping packets in CSP format in the TDM.

The continued active development also allows GSOC to evaluate the approach to support small satellites in general. This includes evaluating technical feasibility, as well as shaping requirements for small satellite manufacturers.

4.3 GENERAL SOLUTION FOR MULTI-MISSION SUPPORT

In line with GSOC's multi-mission strategy, a future scenario could see the use of EGS-CC to support various types of missions, based on different missions designs and protocols. Figure 9 shows how the use of different RIs could enable the parallel operation of CCSDS/ECSS based missions, CSP based missions, and other missions with custom protocols, all while using the same EGS-CC kernel. If required, a custom RI could be defined in collaboration in with the manufacturer during the design phase of a mission.

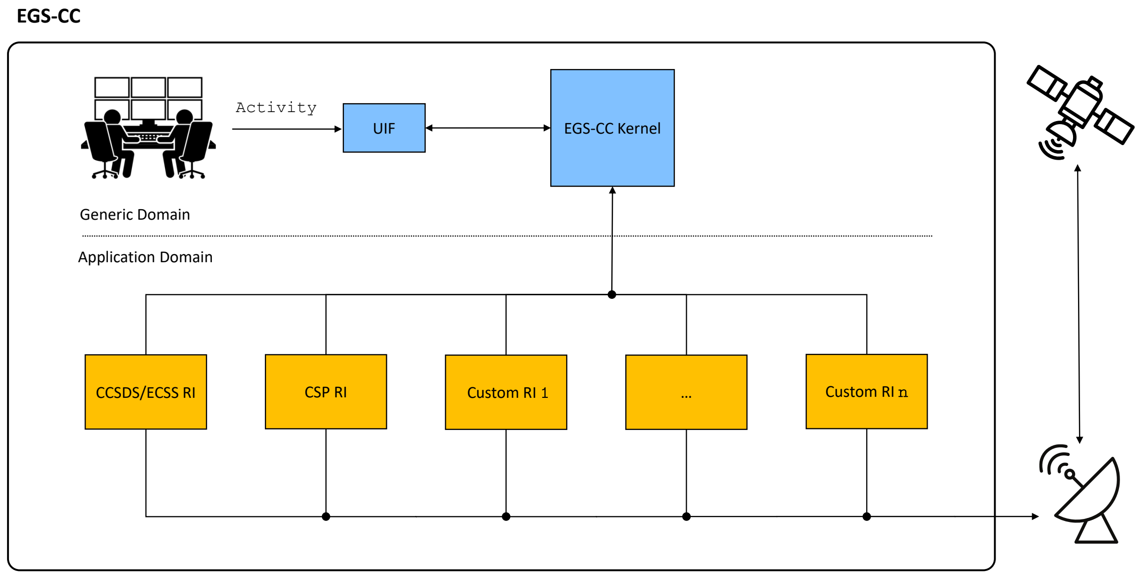


Figure 9. EGS-CC with multiple Reference Implementations

4.4 INTERFACE TO THE GROUND STATION

To handle new designs and protocols, an important question that needs to be solved is the interface between the MCS and the ground station. The traditional approach, based on an SLE connection from the operations center to the ground station, heavily relies on CCSDS standards for the TC Space Data Link (CCSDS 232.0-B-4) and TM Space Data Link (CCSDS 132.0-B-3). In addition, hardware like the CORTEX, which is used by currently compatible ground station for modulation and decoding, is based on CCSDS standards for Radio Frequency and Modulation Systems (CCSDS 401.0-B-32) and TM/TC Synchronization and Channel Coding (CCSDS 131.0-B-4/CCSDS 231.0-B-4). Non-compliance to these standards impacts the SLE protocol and therefore the interface between MCS and ground station.

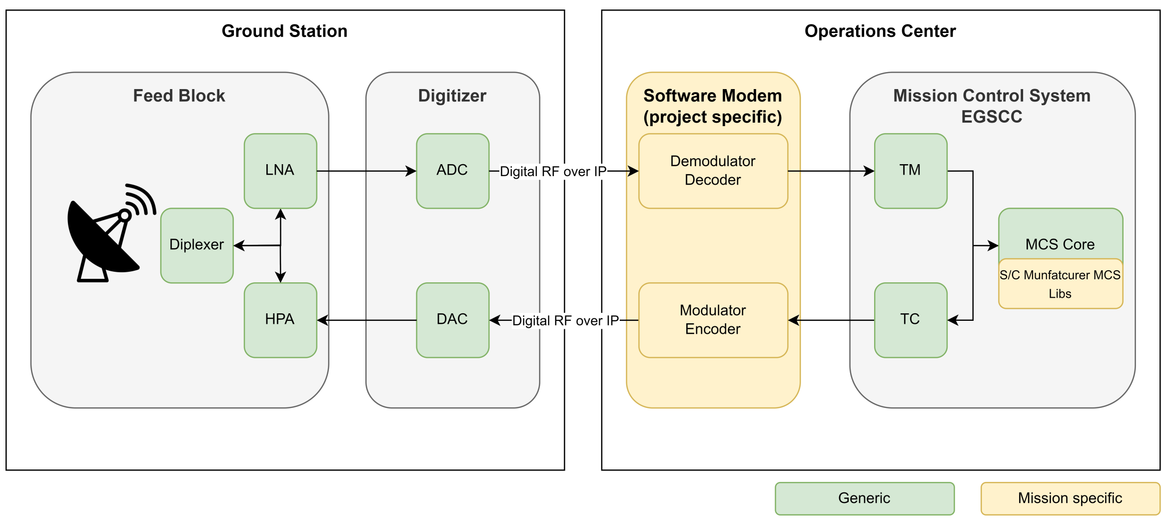


Figure 10. Software Modem interface to the Ground Station

For communications not complying to these standards, as for example in a mission based on CSP, the RF signal could be routed directly to a project-specific software modem located at the operations center. This modem could take care of modulation and coding and provides flexibility to deal with special use cases like Doppler compensation or custom idle sequences. This design is shown in Figure 10.

5 CONCLUSION

This paper proposed a new concept for integrating small satellite missions into the GSOC multi-mission infrastructure, which is based on EGS-CC. An overview of the current multi-mission strategy and the core tools used at GSOC was given. The CubeL mission has shown how problems can arise when the traditional approach to operations at GSOC is challenged by New Space designs typically found in small satellites. It was demonstrated how these types of problems can be solved by integrating the alternative design philosophy through a reference implementation in EGS-CC.

This concept for integrating a small satellite mission marks a departure from the traditional approach at GSOC, where a manufacturer is expected to be compliant to CCSDS/ECSS and provide certain deliverables like the MIB. Using EGS-CC could allow the integration of different types of protocols and designs into the multi-mission infrastructure, which shifts the requirements placed on a manufacturer.

For example, a future scenario could see a manufacturer being expected to provide an API that can be exposed as EGS-CC activities. GSOC could also potentially consider to specify additional supported protocols next to CCSDS/ECSS. This could lower technological barriers for small satellite manufacturers to be compliant with the multi-mission infrastructure, but at the same time maintain a certain level of standardization.

It becomes clear that there could be an interest in finding common ground. Customers of small satellite manufacturers may be interested in lowering launch costs, but at the same time desire the services of a control center, such as around-the-clock, high-reliability operations. Small satellite manufacturers may be looking to appeal to larger agencies by providing more standardized features and services. Control centers like GSOC may have an interest in opening up to the growing market of small satellites. Certainly, finding a balance between traditional mission operations and New Space concepts will require making tradeoffs. Nevertheless, it could be possible to capitalize on the strengths of both domains.

REFERENCES

[1] *"German Space Operations Center"*, Accessed: 26.03.2024. [Online]. Available: https://www.dlr.de/en/research-and-transfer/projects-and-missions/iss/the-german-space-operations-center.

[2] B. Eilertsen, M. Krynitz, and K. Olafsson, *Newspace - forcing a rethink of ground networks*, in *SpaceOps 2016 Conference*. DOI: 10.2514/6.2016-2599. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2016-2599.

[3] C. Stangl, B. Lotko, M. P. Geyer, M. Oswald, and A. Braun, *GECCOS - The New Monitoring and Control System at DLR-GSOC for Space Operations, based on SCOS-2000*, in *13th International Conference on Space Operations (SpaceOps 2014)*, American Institute for Aeronautics and Astronautics (AIAA), 2014, ISBN: 9781634398343.

[4] A. Johnstone, *CubeSat Design Specification Rev 14.1*. Cal Poly SLO, 2022, page 7ff.

[5] *"SatNOGS - Open-Source Global Network of Satellite Ground Operations"*, Accessed: 28.03.2024. [Online]. Available: https://satnogs.org/about.

[6] *"CubeSat Space Protocol"*, Accessed: 28.03.2024. [Online]. Available: https://github.com/libcsp/libcsp.

[7] T. Brügge, P.-A. Lagadrilliere, A. Geda, and A. Balan, *TT&C over S-Band with CubeL: Finding a Middle Way Between CSP, CCSDS and ECSS*, in *SpaceOps 2023 Proceedings*, Mohammed Bin Rashid Space Centre, 2023.

[8] *"EGS-CC - The European Infrastructure for Monitoring and Control of Space Systems"*, Accessed: 02.04.2024. [Online]. Available: http://www.egscc.esa.int/index.html.

LIST OF ABBREVIATIONS

AP	Activity Processor
API	Application Programming Interface
CCSDS	Consultative Committee for Space Data Systems
CSP	CubeSat Space Protocol
ECSS	European Cooperation for Space Standardization
EGS-CC	European Ground System - Common Core
G/S	Ground Station
GECCOS	GSOC Enhanced Command and Control System for Operating Spacecrafts
GSOC	German Space Operations Center
HK	House Keeping
ISS	International Space Station
JNI	Java Native Interface
LEOP	Launch and Early Orbit Phase
M&C	Monitoring and Control
MCE	Monitoring and Control Element
MCM	Monitoring and Control Model
MCS	Mission Control System
MIB	Mission Information Base
MUM	Multi-Mission
NCTRS	Network Control and Telemetry Routing System
ProTos	Procedural Tool Suite
PUS	Packet Utilization Standard
RF	Radio Frequency
RI	Reference Implementation
SatNOGS	Satellite Networked Open Ground Station
SCOS-2000	Satellite Control and Operation System 2000
SDK	Software Development Kit
SLE	Space Link Extension
SSB	SLE Switch Board
TC	Telecommand
TDM	Tailoring Data Model
TM	Telemetry
UHF	Ultra High Frequency
UI	User Interface
ZMQ	ZeroMQ Messaging Framework

LIST OF FIGURES