CONVERGE
view-to-communicate and communicate-to-view

CONVERGE - Telecommunications and Computer Vision
Convergence Tools for Research Infrastructures

D1.2: Specification of interfaces and access policies (initial)

Work package	WP1	**Task**	Task 1.3 and Task 1.4
Due date	31/10/2023	**Submission date**	31/10/2023
Editor	Paulo Marques (ALLBESMART LDA)		
Internal Reviewers	Walid Dabbous (INRIA), Filipe Conceição (INTERDIGITAL), Filipe B. Teixeira (INESC TEC)		
Version	1.0		
List of contributors (Alphabetically ordered, grouped by entity)	Diogo Lopes (Adapttech), Frederico Carpinteiro (Adapttech), Miguel Neves (Adapttech), Sérgio Silva (Adapttech), Luis Pereira (ALLBESMART LDA), Paulo Marques (ALLBESMART LDA), Tiago Alves (ALLBESMART LDA), Jorge García Vidal (BSC), Maria-Cristina Marinescu (BSC CNS), Victor-Alejandro Ortiz (BSC CNS), Dirk Slock (EURECOM), Raymond Knopp (EURECOM), Juha Kela (Finwe Oy), Ahmad Shokair (GREENERWAVE), Youssef Nasser (GREENERWAVE), André Coelho (INESC TEC), Filipe B. Teixeira (INESC TEC), Helder Fontes (INESC TEC), Luís Pessoa (INESC TEC), Luís F. Teixeira (INESC TEC), Manuel Ricardo (INESC TEC), Damien Saucez (INRIA), Manel Khelifi (INRIA), Walid Dabbous (INRIA), Filipe Conceição (INTERDIGITAL), Okan Yurduseven (QUB), María García-Fernández (QUB), Guillermo Álvarez-Narciandi (QUB), Mondeep Saikia (QUB), Serge Fdida (SORBONNE), Antti Pauanne (UOULU), Miguel Bordallo Lopez (UOULU), Niklas Vaara (UOULU), Pekka Sangi (UOULU)		
Dissemination Level	PU ***PU:*** *Public, fully open, e.g., web (Deliverables flagged as public will be automatically published in CORDIS project's page)*		

	SEN: *Sensitive, limited under the conditions of the Grant Agreement*

Keywords	Access policies, Architecture, Interfaces

Co-funded by the European Union

The CONVERGE project has received funding under the European Union’s Horizon Europe research and innovation programme under Grant Agreement No 101094831, including top-up funding by UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee.

CHANGE REGISTER

Version	Date	Editor	Organization	Changes
A	24.04.2023	Paulo Marques	ALLBESMART LDA	Table of contents proposal
B	24.07.2023	Paulo Marques	ALLBESMART LDA	Table of contents adjustments
C	17.10.2023	Paulo Marques	ALLBESMART LDA	Review of contributions
D	27.10.2023	Walid Dabbous Filipe Conceição	INRIA Interdigital	Internal review
E	31.10.2023	Filipe B. Teixeira	INESC TEC	Quality assurance review
F	31.10.2023	Luís Pessoa	INESC TEC	Document formatting and minor corrections

EXECUTIVE SUMMARY

The main objective of the CONVERGE project is the development of an innovative toolset combining radio and vision-based communications and sensing technologies to enable an emerging area of research aligned with the motto "view-to-communicate and communicate-to-view", and through the integration of this toolset, advance the state of the art of a set of Research Infrastructures (RIs), to the greatest extent aligned with the ESFRI SLICES-RI.

This deliverable reports the first iteration in the definition of access modes and policies for the tools developed by the CONVERGE project. These access modes and policies adhere to the best practices used in established RIs, namely those considered in the ESFRI SLICES-RI. The SLICES-RI is therefore used as a model to guide our discussion of access types, access modes, and access policies for the CONVERGE tools.

The document also presents in detail the CONVERGE service-oriented architecture (its preliminary version was presented in D1.1 – Requirements and use cases) which has been devised to be aligned with a 5G network architecture. An example of a CONVERGE use case test session is then provided along with a step-by-step flowchart and instructions for starting up and configuring the CONVERGE chamber. The remainder of the interfaces, including the set of twelve CONVERGE Application Programming Interfaces (APIs), is then presented with the associated attributes and parameters, following the 3GPP design principles and guidelines.

This deliverable corresponds to the first version of the specification of access policies and interfaces. A final version will be delivered by M18 (to be referenced as D1.3), where it is expected that both the access policies as well as the CONVERGE interfaces will be described with more detail, allowing for the beginning of the implementation planned within WP3.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5GTN	5G Test Network
C2V	Communicate to View
CN	Core Network
CV	Computer Vision
CU	O-RAN Central Unit
DU	O-RAN Distributed Unit
E2E	End to End connectivity
ESFRI	European Strategy Forum on Research Infrastructures
FAIR	Findable, Accessible, Interoperable, Reusable
FH	Front Haul
FR1/2	Frequency Range 1/2
LIS	Large Intelligent Surface
MAC	Medium Access Control
ML	Machine Learning
NFV	Network Functions Virtualization
O-RAN	Open Radio Access Network
OAI	Open Air Interface
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
RIC	RAN Intelligent Controller
RI	Research Infrastructure
RIS	Reconfigurable Intelligent Surface
SDR	Software Defined Radio
SLICES-RI	Scientific Large-scale Infrastructure for Computing/Communication Experimental Studies – Research Infrastructure
UE	User Equipment
V2C	View to Communicate

1 INTRODUCTION

The main objective of the CONVERGE project is the development of an innovative toolset combining radio and vision-based communications and sensing technologies to enable an emerging area of research aligned with the motto "view-to-communicate and communicate-to-view", and through the integration of this toolset, advance the state of the art of a set of research infrastructures to the greatest extent aligned with the ESFRI (European Strategy Forum on Research Infrastructures) SLICES-RI (Scientific Large-scale Infrastructure for Computing / Communication Experimental Studies - Research Infrastructures).

The toolset to be developed consists of (1) large vision-aided reconfigurable intelligent surface (RIS), in this context named as Large Intelligent Surface (LIS) enabling experimentation of massive MIMO wireless communications and 3D positioning and environment sensing, (2) vision-aided fixed and mobile 5G OAI base station (BS) enabling communications and experimentation with User Equipment (UE)/mobile terminals, (3) a vision-radio simulator and 3D environment modeller to create a digital twin allowing for experiment planning, as well as machine learning algorithms for processing multimodal data, including communications signals, video streams, as well as radio frequency sensing and network traffic traces. This toolset will be installed in experimental chambers and used by experimenters (accessing it remotely or locally) to pre-configure the positions and mobility patterns of mobile base stations, obstacles, and user equipment, configure test sessions and run experiments, potentially involving humans, to gather vision-radio datasets.

The usage of the CONVERGE toolset shall be done via specific interfaces for each tool. Additionally, a user interface is planned to be the main application enabling a researcher to access and interact with the toolset. It will allow the user to access and configure individual tools, not only the physical tools located in the CONVERGE chamber but also the virtual tools, namely the 3D environment modeller and vision-radio simulator as well as the ML algorithms for dataset analysis, and will further include configuration and trace collection functions. The configuration functions will allow setting-up the toolset and the test sessions running on it; configuration may involve individual tools or a set of tools cooperating in a given experiment or simulation. The trace collection functions will enable the synchronised and space-referenced collection of traces and results from the tools which, in CONVERGE, are varied and include visual or RF-based data streams from a multitude of cameras and RF sensing, signal statistics such as signal to noise ratios, positioning of equipment, or traffic indicators such as throughputs and frame delivery ratios. The trace collection functions consist of three main components:

- Software probes, executed in the base stations, user equipment and large intelligent surface to gather specific time-stamped and space referenced traces and videos. These probes will act as trace/video publishers to a data broker associated with a high-performance database.
- High-performance data broker and database, executed at the premises of the experimental chamber or in the cloud. This data broker receives the timestamped information, stores it, and manages the access to the traces, including real-time subscription of the traces or the download of past traces.
- High-precision time synchronisation system, executed at each tool with trace generation capability, ensuring that traces are correctly timestamped prior to their publishing. High-precision time synchronisation is important for, not only, synchronising multiple traces originating from the same node, but also when using traces regarding the cooperative perception of multiple nodes.

This deliverable D1.2 reports the work made during the initial phase of Task T1.3: "Tool Access Modes and Policies" and Task T1.4: "User and Tool Interface Specification". The focus of T1.3 is the establishment of the CONVERGE baseline access policies for the tools developed within the project. Based on the example of the SLICES-RI, the aspects related to access types, access modes, and access policies for the CONVERGE tools are discussed. The focus of T1.4 is on the design of a CONVERGE service-oriented architecture, which is inspired in the 3GPP 5G core network architecture, accompanied by its corresponding open APIs. The further developments of these tasks, after the delivery of the present document D1.2, will be documented in D1.3, which is scheduled for delivery in M18.

2 CONVERGE TOOLS ACCESS MODES AND POLICIES

The access modes and policies are instrumental in allowing experimenters to have an impact using the tools developed by CONVERGE. As we have defined since setting up the CONVERGE project: (i) the access should follow the best practices in established RIs, such as those used in ESFRI, following the FAIR (Findable, Accessible, Interoperable, Reusable) principles, and taking into account the specific access policies of the RIs involved in the project, and (ii) our task should study notably the opportunity to propose the different types of access used by the SLICES-RI.

The SLICES-RI is in the Preparation phase of the ESFRI lifecycle, with work dedicated to defining the SLICES user needs, services, access and training strategy. CONVERGE will benefit from regular updates from the SLICES definitions. SLICES has already defined, during its design phase, the baseline for its services and access modes and types, based on the analysis of the user requirements and three examples of usage of the platform in the field of cloud/edge, IoT, and wireless/wired networking. On this basis, SLICES has defined its access modes through the three types of offered interfaces, and with the cases of the usage of some of its services in four basic and typical examples: a basic experiment, an orchestrated experiment, a basic data-intensive experiment, and a basic course.

Based on the example of the SLICES-RI, CONVERGE tools must include in their specifications (as part of final WP1 deliverable D1.3) the aspects related to the access: 1) Access types, 2) Access modes, and 3) Access policy, as described below.

Access types

The tools developed by CONVERGE have to be compatible with the three types of access that will be enabled through SLICES, in compliance with the access methods defined in the ESFRI 2020 white paper for "Making Science Happen - A new ambition for Research Infrastructures in the European Research Area" [1] as follows:

- Trans-national access (Physical access): Apart from accessing the equipment, the facilities and the laboratories, users will also be offered the required technical and scientific assistance to learn and use the infrastructure. Though such a type of access is more applicable to RIs that cannot be easily accessed remotely for fine-tuning, this is envisioned to facilitate experiments requiring physical presence at one of the SLICES sites, such as for instance Bring-Your-Own-Equipment (BYOE). In such experiments, researchers can install their prototype equipment at one of the SLICES sites and combine it in their experiments with the equipment offered by SLICES;

- Trans-national virtual access (Remote access): The majority of the tools and services that are designed and shall be developed for SLICES deal with the remote access method, in order to present to the users of the platform a unified solution for retrieving, selecting, reserving their experimental components and deploying their experiment on top of them. Through remote access, the users of the platform will take advantage of the tools for controlling their experiment and the environment parameters in an organized manner, under real world settings. This type of access is envisioned as the preferred method of access for the vast majority of SLICES users;

- Virtual access: Virtual access typically concerns access to data and digital tools. The provision of Virtual access to SLICES is aided through sophisticated cloud services and communication networks and allows for the remote access to repositories and archives of produced experimentation results. The vision of SLICES for allowing virtual access is to offer a pan-European operational networking and computer infrastructure to facilitate scientific research with instrumentation and experimentation capabilities.

Access modes

The CONVERGE tools will be in the same context for access modes as the SLICES-RI: mostly accessed via the Internet in a remote fashion (remote access and virtual access modes) by users. Like in SLICES, services will be computer provided and accessible through a dedicated web portal, including dedicated APIs (through the planned REST software architecture), and, also in order to support experimentation by users from research domains different from future Internet and distributed systems, high levels tools such as Jupyter notebooks and workflow systems. Additionally, the CONVERGE tools should meet also the three access modes described in the document "European Charter for Access to Research Infrastructure" [2] of the European Commission:

- Excellence-driven;
- Market-driven;
- Wide.

Taking the example of how SLICES-RI envisage these three types of access: large academic experiments are expected to access the platform under the "excellence-driven" mode while smaller scale experiments will benefit from the "wide" mode. The "wide" access mode guarantees the broadest possible access to scientific data and digital services provided by the Research Infrastructure to users wherever they are based. Business and industry are mainly expected to access the platform under the "market-driven" mode. Considering its user community, an initial estimation of SLICES is that "excellence-driven" mode will represent 60% of the time platform, "wide" 20%, and "marked-driven" 20%, which seems to be also a good first estimation for the CONVERGE tools. SLICES has planned that these numbers will be tightly monitored and communicated to the SLICES Supervisory Board. So the CONVERGE project Steering Committee will be regularly informed of the updates for these numbers. Similarly, for SLICES, the costs to access the platform for the "market-driven" mode are yet to be discussed and decided. The two other modes, "excellence-driven" and "wide", are, in the case of SLICES, open and free of charge for European academics. If needed, calls for proposals—such as those organized by PRACE for example— will be organized to allocate resources based on best usages by the SLICES Access Committee. Considering that CONVERGE tools are expected to fit into nodes of SLICES, the same adopted cost rules should be followed.

It should be noted that from a user point of view, accessing SLICES will share some similarities to accessing existing digital infrastructure testbeds, although SLICES will offer a larger scale and more diverse infrastructure as well as an advanced monitored and user-controllable infrastructure for researchers in wireless/wired networking, IoT, Cloud, and distributed systems communities. The CONVERGE tools have to take these elements into consideration in order to fit to SLICES and to other ESFRI RIs.

Another point to be taken into account for the development of the CONVERGE tools is related with the data produced through the use of the tools. As SLICES will be registered into the European Open Science Cloud (EOSC) catalogue, the data produced and offered through SLICES will be annotated appropriately, while SLICES will provide EOSC compatible mechanisms to easily access the platform (authentication and authorization) and to manipulate data (EOSC FAIR mechanisms). The CONVERGE tools should allow and facilitate the work to be done to meet these requirements.

Access policy

The access policy of any European research infrastructure should be built according to the "European Charter for Access to Research Infrastructure" [2]. According to this Charter, "Research Infrastructures should have a policy defining how they regulate, grant and support Access to (potential) Users from academia, business, industry and public services".

SLICES-RI has not yet built its access policy. It will definitely be built based on this Charter and by benefiting from the expertise developed by the partners whilst operating research infrastructures.

SLICES has already stated that the policy will follow two principles: (i) Excellence-driven access through the Access Committee selection, including a market-driven access for Industry, (ii) Global approach and privileged access regardless of their location for all EU research groups, also open to other institutions worldwide. SLICES also plans to open access for education and training purposes by requesting special permissions to the Access committee.

The access to the CONVERGE tools will adhere to the SLICES policy, which envisages that:

- The elements for the implementation of its access policy will be:
    - The access unit;
    - The monitoring and accounting (users, resources, data);
    - The ease of use through a RESTful API to control and interact with the infrastructure management system;
    - A clear and didactic documentation (including tutorials, hands-on, etc.).
- The envisioned access unit would be a slice, i.e., a time bounded and exclusive reservation of a set of resources. According to the resource kind (from an IoT device to a complete network/data centre, or a set or subset of tools in the specific case of CONVERGE), the duration may vary up from hours to several days;
- As technology allows a lot of flexibility regarding access, SLICES will also implement a Policy-based access that could embed constraints such as privacy concerns or industry usage. For example, in the case of a Market access by industry, a policy can keep the data or configuration private. Likewise, in the situation when the experiment manipulates data that can affect the privacy of an individual, specific policies can be enforced;
- The monitoring and accounting will be done on a per user basis and consolidated to capture research group, institutions usages or simply for charging costs. Such metrics are equally important to drive the evolution of the testbed;
- A one-stop shop solution will be provided through a single portal to request the identification of a user only once. This is mandatory, as users need to be traced back and identified in case of issues. A user first line support will also be developed to assist the experimenters in understanding the facility and developing their tests;
- Two levels of support will be provided, one central will be generic and provide the single point of presence of SLICES with its community of experimenters. It will be mostly suited to handle general questions, tutorials and basic access. A second line is considered with more advanced support if need be and will be shaped according to the main technological solutions deployed, namely cloud/compute, wireless or IoT, and including the CONVERGE tools;
- Tutorials, hands-on will be made available on-line and proposed during major conference events as well as dedicated SLICES schools for young researchers;
- As digital is a fast-moving field, the access policy is subject to evolution to adapt to new needs and usages. The tracking of needs and the adaptation of the policy will be managed by the SLICES management with the help of its user committee.

CONVERGE integration into SLICES Web Portal

The CONVERGE complete toolset comprised by the four tools: 1) vision-aided large intelligent surface, 2) vision-aided base stations and user equipment, 3) vision-radio simulator and 3D environment modeler and 4) machine learning algorithms will be built into the University of Oulu laboratory as part of the existing 5G Test Network (5GTN) RI as part of the work planned under WP3, namely T3.5 “Integration and demonstration of the complete toolset”. University of Oulu is currently creating the environment to achieve full integration of 5GTN into SLICES-SC, as part of the ongoing SLICES-SC project. The goal of this integration is to connect the 5GTN to the SLICES-SC Web Portal to enable access and testing through it, namely to enable the access to all 5GTN tech areas including 5G core, virtual compute nodes, data storage, edge servers and also base Stations. When full integration of 5GTN to SLICES-SC is achieved, the SLICES-SC Web Portal will be available to be used by CONVERGE project partners and possible external users outside of the project to access the CONVERGE test environment remotely. SLICES-SC web portal is to be used to identify the internal users and to know the identity of the external users and grant access to the 5GTN test infrastructure (including CONVERGE tools). After the access to SLICES-SC Web Portal is granted, the users need to create a test project within the Portal. Inside the project users are then able to create their own test set-up based on the resources like compute nodes, data storage and base stations that are made available by the 5GTN RI. It must be considered that although the creation of the test setup can be done remotely, in some cases not all of the desired tasks can be done remotely since the actual testing, depending on the test setup, often requires also manual work by persons present at the test site. For example, the UEs normally need to be operated by a person. Also, the test configuration involving available test equipment and software usually needs manual configuration. Nevertheless, the possibility for remote use opens new possibilities for the CONVERGE project to plan the tests and test setups remotely and at the same time open the test platform for possible external use also. While the final Web Portal to be used by SLICES-RI in the long term is still under specification as part of SLICES preparatory phase within the ESFRI lifecycle, the experience gained with the 5GTN integration into the SLICES-SC portal will be important for steering the CONVERGE developments and ensuring a successful integration of CONVERGE into SLICES-RI in the long term.

3 CONVERGE SERVICE-ORIENTED ARCHITECTURE

High-level architecture

The CONVERGE high-level architecture is shown in Figure 1 and consists of 3 main building blocks: the CONVERGE Chamber, the CONVERGE Core, and the CONVERGE Simulator. The rationale for this division stems from the fact that CONVERGE features both 1) a physical infrastructure (the CONVERGE Chamber) where physical experiments using a combination of gNB, UE and LIS equipment will be possible, including physical mobility of the equipment and 2) a simulator infrastructure, where a digital twin of the Chamber can be used for planning experiments to be carried out later in the CONVERGE Chamber, or for testing/extrapolating scenarios which are difficult or impossible to test experimentally. Additionally, a third block is necessary for interfacing with the user and controlling the physical or virtual experiments, for storing/retrieving datasets and ML models, as well as enabling the training of ML models, the CONVERGE Core.

The CONVERGE research infrastructure will be accessible by external experimenters for the realisation of real time (physical) experiments within the chamber or through off-line/virtual mode for the access to the simulator component, ML tools and datasets. The different access modes of the CONVERGE Chamber have implications on the APIs to be used during the test session, which are addressed in the next sections where the different APIs are described.

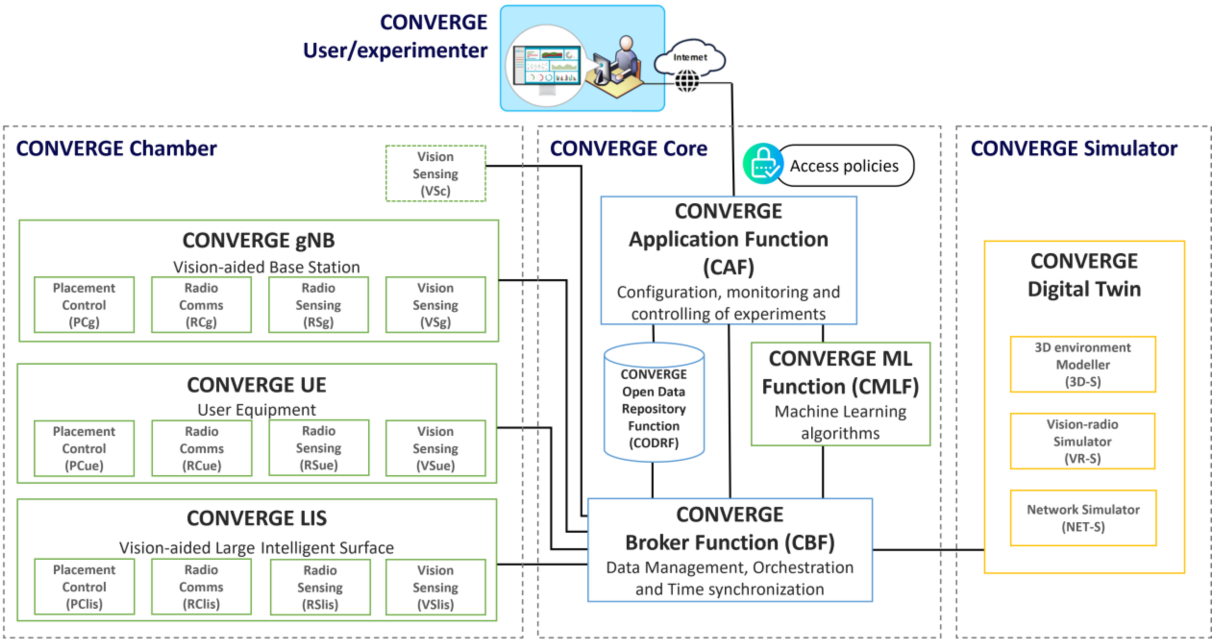

Figure 1 – CONVERGE high-level architecture.

CONVERGE Chamber

The CONVERGE Chamber represents the experimental chamber to be developed in the project, which will be deployed in different research infrastructure sites. In the chamber, there are three types of equipment: the CONVERGE gNB, the CONVERGE UE, and the CONVERGE LIS. The CONVERGE gNB will have functionality for controlling the placement of gNB (PCg in Figure 1), radio communications functionality (RCg) defined for 5G gNB, functionality for radio sensing of gNB (RSg), and functionality for vision sensing of gNB (VSg). The CONVERGE UE will have functionality for controlling the placement of UE (PCue), radio communications functionality of UE (RCue) as defined for 5G, functionality for radio sensing of UE (RSue), and functionality for vision sensing of UE (VSue). The CONVERGE LIS will have functionality for controlling the placement of LIS (PClis), radio communications functionality typical of a LIS (RClis), functionality for radio sensing (RSlis), and functionality for vision sensing by taking advantage of video cameras installed in the LIS (VSlis).

In addition, the CONVERGE architecture can optionally have a functionality for video sensing of the experiments taking place inside the Chamber (VSc). The CONVERGE Chamber equipment will be connected and controlled by the CONVERGE Core building block.

To support remote user access for acquisition, visualization, and manipulation of radio and image data, specific hardware platforms will be necessary. Existing FPGA-based Systems-on-Chip are good candidate platforms. Specifically, AMD’s RF-SoC family provides a plethora of RF and other interfaces under a Debian-like operating system that can be accessed remotely. The OS is hosted by the processor side, which can include, among other things, Python environments. This allows for control and data visualization via Jupyter servers [4], and also the use of well-known Machine Learning libraries (e.g., PyTorch), providing a low barrier to the use of the chamber tools, and allowing for researchers to implement ML-based radio/image solutions with real-time data. Furthermore, the reconfigurability aspect of these platforms, afforded by the integrated FPGA, brings an additional layer of experimentation potential. Application-specific hardware designs can be offloaded to the FPGA, also via Python calls, allowing researchers to deploy the computing blocks which comprise their solutions or experiments. Significant productivity and cost benefits can be achieved by alleviating researchers the nontrivial setup effort and by providing access to such platforms which would otherwise be prohibitively expensive to acquire.

CONVERGE Core

The CONVERGE Core building block is responsible for controlling the operation of the equipment located in the chamber and it consists of several functions. The CONVERGE Application Function (CAF) will interface with users/experimenters that, through remote access from the internet and according to specified access policies, enable them to configure, monitor, and control experiments to be carried out in the CONVERGE infrastructure.

The management of a CONVERGE session will be performed by the CONVERGE Broker Function (CBF) that will orchestrate the execution of a CONVERGE session carried out by actual equipment or simulation tools, being also responsible for coordinating the storage of data resulting from a CONVERGE session.

The CONVERGE Open Data Repository Function (CODRF) will store data generated by CONVERGE, including open datasets and machine learning functions.

The CONVERGE Machine Learning Function (CMLF) will enable the access to the functions made available by the machine learning tools developed in the project, including the training of models using the collected datasets; these tools may be used by external users, CONVERGE equipment, or CONVERGE Simulator tools.

CONVERGE Simulator

The CONVERGE Simulator building block consists of three simulation tools: the 3D Environment Modeller (3D-S), the Vision Radio Simulator (VR-S), and the Network Simulator (NET-S). Together, these tools will define the CONVERGE Digital Twin that will enable recreating digitally a physical environment, modelling the radio and vision communications in the environment by using ray tracing-like capabilities, and generating simulated traffic considering the realistic physical channel models obtained by the above mentioned two tools. Sessions in the CONVERGE Simulator are expected to be controlled also by the CONVERGE Broker Function and the simulations scripts and simulation results are stored in the Open Data Repository Function.

Service-oriented architecture

The CONVERGE architecture will be, in fact, implemented as a service-oriented architecture aligned with a 5G network architecture (Figure 2), and therefore based on 3GPP [3]. The main reasons for this 5G alignment include: 1) the need for basic 5G core network functionality required to support the operation of UE and gNB, such as AMF, SMF and UPF, 2) the completeness and versatility of 3GPP 5G New Radio (5G NR) specifications which can be reused, and 3) the possibility of introducing the new functions required to control the CONVERGE Chamber and the CONVERGE Simulator, aligned with standard development.

In the CONVERGE control and sensing plane (cf. Figure 2, CONVERGE Core) the functionality to accommodate traffic servers is added (CONVERGE Traffic Function) as well as servers for receiving the radio sensing and vision information gathered from the CONVERGE equipment (CONVERGE Video Function). Moreover, a set of new CONVERGE functions is added which will enable the simple and direct control of the CONVERGE Chamber equipment and CONVERGE Simulator tools. These control functions (CUECF, CLISCF, CgNBCF, CTCF, CODRF, C3DSCF, CVRSCF, and CNETSCF) will be implemented as webservices (RESTful API) invoked by other functions. The detailed services to be offered by each new CONVERGE function and the new interfaces are presented in Chapter 5.

Figure 2 shows the CONVERGE service-based architecture, followed by the identification of all interfaces and functions.

CONVERGE interfaces and functions

- **3GPP interfaces [3]**
    - N1, N2, N3, N4, N6, Namf, Nnrf, Nsmf
- **CONVERGE internal interfaces**
    - C3ds, Cgnb, Cgnbdp, Clis, Clisdp, Codr, Cts, Cue, Cuedp, Cvrs
- **CONVERGE core interfaces**
    - Ncuef, Ncaf, Nclisf, Ncgnbf, Ncbf, Nctf, Ncmlf, Ncodrf, Ncvf, Nnetsf, Nvrsf, N3dsf
- **CONVERGE functions**

C3DFCF	CONVERGE 3D Simulator Control Function
CVRSCF	CONVERGE Vison Radio Simulator Control Function
CNETSCF	CONVERGE Network Simulator Control Function
CVCF	CONVERGE Video Control Function
CODRF	CONVERGE Open Data Repository Function
CMLF	CONVERGE Machine Learning Functions
CTCF	CONVERGE Traffic Servers Control Function
CBF	CONVERGE Broker, Session orchestration, data management
CgNBCF	CONVERGE dedicated gNB Control Function
CLISCF	CONVERGE dedicated LIS Control Function
CAF	CONVERGE Application Function
CUECF	CONVERGE dedicated UE Control Function

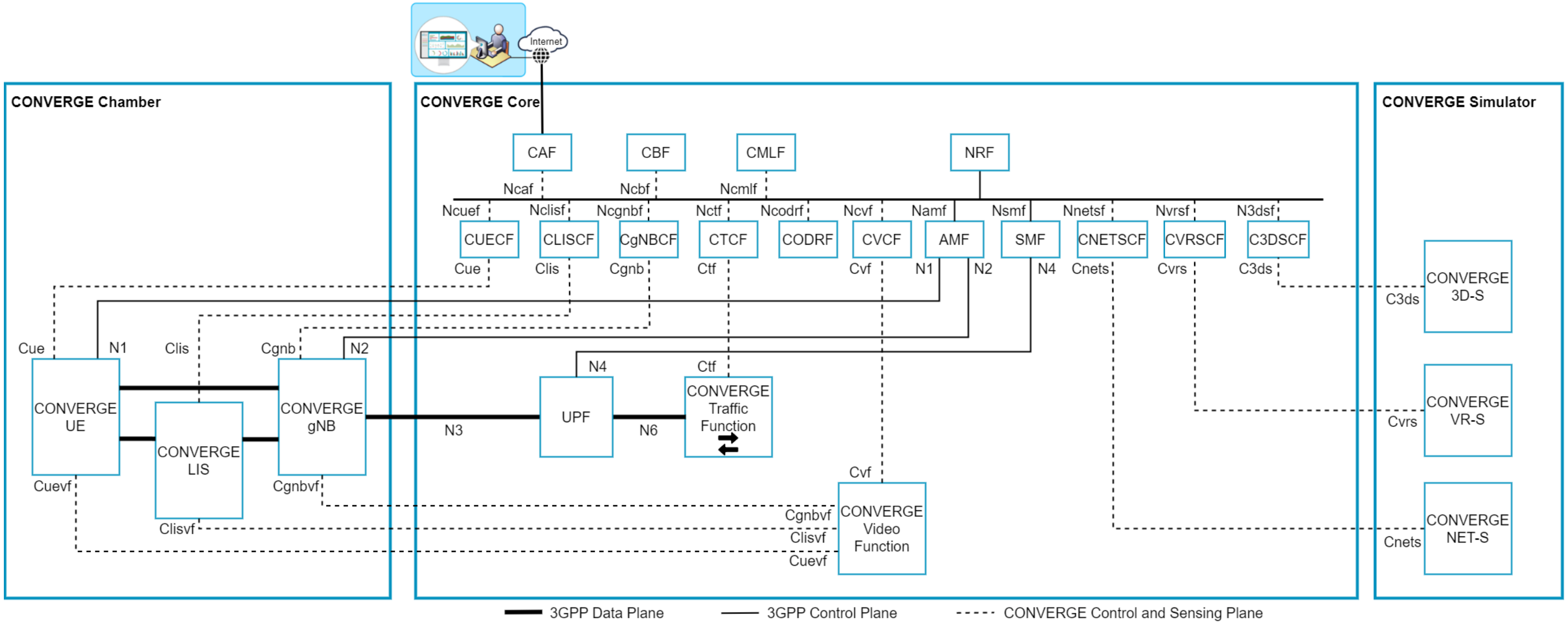

Figure 2 – CONVERGE service-oriented architecture.

CONVERGE vision-aided gNB aligned with O-RAN

Due to the relevance of the vision-aided gNB within the CONVERGE architecture, it is worth explaining in greater detail its internal architecture. The communications functionality of the vision-enabled gNB will be based on the O-RAN architecture [5], including its framework for supporting intelligent controllers, as shown in Figure 3.

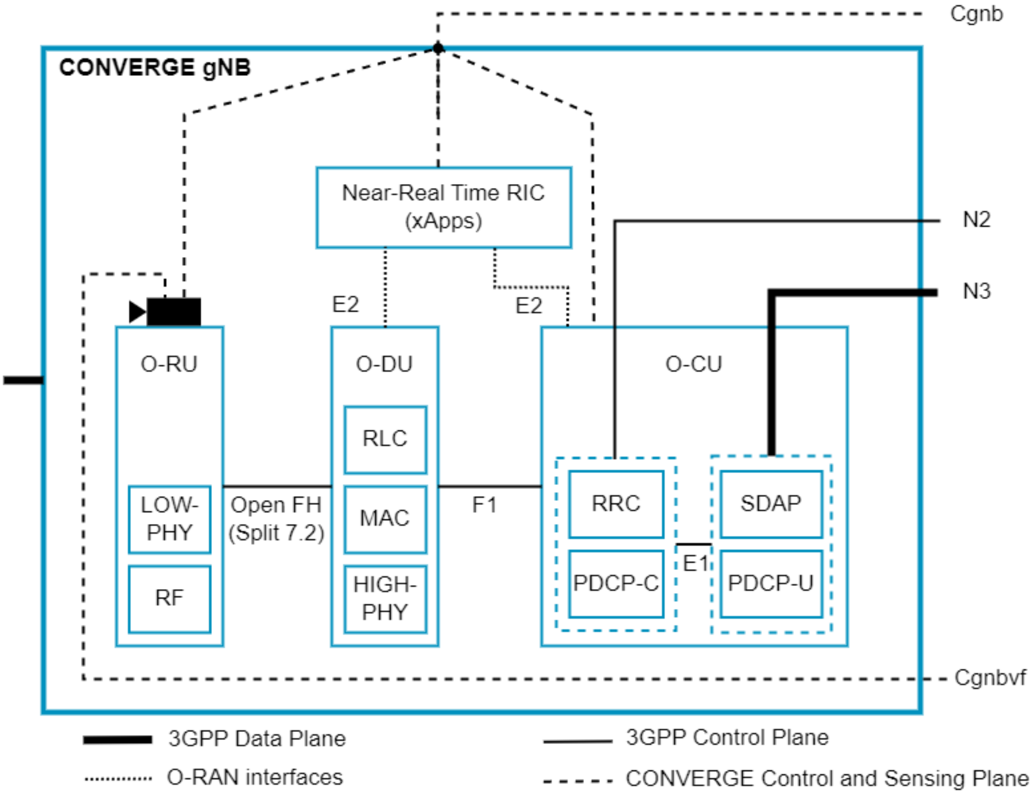

Figure 3 – CONVERGE vision-aided gNB architecture aligned with O-RAN.

O-RAN is an open and standardized architecture for the design and operation of mobile networks, particularly focused on the Radio Access Network (RAN). It aims to promote interoperability and flexibility in the design and deployment of 5G and beyond networks. The key components of O-RAN architecture are:

- RAN Intelligent Controller (RIC): The RIC is a crucial part of the O-RAN architecture. It's responsible for implementing functions that traditionally reside in the baseband unit of the RAN.
- Central Unit (CU): The CU in O-RAN architecture is responsible for functions such as protocol termination, radio resource management, and support for higher-layer applications.
- Distributed Unit (DU): The DU handles functions like the physical layer processing of the radio interface, such as modulation and coding. It interfaces directly with the antennas.
- Open Fronthaul: O-RAN considers the use of standardized and open interfaces for the fronthaul connection between the DU and the CU (designated as Open FH in Figure 3). This allows for multi-vendor interoperability and flexibility in network design.
- xApps (External Applications): As will be explained below in greater detail, xApps are software applications that run on the RIC.

- Open Interfaces: O-RAN defines open and standardized interfaces for communication between various network elements, including between the RIC and the CU/DU, and between different network functions.
- Virtualization and Orchestration: O-RAN supports virtualization technologies to allow for the flexible deployment of network functions.
- Multi-vendor Environment: O-RAN aims to break vendor lock-in by promoting a multi-vendor ecosystem.
- AI/ML Integration: The O-RAN architecture is designed to accommodate machine learning algorithms. These can be employed to optimize network performance and enable intelligent decision-making.

The essential components of the O-RAN architecture will be adopted in CONVERGE, including its Radio Unit (O-RU), Distributed Unit (O-DU), and Centralized Unit (O-CU), aligned with the standardized 3GPP F1 interface and the Open FH functional split 7.2 [3].

Vision-sensing capabilities will be added to the vision-aided gNB, represented by the video camera on top of O-RU in Figure 3. The CONVERGE gNB will be jointly controlled and accessed via a web service-based interface located in the CONVERGE Core and accessible via the Cgnb interface. The set of web services to be offered to the remote users of this tool is yet to be defined but should enable the validation of new xApps within the near-real time Radio Intelligent Controller (near-RT RIC) taking advantage of multi-modal radio and vision information. An xApp is essentially a software application that runs on top of a RIC platform, which is designed to leverage the capabilities and resources provided by the RIC and are responsible for implementing specific functionalities or services that enhance the operation of the RAN in a 5G network. The RIC is a part of the RAN infrastructure, and its primary objective is to provide intelligent control and optimization functions for radio resources. The concept of near-real time means that these operations occur with very low latency, implying that decisions and actions can be taken quickly to adapt to changing network conditions. xApps within the near-real time RIC can be used for several functions, such as:

- Orchestration and Control: xApps can orchestrate and control various aspects of the RAN, such as beamforming, power allocation, and mobility management.
- Resource Optimization: They can optimize the utilization of radio resources, ensuring that they are allocated efficiently based on the current network conditions.
- Quality of Service (QoS) Management: xApps can influence QoS parameters to ensure that different types of services (e.g., voice, video, data) receive appropriate levels of service quality.
- Intelligence and Analytics: xApps can use AI/ML algorithms to analyse data from the RAN and make intelligent decisions to improve network performance. These algorithms should be provided by the CONVERGE Machine Learning Function.
- Policy Enforcement: They can enforce network policies related to, for example, security, spectrum allocation, and interference management.

4 EXAMPLE OF A CONVERGE TEST SESSION

In this Chapter, we present an example of a CONVERGE test session where a researcher may be experimenting with a vision-aided proactive beam-switching mechanism for 5G FR2 (Frequency Range 2 which includes frequency bands from 24.25 GHz to 71.0 GHz) that makes use of three CONVERGE tools: Vision-aided gNB, LIS and ML algorithms. The goal of this example is to present the interactions between different functions of the CONVERGE architecture and the interfaces required to implement this test case.

In this use case an experimenter wants to test a vision-aided proactive beam-switching mechanism using CONVERGE chamber as illustrated in Figure 4.

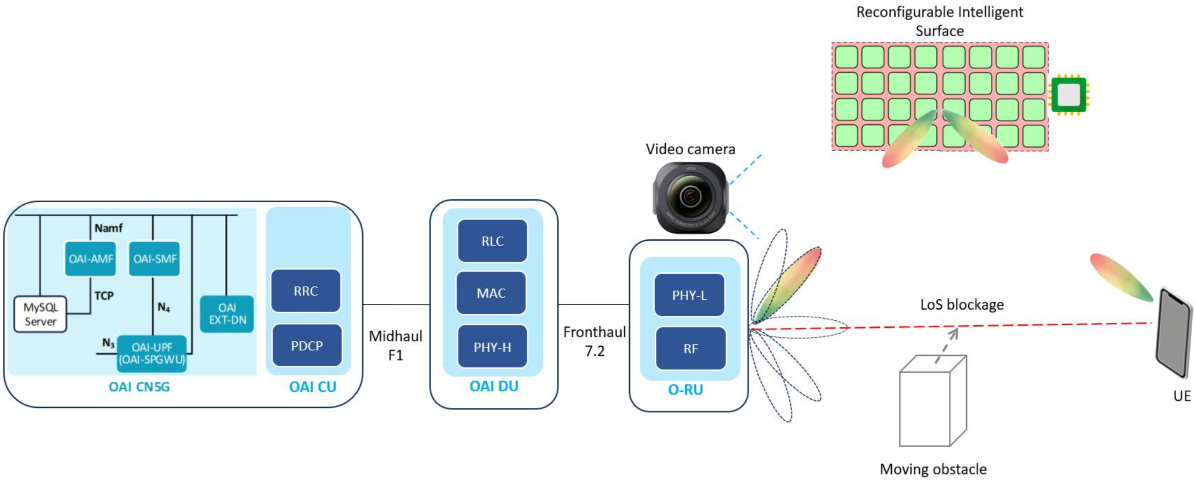

Figure 4 – Conceptual diagram of a vision-aided proactive beam-switching mechanism using the CONVERGE chamber.

Beam failure happens when the UE experiences poor channel conditions, for instance, caused by a sudden Line-of-Sight (LoS) blockage. Once the failure is detected, the gNB will restart the beam sweeping process to reselect a suitable beam for the affected UE. This introduces delays and disruptions in the 5G data link. To obviate this problem, visual data from video cameras such as RGB-D images that capture various hidden features in wireless environments (e.g., object locations and mobility patterns) can be exploited to anticipate the problem happening. In so doing, one can accurately predict future mmWave channel conditions without consuming data resources to probe and estimate the radio propagation channel. Based on video camera sensors deployed at the gNB, a machine-learning algorithm can predict LoS signal blockage and trigger a vision-aided proactive switch to another beam to maintain a stable connection. This use case can also benefit from RIS: before a blockage event occurs, the vision-enabled gNB can switch to a beam directed at the RIS, which then reflects the mmWave signal towards the UE.

Figure 5 shows a flow chart with the high-level step-by-step procedures required to startup and configure the CONVERGE testbed for this test session.

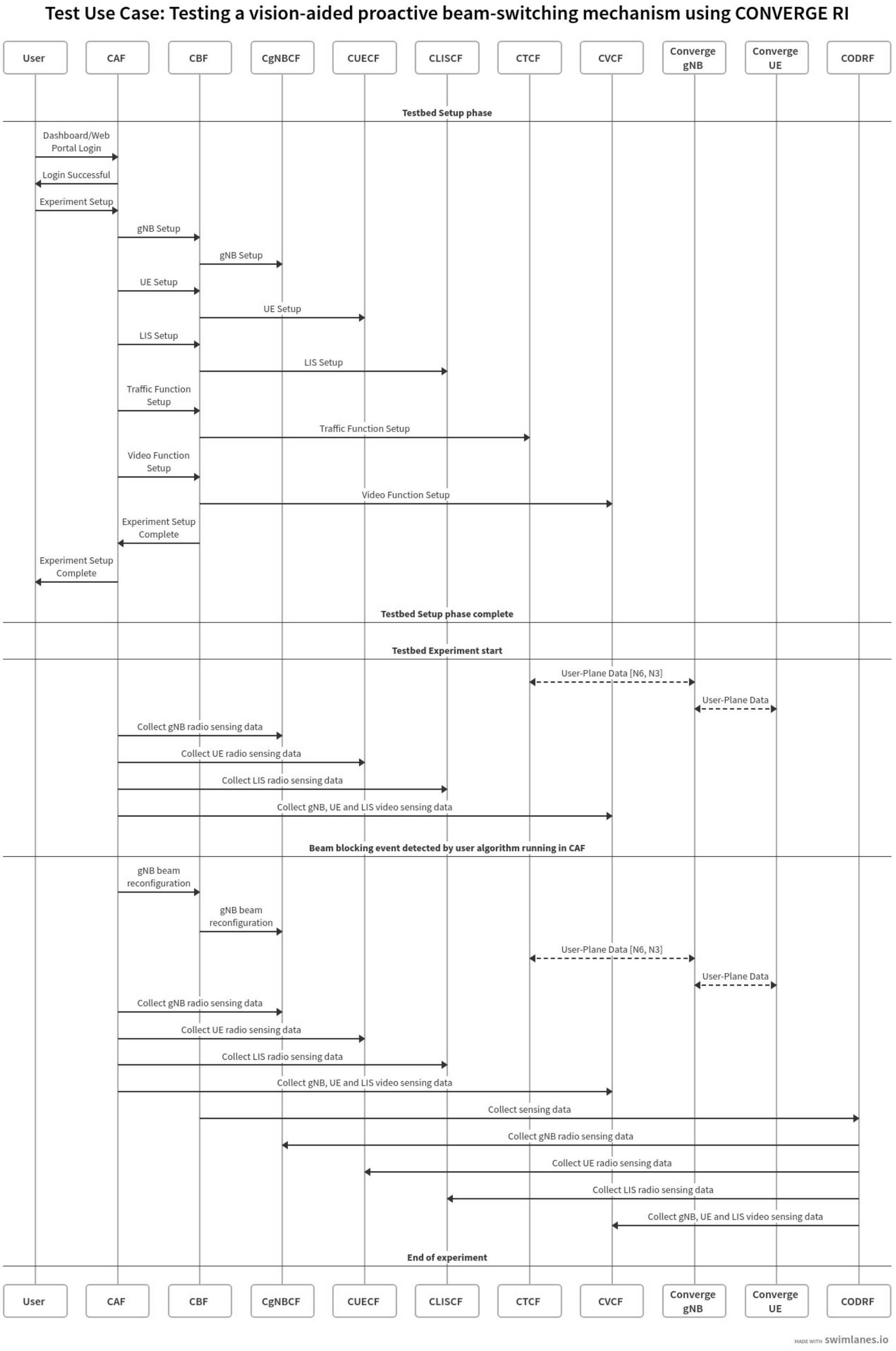

Figure 5 – CONVERGE startup and configuration procedures.

The user starts by the login into the CONVERGE platform, to be authenticated and have access rights to use the testbed and issue a start command to the testbed using the CONVERGE Application Function (CAF). Then, the tester can issue a testbed start command to the CONVERGE Broker Function (CBF) that in turn will issue separate starting commands to all the components that will be required for this use case:

- CONVERGE gNB, the start command issued by the CBF is received in the CONVERGE gNB Core Function, CgNBCF, via Ncgnbf interface. After receiving a start command this function triggers starting actions of each of the individual CONVERGE gNB components, ensuring that the correct configuration is applied: near-NT RIC, O-CU, O-DU, O-RU and Camera Video Stream;
- CONVERGE LIS: receives a start command from the CBF and trigger actions in the internal LIS components;
- CONVERGE UE: receives a start command from the CBF and trigger actions in the internal UE components;
- CONVERGE Traffic Control Function (CTCF), after every other module has reported an online status, the CBF will issue a start data push to the CONVERGE Traffic Control Function that in turn will configure the CONVERGE Traffic Function (CTF), with relevant configurations to establish a data connection and data stream from the CTF to the CONVERGE User Equipment. Moreover, the Tester can take action upon the CTCF directly to manage the data flow, e.g.: stop data stream, and start data stream.

During the startup procedure, the user will have visual indications of the status of each individual task being performed automatically by the CONVERGE Core and have a final visual indication that the testbed has fully started.

After startup, the user can make a request for RAN data related to the running experiment, by issuing a RAN data subscription request to CgNBCF. This request will establish a WebSocket as a transport layer where the producer will be the near-RT RIC and the consumer will be the CONVERGE Web Interface, where the user will be able to visualize and analyse the collected data in real time, or, download it for further offline processing.

To modify the CONVERGE Chamber physically, hence modifying the radio channel conditions, by introducing a LoS blocking element between the O-RU and the CONVERGE UE, the user will be able to control an object via the CAF that will take direct action upon the XYZ robotic crane in the CONVERGE Chamber and move the obstacle programmatically.

After the test is complete, the user should stop the operation of the testbed, by issuing a testbed shutdown command from the CONVERGE Web Interface. This command will be received by the CBF and trigger shutdown action commands for the modules relevant to the experiment: CONVERGE gNB CONVERGE LIS and CONVERGE UE using the CONVERGE Core interfaces, Ncgnbf, Nclisf and Ncuef.

5 SPECIFICATION OF THE CONVERGE CORE INTERFACES

CONVERGE follows a service-oriented architecture very much inspired in the 3GPP 5G CN (core network) architecture.

The 3GPP 5G CN architecture is defined as service-based and the interaction between network functions is represented in two ways:

- A service-based representation, where network functions (e.g., AMF) within the Control Plane enables other authorized network functions to access their services. This representation also includes point-to-point reference points where necessary.
- A reference point representation, shows the interactions that exist between the NF services in the network functions described by the point-to-point reference (e.g. N11) between any two network functions (e.g. AMF and SMF).

The 5G CN System Architecture contains the following service-based interfaces: Namf, Nsmf, Nnef, Npcf, Nudm, Naf, Nnrf, Nnssf, Nausf, Nudr, Nudsf, N5g-eir and Nnwdaf.

The CN APIs in 3GPP is defined by OpenAPI 3.0.0 specifications in YAML format, following guidelines in 3GPP TS 29.501. The 3GPP TS 29.501 defines the designs principles and guidelines for the 5G CN APIs. These principles should be followed when drafting 5G Service Based Interface specifications. The APIs are defined by OpenAPI3.0.0 specification in YAML format. Following sections present in details the new CONVERGE interfaces and their functionalities, following these principles.

5.1 Ncgnbf API

CgNBCF: CONVERGE dedicated gNB Control Function

Key responsibilities of the Ncgnbf interface

In the context of the CONVERGE service-oriented architecture, the CgNBCF is the CONVERGE gNB Control Function, which enables the gNB Start/Stop, configuration, reconfiguration and data collection, by exposing resources through a dedicated interface (API) named Ncgnbf.

Table 1 summarizes the resources made available by this service.

Table 1 - Resources made available by the service.

Resource Name	Resource URI	HTTP Method	Description
CgNB Start/Stop	/control/{start/stop}/{gNB_ID}	GET	The CgNBCF sends a start command via Cgnb interface to NT-RIC, O-CU, O-DU and O-RU
CgNB Configuration status	/configuration/{gNB_ID}	GET	Get the running CgNB configuration
CgNB Reconfiguration	/configuration/{gNB_ID}	POST	The CgNBCF creates and sends a reconfiguration command via Cgnb interface to O-CU, O-DU, O-RU and NT-RIC.
CgNB Data Collection	/telemetry/{gNB_ID}	GET	The CgNBCF retrieves the collected RAN data

In the following we present examples of the use of the Ncgnbf API, showing the interaction between a Service Consumer of the CgNBCF API and the CgNB for the operations CgNB start/stop (Figure 6) and CgNB configuration (Figure 7).

The CgNB Control Start/Stop operation requires a gNB_ID path parameter that should match the gNB ID of the testbed, then the CONVERGE gNB Core Function issues the start/stop command to the gNB components via local Cgnb interface. If this operation is successful, a HTTP 200 ‘OK’ response is sent to the consumer, or, if there are no match in the id path parameter a HTTP 404 ‘Not Found’ is sent to the consumer.

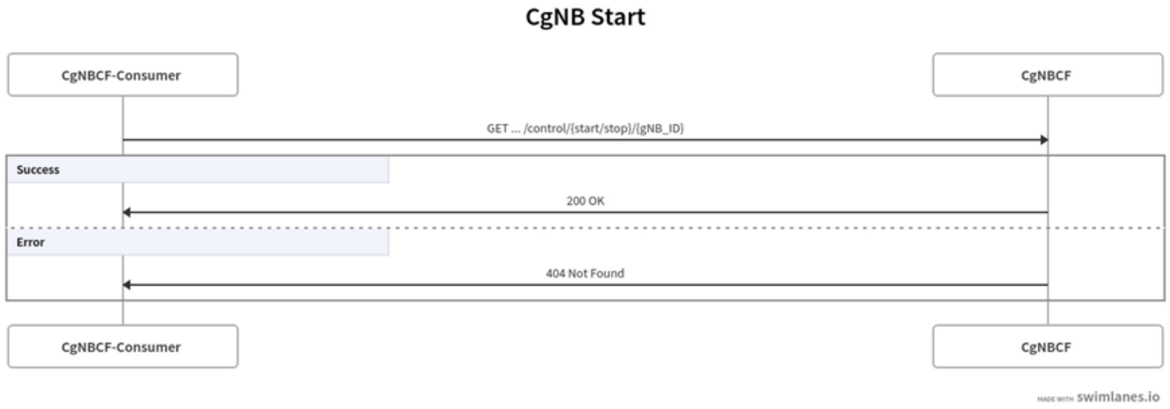

Figure 6 – CgNB start/stop: Interaction between a Service Consumer of the CgNBCF API and CgNBCF.

The CgNB configuration can be retrieved using the HTTP GET request, with the gNB_ID as a path parameter that should match the gNB ID of the testbed, while using an HTTP Post Method, shall be used to update the running configuration for the CgNB by adding the CgNDConfiguration object to the Post request body payload. The response to both request types will contain the CgNBConfiguration object if the request outcome is successful or a HTTP error indicator message if the request fails. The CgNBConfiguration object is described in Table 2 – CgNBConfiguration Attributes.

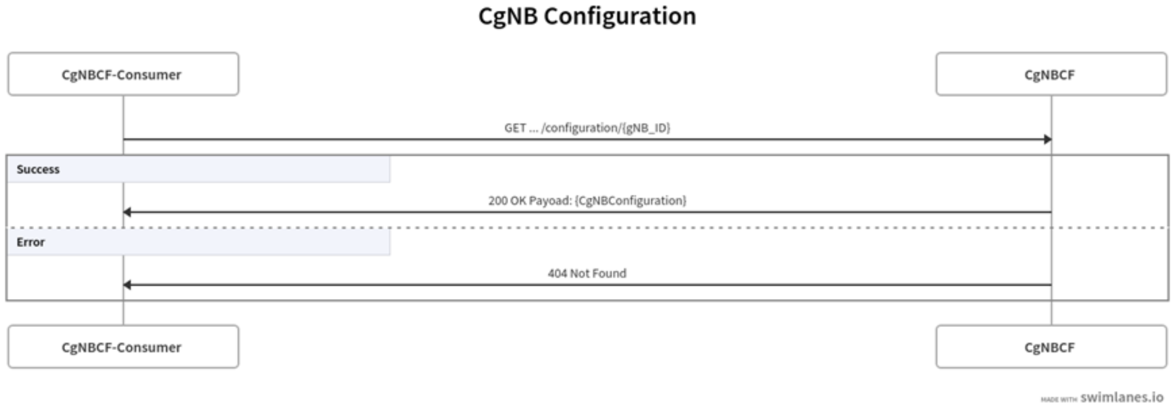

Figure 7 – CgNB configuration: Interaction between a Service Consumer of the CgNBCF API and CgNBCF.

Table 2 - CgNB Configuration Attributes.

Attribute name	Data type	Description
O_CU	JSON Object	Contains needed operational parameters to configure the O-CU component
O_DU	JSON Object	Contains needed operational parameters to configure the O-DU component
O_RU	JSON Object	Contains needed operational parameters to configure the O-RU component
CGNB_PLACEMENT	JSON Object	Contains gNB Placement parameters needed to move the gNB to a new location in the CONVERGE Chamber
CGNB_RADIO_SENSING	JSON Object	Contains gNB Radio Sensing Configurations

Table 3 - O_CU Attributes

Attribute name	Data type	Description
CN	JSON Object	Contains needed operational parameters to configure the O-CU to CONVERGE Core Network interfaces.
F1	JSON Object	Contains needed operational parameters to configure the F1 interface between O-CU and O-DU.

Table 4 - O_DU Attributes

Attribute name	Data type	Description
F1	JSON Object	Contains needed operational parameters to configure the F1 interface between O-CU and O-DU.
7_2	JSON Object	Contains needed operational parameters to configure the O-DU O-RU 7.2 Split
RU	JSON Object	Contains needed parameters to configure the 5G Radio Unit.

Table 5 - O_RU Attributes

Attribute name	Data type	Description
7_2	JSON Object	Contains needed operational parameters to configure the O-DU O-RU 7.2 Split
RU	JSON Object	Contains needed parameters to configure the 5G Radio Unit.

Table 6 - F1 Attributes

Attribute name	Data type	Description
Remote_IP	string	IP of the remote peer in the F1 interface configuration
Remote_Port	integer	Port of the remote peer in the F1 interface configuration
Local_IP	string	Local IP to be bound by the F1 interface locally
Local_Port	integer	Local Port to be bound by the F1 interface locally

Table 7 - RU Attributes

Attribute name	Data type	Description
Remote_IP	string	IP of the remote peer in the F1 interface configuration
Remote_Port	integer	Port of the remote peer in the F1 interface configuration
Local_IP	string	Local IP to be bound by the F1 interface locally
Local_Port	integer	Local Port to be bound by the F1 interface locally

Table 8 - CGNB_PLACEMENT Attributes

Attribute name	Data type	Description
X	integer	X axis coordinate for the CgNB placement
Y	integer	Y axis coordinate for the CgNB placement
Z	integer	Z axis coordinate for the CgNB placement

Table 9 - CGNB_RADIO_SENSING Attributes

Attribute name	Data type	Description
Peridiocity	integer	Measurement report interval in milliseconds
IQ samples	float	In-phase (I) and quadrature-phase (Q) components of the UL radio signal

A CgNB Data Collection request can be made with an HTTP GET Request with the gNB_ID as a path parameter that should match the gNB ID of the testbed (Figure 8). The response will contain the CgNBTelemetry object in the response body if the request is successful or indicate via HTTP error indicator message that the request has failed. The Ncgnbf API is further detailed in YAML format in ANNEX 1.

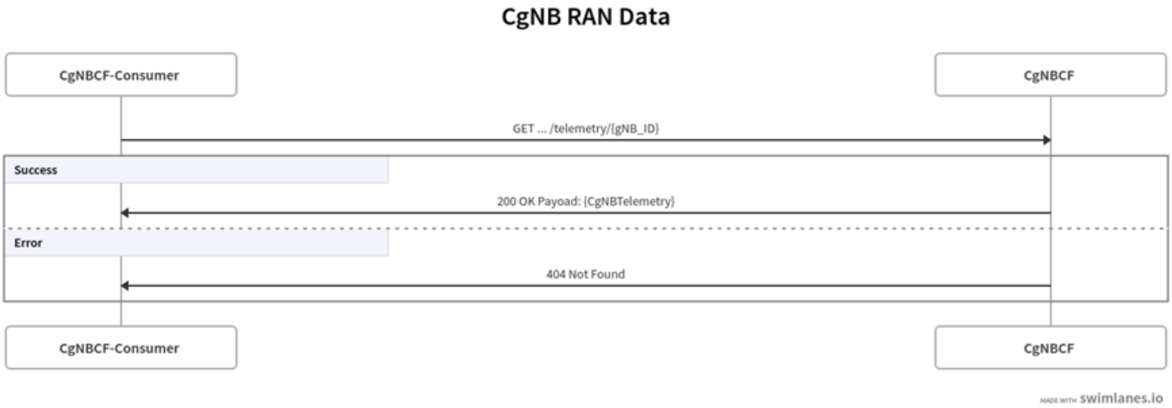

Figure 8 – CgNB RAN data collection configuration: Interaction between a Service Consumer of the CgNBCF API and CgNBCF.

Table 10 - CgNB Telemetry Attributes.

Attribute name	Data type	Description	Unit
RSSI	integer	Received Signal Strength Indicator (RSSI) measures the linear average of the total received power observed only per configured OFDM symbol and in the measurement bandwidth, over *NRB* resource blocks (RB).	dBm
RSRP (SS-RSRP and CSI-RSRP)	integer	Reference Signal Received Power (RSRP) measures the linear average power of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth	Reported values between 0 (<-156 dBm) and 127 (>-31 dBm)
RSRQ (SS-RSRQ and CSI-RSRQ)	integer	Reference Signal Received Quality (RSRQ) is used in 5G NR networks to determine the quality of the radio channel based on Synchronization Signals (SSs).	-20 dB to -10 dB
PHR	integer	Power Headroom Report (PHR) reporting refers to a mechanism that allows the gNB to assess the available power margin of a UE. Power Headroom reporting is important for efficient power management and resource allocation in the network.	Reported values between 0 (<-32 dB) and 63 (> 38 dB)
SINR (SS-SINR	integer	Signal to Interference plus Noise Ratio (SINR) is a quality measurement which	Reported values between 0 (<-23

and CSI-SINR)		represents the ratio of the wanted signal power to the interference plus noise power.	dB) and 127 (> 40 dB)
CQI	integer	Channel Quality Indicator (CQI) values allow a UE to quantify and report (using either the PUCCH or the PUSCH) its downlink radio channel conditions within a specific Bandwidth Part.	Reported values between 0 and 15
MCS	integer	The Modulation and Coding Scheme (MCS) corresponds to a row within the relevant MCS look-up table, and it is allocated by an algorithm belonging to the gNB.	Reported values between 0 and 31
BLER	float	The Block Error Rate (BLER) is defined as the number of erroneous received code blocks.	Reported values between 0 and 100
Bitrate	float	The bitrate is the total number of bits transferred per second.	bit/s
RI	integer	The Rank Indicator (RI) is a feedback parameter that is sent by the UE to the gNB to indicate the number of independent spatial streams that can be supported by the MIMO radio channel.	Reported values between 1 and 8

Other indirect Physical Layer parameters that can be computed on the gNB side are specified by 3GPP TS 38.215 [3]:

- UL Relative Time of Arrival (TUL-RTOA)
- gNB Rx - Tx time difference
- UL Angle of Arrival (UL AoA)
- Timing advance (TADV)

5.2 Ncuef API

CUECF: CONVERGE dedicated UE Control Function

In the context of the CONVERGE service-oriented architecture, the CUECF is the CONVERGE dedicated UE control function.

Key responsibilities of the CUE

1) Placement control: The position of the UE within the CONVERGE chamber may be controllable freely in a x-y-z coordinate system, allowing experimenters to evaluate the impact of different LIS positions within a certain communications, sensing or integrated communication and sensing scenario.
2) Radio communications: Refers to the control of the UE, specifically for configuration aspects related to the resources available for communications.
3) Radio sensing: Refers to the control of the UE that is specific to radio sensing applications. This allows for radio sensing resource configuration (e.g., radio sensing resource allocation) and radio sensing measurement configuration. This allows the UE to be instructed to, e.g.,

perform specific measurements over the dedicated or allocated radio resources that serve a sensing related purpose.

Table 11 - Ressources made available by the service.

Resource Name	Resource URI	HTTP Method	Description
CUE Start/Stop	/control/{start/stop}/{UE_ID}	GET	The CUECF sends a start command via CUE interface to the relevant UE.
CUE Configuration status	/configuration/{UE_ID}	GET	Get the running CUE configuration
CUE Reconfiguration	/configuration/{UE_ID}	POST	The CUECF creates and sends a reconfiguration command via CUE interface to the relevant UE.
CUE Data Collection	/telemetry/{UE_ID}	GET	The CUECF retrieves the collected UE data

Service Operations

CUE Control Start/Stop operation requires a UE_ID path parameter that should match the UE ID in the testbed. The CONVERGE UE Core Function then issues a start/stop command to the UE components via local CUE interface. If this operation is successful, a HTTP 200 ‘OK’ response is sent to the consumer. A failure in this procedure if there are no match in the id path parameter a HTTP 404 ‘Not Found’ is sent to the consumer.

The full list of CUE telemetry parameters from 3GPP TS38.215 is described in detail in Annex 2.

5.3 Ncaf API

CAF: CONVERGE Application Function

In the context of the CONVERGE service-oriented architecture, the CAF is the CONVERGE Application Function, which will interface with CONVERGE users that, through remote access from the internet and according to specified access policies, enable them to configure, monitor, and control experiments to be carried out in the CONVERGE infrastructure through a dedicated interface (API) named Ncaf.

Key responsibilities of the NCAF interface

The interface of the NCAF will enable to configure, monitor and control experiments, namely:

1) Account Management: Enables the user authentication, logout and password reset.
2) Data retrieval from Open Data Repository: depending on the access policies,a user can download a dataset of a previous experiment.
3) Experiment Setup: The experiment setup will be defined by the user throught the CAF. Also, the user can setup the video and radio sensing to be collected.
4) Machine Learning models: the user can download or upload a Machine Learning Model.

Services operations

In the following we provide a summary of the resources made available by this service.

Table 12 - Resources made available by the CAF service.

Resource Name	Resource URI	HTTP Method	Description
Authenticate User	/accounts/login	POST	Authenticate and login a user
Logout User	/accounts/logout	POST	Logout a user
Reset Password	/accounts/reset	POST	Request a password reset for a user
Download Dataset	/open-data-repository/datasets/{id}	GET	Download an available dataset
Experiment Setup Status	/experiment-setup/{id}/status	GET	Get experiment setup status
Experiment Setup	/experiment-setup	POST	Set experiment setup
Experiment Video Sensing	/experiment-setup/{id}/video-sensing	GET	Get video sensing data
Experiment Radio Sensing	/experiment-setup/{id}/radio-sensing	GET	Get radio sensing data
ML Model Download	/ml-models/{id}	GET	Download an ML model
ML Model Upload	/ml-models	POST	Upload an ML model

Table 13 - Accounts Attributes.

Attribute name	Data type	Description
User ID	Integer	Unique identifier for a user
Username	String	Unique name for login of a user
Password	String	Password for user authentication
Email address	String	Email address associated with a user for notifications and account recovery purposes

Table 14 - Dataset Attributes.

Attribute name	Data type	Description
Dataset ID	Integer	Unique identifier for the dataset
Dataset name	String	Name of the dataset

Table 15 - Experiment Setup Attributes.

Attribute name	Data type	Description
Experiment ID	Integer	Unique identifier for the experiment setup
gNB Configuration	JSON object	Configuration for the gNB
UE Configuration	JSON object	Configuration for the UE
LIS Configuration	JSON object	Configuration for the LIS

Traffic Control Configuration	JSON object	Configuration for the Traffic Control Function
Status	String	Status of a given experiment

Table 16 - Video Sensing Attributes.

Attribute name	Data type	Description
Experiment ID	Integer	Unique identifier for the experiment setup
Timestamp	DateTime	Timestamp when the video data was captured
Video Data	Binary/URL	Video file or stream link

Table 17 - Radio Sensing Attributes.

Attribute name	Data type	Description
Experiment ID	Integer	Unique identifier for the experiment setup
Radio Data	JSON object	Radio sensing data

Table 18 - ML Algorithms Attributes.

Attribute name	Data type	Description
ML Model ID	Integer	Unique identifier for the ML model
ML Model Name	String	Name of the ML model
Model File	Binary	Model file or download link

5.4 Nclisf API

CLISCF: CONVERGE dedicated LIS Control Function

In the context of the CONVERGE service-oriented architecture, the CLISCF is the CONVERGE Large Intelligent Surface Control Function, which enables the control of experiments involving the LIS, by exposing resources through a dedicated interface (API) named Nclisf.

Key responsibilities of the CLISCF interface

The interface of the CLISCF will enable to control the LIS in different aspects, namely:

1) Placement control: The horizontal position of the LIS within the chamber (e.g. along a wall) may be controllable, allowing experimenters to evaluate the impact of different LIS positions within a certain communications, sensing or integrated communication and sensing scenario.
2) Radio communications: This corresponds to the control of the LIS specifically for radio communications applications, allowing for the selection of pre-defined communications beams in terms of beam-width and beam target reflection angles, as well as fully customisable LIS phase profiles in order to enable maximum flexibility for less conventional research experiments. The control of a communications beam in real-time should be realised through multiple calls of this service along time.
3) Radio sensing: This corresponds to the control of the LIS specifically for radio sensing applications, allowing for the configuration of fully customisable phase profiles with controllable time-varying characteristics, enabling maximum flexibility for the research community endeavours in this field of reconfigurable intelligent surface-aided sensing.

Services operations

In the following we provide a summary of the resources made available by this service, as well as a list of attributes for the parameters envisioned to be exchanged via the API.

Table 19 - Resources made available by the CLISCF service.

Resource Name	Resource URI	HTTP Method	Description
Placement start/stop	/control/placement/{start/stop}/{LIS_ID}	POST	A start command should initiate the movement of the LIS to the target position; A stop command should stop the movement at any time.
Placement configuration	/configuration/placement/{LIS_ID}	POST	Configures the placement position of the LIS.
Placement status	/configuration/placement/{LIS_ID}/status	GET	Get the current LIS placement information.
Radio communications start/stop	/control/radio-comms/{start/stop}{LIS_ID}	POST	A start command should deploy the configured LIS radio communications setup; A stop command should revert the LIS radio communications setup to default.
Radio communications configuration	/configuration/radio-comms/{LIS_ID}	POST	Configures the LIS radio communications parameters.
Radio communications feedback	/configuration/radio-comms/{LIS_ID}/feedback	POST	Enables the provision of feedback in terms of UE received power, to allow for automatic beam calibration.
Radio communications status	/configuration/radio-comms/{LIS_ID}/status	GET	Get the current LIS radio communications parameters.
Radio sensing start/stop	/control/radio-sensing/{start/stop}/{LIS_ID}	POST	A start command should deploy the configured LIS radio sensing setup; A stop command should stop the sensing time sequence and revert the LIS radio sensing setup to default.
Radio sensing configuration	/configuration/radio-sensing/{LIS_ID}	POST	Configures the LIS radio sensing parameters.
Radio sensing status	/configuration/radio-sensing/{LIS_ID}/status	GET	Get the current LIS radio sensing parameters.

Table 20 - List of Attributes

Attribute Name	Data Type	Description
Placement configuration	float	This value indicates the target absolute position of the LIS in metres. Value between 0 and x_max, where x_max represents the maximum travel of the LIS robotic positioner.
Radio communications configuration	JSON Object	Contains the required parameters to configure the LIS radio communications capabilities.
Radio communications feedback	Integer	Feedback from UE to LIS reporting the Reference Signal Received Power (RSRP), to allow for automatic beam calibration; Reported values between 0 (<-156 dBm) and 127 (>-31 dBm).
Radio sensing configuration	JSON Object	Contains the needed operational parameters to configure the LIS radio sensing capabilities.

Table 21 - List of Attributes of Radio communications configuration

Attribute Name	Data Type	Description
Pre-configured beam	JSON Object	Contains the required parameters to configure a beam from a list (code-book) of pre-defined beams.
Custom phase profile	Binary vector	Binary vector with a size that is equal to the total number of unit cells of the LIS.

Table 22 - List of Attributes of Pre-configured beam

Attribute Name	Data Type	Description
Source angle	Integer vector	Vector of two values in the range (0-720) representing the beam azimuth (-90º to +90º) and elevation (-90º to +90º) with a step-size of 0.25º.
Destination angle	Integer vector	Vector of two values in the range (0-720) representing the beam azimuth (-90º to +90º) and elevation (-90º to +90º) with a step-size of 0.25º.
Beam-width	Integer	Value in the range (0-100) representing the beam-width from a list of pre-defined beam-widths.

Table 23 - List of Attributes of Radio sensing configuration

Attribute Name	Data Type	Description
Time-step	float	The time-step required for the update of the LIS phase profile in ms. Value between 1 and ts_max, where ts_max represents the maximum allowed time-step.
Sensing custom phase profile	Binary vector	Binary vector with a size that is a multiple of the total number of unit cells of the LIS; using the value 0 defines a random phase profile.

5.5 Ncbf API

CBF: CONVERGE Broker, Session orchestration, data management

In the context of the CONVERGE service-oriented architecture, the CBF is the CONVERGE Broker Function, that will orchestrate the execution of a CONVERGE session carried out by equipment or simulation tools, being also responsible for coordinating the storage of data resulting from a CONVERGE session.

Key responsibilities of the NCBF interface

The interface of the NCBF will enable to coordinate the experiments, namely:

1) gNB Management: configure the placement, radio communicatins and radio sensing of the gNB. It also enables to upload and get the status of an x-APP running on the gNB.
2) UE Management: configure the placement, radio communicatins and radio sensing of the UE.
3) LIS Management: configure the placement, radio communicatins and radio sensing of the LIS.
4) Traffic Control Function Management: configure the parameters of the CONVERGE Traffic Control Function.
5) Simulation Session Management: configure the parameters for the 3D-S, VR-S and NET-S simulators.
6) Open Data Repository Management: responsible for retrieving ad uploading a dataset from/to the CONVERGE Open Data Repository.
7) Machine Learning Algorithms: allows to retrieve and upload a Machine Learning model.
8) Experiment status: allows to get the current experiment status.

Services operations

In the following we provide a summary of the resources made available by this service, as well as a list of attributes for the parameters envisioned to be exchanged via the API.

Table 24 - Resources made available by the CBF service.

Resource Name	Resource URI	HTTP Method	Description
gNB Placement Status	/gNB/placement-status/{gNB_ID}	GET	Get the status of gNB placement
gNB Placement Setup	/gNB/placement-setup/{gNB_ID}	POST	Setup the gNB placement
gNB Radio Communications Status	/gNB/radiocommunications-status/{gNB_ID}	GET	Get the status of gNB radio communications
gNB Radio Communications Setup	/gNB/radiocommunications-setup/{gNB_ID}	POST	Setup the gNB radio communications
gNB Radio sensing Status	/gNB/radiosensing-status/{gNB_ID}	GET	Get the status of gNB radio sensing
gNB Radio Sensing Status	/gNB/radiosensing-setup/{gNB_ID}	POST	Setup the gNB radio sensing
gNB Video sensing Status	/gNB/videosensing-status/{gNB_ID}	GET	Get the status of gNB video sensing
gNB Video Sensing Status	/gNB/radiosensing-setup/{gNB_ID}	POST	Setup the gNB video sensing
gNB x-APP Status	/gNB/xapp-status/{id}	GET	Get the status of a specified x-APP
gNB x-APP Upload	/gNB/xapp-setup/{id}	POST	Upload a x-APP to gNB
UE Placement Status	/UE/placement-status/{UE_ID}	GET	Get the status of UE placement
UE Placement Setup	/UE/placement-setup/{UE_ID}	POST	Setup the UE placement
UE Radio Communications Status	/UE/radiocommunications-status/{UE_ID}	GET	Get the status of UE radio communications
UE Radio Communications Setup	/UE/radiocommunications-setup/{UE_ID}	POST	Setup the UE radio communications
UE Radio sensing Status	/UE/radiosensing-status/{UE_ID}	GET	Get the status of UE radio sensing
UE Radio Sensing Status	/UE/radiosensing-setup/{UE_ID}	POST	Setup the UE radio sensing
UE Video sensing Status	/UE/videosensing-status/{UE_ID}	GET	Get the status of UE video sensing
UE Video Sensing Status	/UE/radiosensing-setup/{UE_ID}	POST	Setup the UE video sensing
LIS Placement Status	/LIS/placement-status/{LIS_ID}	GET	Get the status of LIS placement
LIS Placement Setup	/LIS/placement-setup/{LIS_ID}	POST	Setup the LIS placement
LIS Radio Communications Status	/LIS/radiocommunications-status/{LIS_ID}	GET	Get the status of LIS radio communications

LIS Radio Communications Setup	/LIS/radiocommunications-setup/{LIS_ID}	POST	Setup the LIS radio communications
LIS Radio sensing Status	/LIS/radiosensing-status/{LIS_ID}	GET	Get the status of LIS radio sensing
LIS Radio Sensing Status	/LIS/radiosensing-setup/{LIS_ID}	POST	Setup the LIS radio sensing
LIS Video sensing Status	/LIS/videosensing-status/{LIS_ID}	GET	Get the status of LIS video sensing
LIS Video Sensing Status	/LIS/radiosensing-setup/{LIS_ID}	POST	Setup the LIS video sensing
CTCF Status	/CTCF/status/{CTCF_ID }	GET	Get the status of CTCF
CTCF Setup	/CTCF/setup/{CTCF_ID}	POST	Setup the CTCF
3D-S Status	/3DS/status	GET	Get the status of 3D-S simulator
3D-S Setup	/3DS/setup	POST	Setup the 3D-S simulator
VR-S Status	/VRS/status	GET	Get the status of VR-S simulator
VR-S Setup	/VRS/setup	POST	Setup the CTCF
NET-S Status	/NETS/status	GET	Get the status of CTCF
NET-S Setup	/NETS/setup	POST	Setup the CTCF
ML Model Upload	/ML/upload	POST	Upload a Machine Learning Model to CMLF
ML Model Download	/ML/download	GET	Download a Machine Learning Model from CMLF
ODR Dataset Download	/ODR/download	GET	Download a Dataset from CODR
ODR Dataset Upload	/ODR/upload	POST	Post a Dataset from CODR
Experiment Status	/CBF/status	GET	Get experiment status

Table 25 – gNB attributes

Attribute name	Data type	Description
O_CU	JSON Object	Contains needed operational parameters to configure the O-CU component
O_DU	JSON Object	Contains needed operational parameters to configure the O-DU component
O_RU	JSON Object	Contains needed operational parameters to configure the O-RU component
CGNB_PLACEMENT	JSON Object	Contains gNB Placement parameters needed to move the gNB to a new location in the CONVERGE Chamber
CGNB_RADIO_SENSING	JSON Object	Contains gNB Radio Sensing Configurations
CN	JSON Object	Contains needed operational parameters to configure the O-CU to CONVERGE Core Network interfaces.
F1	JSON Object	Contains needed operational parameters to configure the F1 interface between O-CU and O-DU.
7_2	JSON Object	Contains needed operational parameters to configure the O-DU O-RU 7.2 Split

RU	JSON Object	Contains needed parameters to configure the 5G Radio Unit.
Remote_IP	string	IP of the remote peer in the F1 interface configuration
Remote_Port	integer	Port of the remote peer in the F1 interface configuration
Local_IP	string	Local IP to be bound by the F1 interface locally
Local_Port	integer	Local Port to be bound by the F1 interface locally
X	integer	X axis coordinate for the CgNB placement
Y	integer	Y axis coordinate for the CgNB placement
Z	integer	Z axis coordinate for the CgNB placement
Peridiocity	integer	Measurement report interval in milliseconds
IQ samples	float	In-phase (I) and quadrature-phase (Q) components of the radio signal
RSSI	integer	Received Signal Strength Indicator (RSSI) measures the linear average of the total received power observed only per configured OFDM symbol and in the measurement bandwidth, over NRB resource blocks (RB).
RSRP (SS-RSRP and CSI-RSRP)	integer	Reference Signal Received Power (RSRP) measures the linear average power of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth
RSRQ (SS-RSRQ and CSI-RSRQ)	integer	Reference Signal Received Quality (RSRQ) is used in 5G NR networks to determine the quality of the radio channel based on Synchronization Signals (SSs).
PHR	integer	Power Headroom Report (PHR) reporting refers to a mechanism that allows the gNB to assess the available power margin of a UE. Power Headroom reporting is important for efficient power management and resource allocation in the network.
SINR (SS-SINR and CSI-SINR)	integer	Signal to Interference plus Noise Ratio (SINR) is a quality measurement which represents the ratio of the wanted signal power to the interference plus noise power.
CQI	integer	Channel Quality Indicator (CQI) values allow a UE to quantify and report (using either the PUCCH or the PUSCH) its downlink radio channel conditions within a specific Bandwidth Part.
MCS	integer	The Modulation and Coding Scheme (MCS) corresponds to a row within the relevant MCS look-up table, and it is allocated by an algorithm belonging to the gNB.
BLER	float	The Block Error Rate (BLER) is defined as the number of erroneous received code blocks.
Bitrate	float	The bitrate is the total number of bits transferred per second.
RI	integer	The Rank Indicator (RI) is a feedback parameter that is sent by the UE to the gNB to indicate the number of independent spatial streams that can be supported by the MIMO radio channel.

The UE radio sensing (telemetry) attributes are available on Annex 1.

Table 26 - LIS Attributes

Attribute name	Data type	Description
Placement configuration	float	This value indicates the target absolute position of the LIS in metres. Value between 0 and x_max, where x_max represents the maximum travel of the LIS robotic positioner.
Radio communications configuration	JSON Object	Contains the required parameters to configure the LIS radio communications capabilities.
Radio communications feedback	Integer	Feedback from UE to LIS reporting the Reference Signal Received Power (RSRP), to allow for automatic beam calibration; Reported values between 0 (<-156 dBm) and 127 (>-31 dBm).
Radio sensing configuration	JSON Object	Contains the needed operational parameters to configure the LIS radio sensing capabilities.
Pre-configured beam	JSON Object	Contains the required parameters to configure a beam from a list (code-book) of pre-defined beams.
Custom phase profile	Binary vector	Binary vector with a size that is equal to the total number of unit cells of the LIS.
Source angle	Integer vector	Vector of two values in the range (0-720) representing the beam azimuth (-90º to +90º) and elevation (-90º to +90º) with a step-size of 0.25º.
Destination angle	Integer vector	Vector of two values in the range (0-720) representing the beam azimuth (-90º to +90º) and elevation (-90º to +90º) with a step-size of 0.25º.
Beam-width	Integer	Value in the range (0-100) representing the beam-width from a list of pre-defined beam-widths.
Time-step	float	The time-step required for the update of the LIS phase profile in ms. Value between 1 and ts_max, where ts_max represents the maximum allowed time-step.
Sensing custom phase profile	Binary vector	Binary vector with a size that is a multiple of the total number of unit cells of the LIS; using the value 0 defines a random phase profile.

Table 27 - CTCF Attributes

Attribute name	Data type	Description
RuleName	String	The name or identifier of the traffic control rule.
TrafficType	String	The type of traffic being controlled (e.g., VoIP, Video Streaming).
Action	String	The action to be taken for traffic matching the rule (e.g., Redirect, Forward).
Destination	String	The destination network or server for the traffic.

Priority	String	The priority level of the rule (e.g., High, Medium, Low).
TotalPackets	Integer	The total number of packets processed by the CTCF.
PacketLoss	Float	The percentage of packet loss in the traffic control.
Latency	Integer	The average latency in milliseconds for traffic.
BandwidthUsage	Float	The current bandwidth usage in megabits per second.

Table 28 - CMLF Attributes

Attribute name	Data type	Description
SessionID	String	User session identifier used to validate that the user is authorized by the broker
Status	Boolean	Indicator of the CMLF service status, which indicates if the service is available
CMLFConfiguration	String	Indicator that the user has been validated and that the CMLF is now able to receive instructions from the user session
Url_predictor	String/File	Location identifier of the configuration file for the model to be trained
Hyperparameters	Array	Set of hyperparameters that complete the configuration of the Machine Learning model.
Dataset	File	Set of files to be used to train the machine learning model.
New_observation	Array	Single observation on which the prediction operation is to be performed
Model_id	String/int	Trained and running model that allows the prediction of new observations provided
New_prediction	Array	Predicted target objective
Model_inventory	List	List containing all the digital components available as pre-trained ML models, made available to the user for use

Table 29 - CODRF Attributes

Attribute name	Data type	Description
SessionID	String	User session identifier used to validate that the user is authorized by the broker
Status	Boolean	Indicator of the CODRF service status, which indicates if the service is available
CODRFConfiguration	String	Indicator that the user has been validated and that the CODRF is now able to receive instructions from the user session
Url_filename	String	Location identifier of the file in the CODRF for upload/download
Dataset	File	Particular file (dataset)
File_inventory	List	List containing all the files in the CODRF made available to the user for use

Table 30 - CNETCF Attributes

Attribute name	Data type	Description
gNB ID	Integer	Unique identifier for the simulated gNB
Coordinates	String	3D Placement coordinates for the simulated gNB
gNB ID	Integer	Unique identifier for the simulated gNB
Frequency band	Float	Carrier frequency of the simulated gNB
Transmission power	Integer	Transmission power of the simulated gNB
UE ID	Integer	Unique identifier for the simulated UE
Coordinates	String	3D Placement coordinates for the simulated UE
UE ID	Integer	Unique identifier for the simulated UE
Frequency band	Float	Carrier frequency of the simulated UE
Transmission power	Integer	Transmission power of the simulated UE
Model type	String	Name of the propagation loss model to use (e.g., Friis, Two-Ray, etc.)
Traffic type	String	Type of traffic to generate (e.g., TCP or UDP)
Bitrate	Float	Bitrate of the traffic to generate
Packet size	Integer	Size of the packets composing the traffic
Simulation ID	Integer	Unique identifier for the simulation
Duration	Float	Duration of the simulation
Iterations	Integer	Number of simulation runs
Simulation ID	Integer	Unique identifier for the simulation
Throughput	Float	Throughput achieved during simulation
Packet Loss Ratio	Float	Packet loss ratio achieved during simulation
Delay	Float	Packet delay achieved during simulation
Jitter	Float	Jitter achieved during simulation
Simulation ID	Integer	Unique identifier for the simulation
Radio Traces	Binary	File with the radio traces captured during a given simulation
Simulation ID	Integer	Unique identifier for the simulation
Events and logs	Binary	File with the events and logs generated during a given simulation

Table 31 - CVRCF Attributes

Attribute name	Data type	Description
Bit rate	Float	The amount of transmitted data in a given amount of time. It refers to the speed at which data is transmitted between the UE and the gNodeB, measured in terms of Mbps.
Bit error rate	Float	The number of bit errors per unit time. It represents the probability of errors occurring during data transmission.

Delay	Float	The average latency in milliseconds for traffic.
RIS position	Float	The specific location and orientations of the RIS
Type of propagation model	Float	The type of the mathematical or empirical representations used to describe how EM waves propagate through the environment. It predicts the behaviour of wireless signals as they travel from a transmitter to a receiver.
Tx position	Float	The location of the transmitter
Rx position	Float	The location of the receiver
Phase	Float	The current adjusted direction, focusing, and amplitude of the reflected or transmitted signals.
Ray tracing	Struct	The used propagation model. It simulates the behavior of EM waves as they interact with objects and surfaces in a 3D environment.
Distance	Float	The current distance between transmitter and receiver, and the distance between the RIS and transmitter and receiver respectively.
Modulation	Struct	Modulation OFDM with 64 QAM with RIS as a used modulation technique.

Table 32 - C3DCF Attributes

Attribute name	Data type	Description
Tx/Rx Locations	Array [Float3]	The 3D location of the transmitters and receivers
Tx/Rx angles	Array [Float4]	The valid angles of transmission and reception “Azimuth, elevation”, the first value corresponding to starting angle, the second to the end angle in counter-clockwise fashion (optional parameter)
3D Environment geometry model (static)	Array [Struct]	A static 3D model of the chamber in “PLY” format containing an array of points representing the basic 3D geometry and visual information. Each point contains as attributes (Float3, 3D_location, int3 RGB_ value).
3D Environment vision-radio model (static)	Array [Struct]	A static 3D model of the chamber in “PLY” format containing an array of points. representing a radio-enhanced 3D model. Each point contains as attributes (Float3, 3D_location, int3 RGB_ value, float3 normal_vector, int label_material, int label_hash, int label_diffraction
3D environment vision-radio model (current)	Array [Struct]	A current 3D model of the chamber in “PLY” format containing an array of points, based on realistic data flows from the chamber. Each point contains as attributes (Float3, 3D_location, int3

		RGB_ value, float3 normal_vector, int label_material, int label_hash, int label_diffraction, float time_t)
Paths ready	Int	Expresses when the path computations are ready and possible error codes
Mapped Paths	Array [struct] [float3]	Paths in the environment, used signal propagation. It can be stored as a file in CSV format containing and array with the traced paths in the 3D environment. Each path contains a set of 3D points from each path interaction with the environment objects, expressing the location of the interaction. The first point is the Tx and the last point is the Rx
Data Flow Inputs	Array [Struct]	Inputs from various data sources, detailing how they influence the environment model.
Number of Radio Devices	Integer	The total count of devices utilized in the simulation.
Number of Cameras	Integer	The total count of cameras used to capture images/videos in the simulation.

Table 33 - CVCF Attributes

Attribute name	Data type	Description
CameraID	String	Camera identifier used to identify a given camera
CameraConfiguration	String	Configuration of a camera in JSON format
CameraStatus	Int	Indicator of a camera status (0-idle, 1-streaming, 2-recording)
SupportModelID	String	Support model identifier used to identify a given support model
SupportModel Configuration	String	Configuration of a support model in JSON format
SuportModelStatus	Boolean	Indicator of the support model service status, which indicates if the service is activated

5.6 Nctf API

CTCF: CONVERGE Traffic Servers Control Function

The CTCF is part of the CONVERGE Core building block, allowing configuration and management of the data flow, e.g.: stop data stream, and start data stream.

Key responsibilities of CTCF interface:

1) Traffic Steering Control: CTCF is tasked with directing and optimizing user plane traffic within the CONVERGE Chamber. It leverages predefined rules and policies to determine how the different types of traffic should be handled as they move between the CONVERGE gNB, CONVERGE UE, and CONVERGE LIS equipment.

2) Interfacing with CONVERGE Core: CTCF collaborates closely with other functions within the CONVERGE Core to ensure seamless connectivity and control over the equipment within the chamber. It communicates with the CONVERGE Application Function (CAF), CONVERGE Broker Function (CBF), and other control functions to coordinate and manage traffic providing access to 5G traffic that can be used with the different tools.
3) Rule-Based Traffic Policies: CTCF operates based on a set of predefined traffic steering rules, which specify how traffic types, source networks, destination networks, and priorities should be managed. These rules are essential for optimizing resource allocation and ensuring quality of service.
4) Real-time Data Handling: CTCF facilitates real-time data handling, allowing for dynamic adjustments to traffic routing based on current network conditions, user demands, and experimental requirements.
5) Integration with CONVERGE Simulator: CTCF may also interact with the CONVERGE Simulator building block to simulate traffic scenarios, enabling researchers to conduct experiments in controlled environments.

In the following we provide a summary of the resources made available by this service, as well as a list of attributes for the parameters envisioned to be exchanged via the API.

Table 34 - Resources made available by the CTCF service.

Resource Name	Resource URI	HTTP Method	Description
Traffic Control Start	/control/{start/stop}/{CTCF_ID}	GET	The CTCF sends a start command to control traffic servers.
Traffic Control Rules	/configuration/{CTCF_ID}/rules	GET	Get the current traffic control rules configuration.
Traffic Control Rules	/configuration/{CTCF_ID}/rules	POST	Add or modify traffic control rules.
Traffic Statistics	/telemetry/{CTCF_ID}	GET	Retrieve statistics related to traffic control.

Service Operations:

Traffic Control Start/Stop Operation: This operation requires a CTCF_ID path parameter that should match the CTCF ID of the service instance. The CTCF issues start/stop commands to control traffic servers via a local interface. If the operation is successful, an HTTP 200 'OK' response is sent to the consumer. If there is no match in the ID path parameter, an HTTP 404 'Not Found' response is sent to the consumer.

The following example displays the CTCF Configuration object.

Traffic Control Rules Configuration: The current traffic control rules configuration can be retrieved using an HTTP GET request with the CTCF_ID as a path parameter. To add or modify traffic control rules, use an HTTP POST request with the CTCF_ID as a path parameter and include the traffic control rules in the request body payload. The response to both request types will contain the traffic control rules if the request outcome is successful, or an HTTP error indicator message if the request fails.

The CTCF defines traffic control rules with specific attributes that dictate how traffic is managed. These attributes include:

Table 35 - Traffic Control Rule Attributes

Attribute Name	Data Type	Description
RuleName	String	The name or identifier of the traffic control rule.
TrafficType	String	The type of traffic being controlled (e.g., VoIP, Video Streaming).
Action	String	The action to be taken for traffic matching the rule (e.g., Redirect, Forward).
Destination	String	The destination network or server for the traffic.
Priority	String	The priority level of the rule (e.g., High, Medium, Low).

Table 3 introduces the telemetry attributes associated with the CTCF. These attributes are used to report on various performance metrics and statistics related to traffic control.

Table 36 - Telemetry Attributes

Attribute Name	Data Type	Description
TotalPackets	Integer	The total number of packets processed by the CTCF.
PacketLoss	Float	The percentage of packet loss in the traffic control.
Latency	Integer	The average latency in milliseconds for traffic.
BandwidthUsage	Float	The current bandwidth usage in megabits per second.

5.7 Ncmlf API

CMLF: CONVERGE Machine Learning Function

In the context of the CONVERGE service-oriented architecture, the CONVERGE Machine Learning Function (CMLF) is part of the CONVERGE core, developed to facilitate the processing of heterogeneous datasets (e.g., video and radio sensing data) produced by the CONVERGE project, as well as those provided by the scientific community. The CMLF provides Machine Learning tools specifically engineered to process and analyse multimodal information to enable new approaches to the paradigm " view-to-communicate" and "communicate-to-view".

Key responsibilities of the CMLF interface:

The interface of the CMLF enables the use, implementation, uploading and downloading of algorithms and data processing techniques, namely:

1) Machine Learning Model Training. This function is called to act as a tool to train the Machine Learning model.
2) Model execution in CONVERGE Chamber. Call to the prediction function of the (public) models hosted at the CONVERGE chamber and execution of the model for remote prediction in the user's servers.
3) Model Uploading and Downloading functionality. Functionality that will allow the user to upload and download pre-trained models for later use
4) Pre-trained model inventory. Allows the user to know a complete list of available digital components.
5) In any of the above cases, a secure interaction between internal and/or external elements of the CONVERGE chamber must be established during the initialization.

Figure 9 shows a high-level procedure required for using the Machine Learning services.

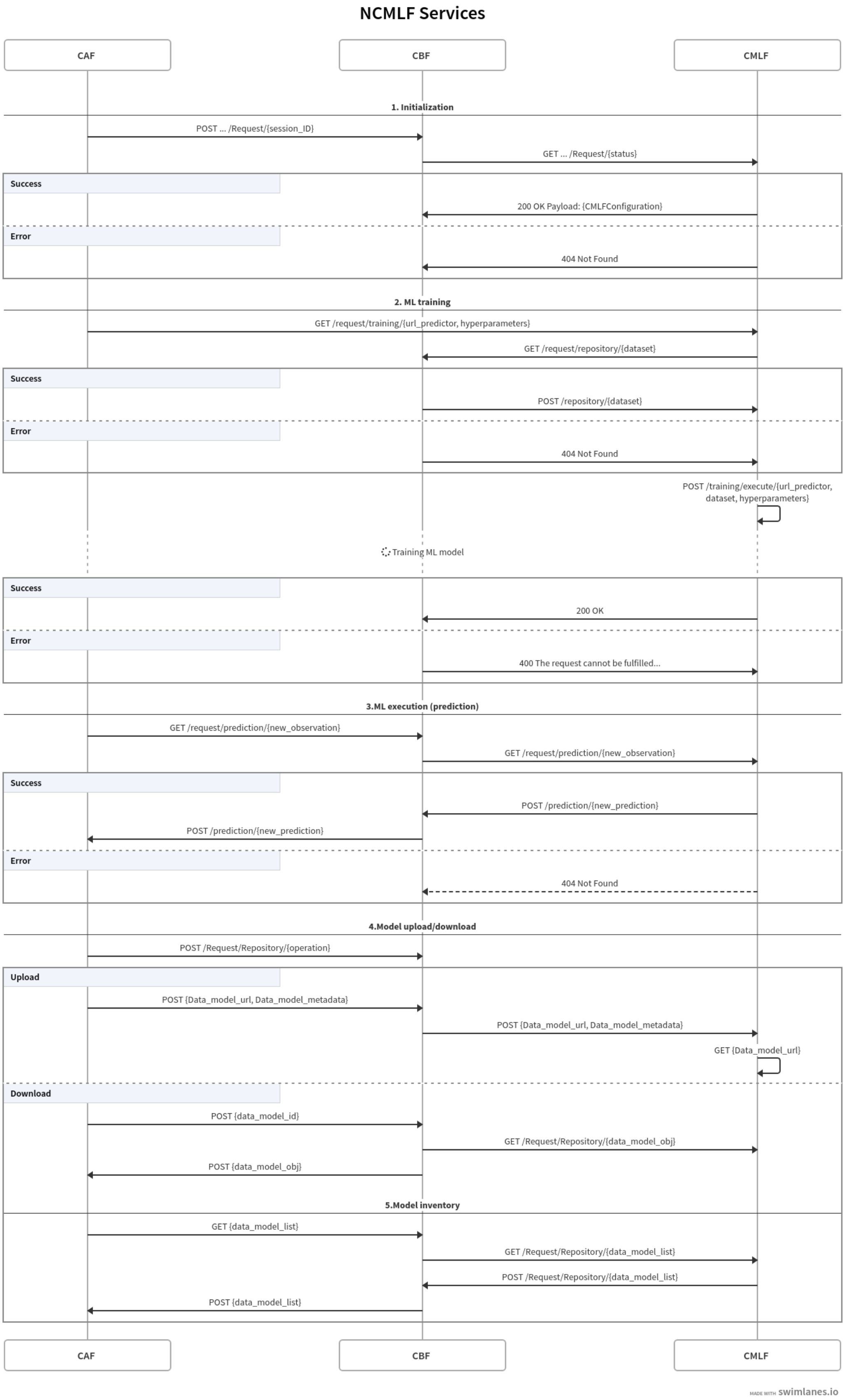

Figure 9 – High-level services provided by the CML function.

Service operations

In the following, we provide a summary of the resources made available by this service, as well as a list of attributes for the parameters envisioned to be exchanged over this interface.

Table 37 - Resources made available by the CMLF service.

Resource Name	Resource URI	HTTP Method	Description
Authenticate user request	/sessions/{SessionID}	GET	Authenticate and log in as a user
Training request	/request/training/{url_predictor, hyperparameters}	GET	Request the training of a new ML model
Download dataset	/request/repository/{dataset}	POST	Download an open-curated dataset
Execute training	/training/execute/{url_predictor, dataset, hyperparameters}	POST	Train a new ML model
New data observation	/request/prediction/{new_observation}	GET	Provide a new observation to a running ML predictor
New Prediction	/prediction/{new_prediction}	POST	Returns a prediction based on the observation previously observed
Upload ML model request	/Request/Repository/{Data_model_url, Data_model_metadata}	GET	Upload an ML model to the repository of digital components.
Register new model	/Request/Repository/{Data_model_url}	POST	Register and assign an ID to the trained model.
Download the ML model request	/Request/Repository/{data_model_id}	GET	Request to download pre-trained ML models for implementation or further training
Deliver requested model	/Request/Repository/{data_model_obj}	POST	Return the digital component requested for download.
List pre-trained models	/Request/Repository/{data_model_list}	GET	Request the list of available digital components in the repository.
Show pre-trained model list	/Request/Repository/{data_model_list}	POST	Shows the list of available digital components in the repository.

Table 38 - List of Attributes

Attribute Name	Data Type	Description
SessionID	String	User session identifier used to validate that the user is authorized by the broker
Status	Boolean	Indicator of the CMLF service status, which indicates if the service is available
CMLFConfiguration	String	Indicator that the user has been validated and that the CMLF is now able to receive instructions from the user session
Url_predictor	String/File	Location identifier of the configuration file for the model to be trained
Hyperparameters	Array	Set of hyperparameters that complete the configuration of the Machine Learning model.
Dataset	File	Set of files to be used to train the machine learning model.

New_observation	Array	Single observation on which the prediction operation is to be performed
Model_id	String/int	Trained and running model that allows the prediction of new observations provided
New_prediction	Array	Predicted target objective
Model_inventory	List	List containing all the digital components available as pre-trained ML models, made available to the user for use

5.8 Ncodrf API

CODRF: CONVERGE Open Data Repository Function

In the context of the CONVERGE service-oriented architecture, the CODRF is the CONVERGE Open Data Repository Function, which enables the storage and retrieval of data generated by CONVERGE. It is implemented as a service operating between the CAF and CBF as noted in the high level architecture and exposes resources through a dedicated interface (API) named Ncodrf.

Key responsibilities of the CODRF interface

The interface of the CODRF will enable different dataset storage/retrieval operations, namely:

1) Creation/storage of datasets. This corresponds to the creation/storage or update of a dataset consisting of radio and/or vision data collected from experiments carried out in the CONVERGE chamber or data collected from the CONVERGE simulator, along with the appropriate metadata (e.g. readme file, data annotation, experimental/simulation conditions).
2) Creation/storage of ML models. This corresponds to the creation/storage or update of trained ML models, along with the required metadata (namely the identifier of the dataset that was used for training the model).
3) Retrieval of datasets and ML models. This corresponds to the service of data retrieval for both stored datasets and ML models.

Types of data

The data to be handled by the CODRF interface will consist mainly of open data, and include the following types:

- Experimental data:
  - Logs from gNBs, UEs and LIS (text or various binary formats)
  - Raw signals from gNBs, UEs and cameras, including time stamped radio streams (IQ samples) and video streams (various binary formats)
- Simulation data: corresponds to data produced within the digital twin component, which includes 3D modelling data, and radio simulation data (ray tracing and network level).
- Model data: corresponds to trained ML models obtained after training within the CMLF (various binary formats).

Services operations

In the following we provide a summary of the resources made available by this service.

Table 39 - Resources made available by the CODRF service.

Resource Name	Resource URI	HTTP Method	Description
Initialization	/initialization/{session_ID}	POST	Authenticates the user request.
Dataset storage	/dataset/storage/{experiment_ID}	POST	Creates a dataset entry referenced by the experiment_ID, and enables the data upload for storage.
Dataset update	/dataset/update/{experiment_ID}	POST	Enables the update of a given part of the data contained in the dataset.
Dataset retrieval	/dataset/retrieval/{experiment_ID}	GET	Enables the retrieval of the data from the dataset referenced by the experiment_ID.
Dataset inventory/search	/dataset/search/{list of keywords}	GET	Returns the list of relevant high-level and metadata information of datasets where the search keywords appear.
ML model storage	/ml_model/storage/{model_ID}	POST	Creates and enables the storage of a ML model referenced by the model_ID
ML model update	/ ml_model /update/{model_ID}	POST	Enables the update of a given part of the ML model data/parameters.
ML model retrieval	/ ml_model /retrieval/{model_ID }	GET	Enables the retrieval of an ML model referenced by the model_ID.
ML model inventory/search	/ ml_model /search/{list of keyword}	GET	Returns the list of relevant high-level and metadata information of ML models where the search keywords appear.

Table 40 - List of Attributes

Attribute Name	Data Type	Description
SessionID	String	User session identifier used to validate that the user is authorized by the broker
Dataset parameters	JSON Object	Contains the required parameters associated with the dataset handling.
ML model parameters	JSON Object	Contains the required parameters associated with the ML model handling.

Table 41 - List of Attributes of Dataset parameters

Attribute Name	Data Type	Description
Metadata	JSON Object	Metadata from dataset
Data contents	File	Particular file (dataset)

Table 42 - List of Attributes of ML model parameters

Attribute Name	Data Type	Description

Metadata	JSON Object	Metadata from ML model
ML model description	File	ML model description file

5.9 Nnetsf API

CNETSCF: CONVERGE Network Simulator Control Function

In the context of the CONVERGE service-oriented architecture, the CNETSCF is the CONVERGE Network Simulator Control Function, which enables the control of network simulations, by exposing simulation resources through a dedicated interface (API) named Nnetsf. The simulation resources include the digital models of the CONVERGE network elements such as the gNB and the UE, but also important auxiliary components such as the Propagation Loss Models and Traffic Generation Applications. These resources allow the CONVERGE Network Simulator to be used as a “Sandbox” of the real CONVERGE chamber, allowing the experimenters to test their developments in a simulated setting without depending on the CONVERGE Chamber availability. This CONVERGE Network Simulator can also be seen as a Digital Twin of the real environment, since its models (such as ML-based Propagation Loss Model) will learn and adapt to the traces captured in the real CONVERGE Chamber.

Key responsibilities of the CNETSCF interface

The interface of the CNETSCF will enable to control the Network Simulator in different aspects, namely:

1) Simulated placement control: As in the case of the real gNB, the positioning of the simulated gNB is fully controllable, allowing it to be set in a fixed (fixed BS) or mobile (mobile BS) configuration along X, Y, Z coordinated. The placement of the UE is also controllable, allowing the experimenters to perform network simulations to evaluate the impact of the relative positioning between the gNB and UE, accounting also to their mobility patterns.
2) Simulation setup: This corresponds to the configuration of the simulated scenario, including the Propagation Loss Model, the Mobility Model, the Network Configuration and Traffic Generation.
3) Simulation orchestration: This corresponds to the control of the start and the stop of the simulations. It also can be used to configure the simulations duration, the random variable seeds and the number of iterations.
4) Simulation data collection: This corresponds to the collection of the performance results (e.g. throughput, packet loss, RTT, etc), mobility traces and radio traces and events that are logged during the simulations.

Services operations

In the following we provide a summary of the resources made available by this service.

Table 43 - Resources made available by the CNETSCF service

Resource Name	Resource URI	HTTP Method	Description
Simulated gNB Placement	/gnb-placement/{gNB-id}	GET	Retrieve simulated gNB placement based on its ID
Simulated gNB Placement Configuration	/gnb-placement/{gNB-id}	POST	Set/update simulated gNB placement considering its ID

Simulated gNB Radio Communications Configuration	/gnb-radio-config/{gNB-id}	GET	Retrieve simulated gNB radio communications configuration considering its ID
Simulated gNB Radio Communications Update	/gnb-radio-config/{gNB-id}	POST	Set/update simulated gNB radio communications configuration considering its ID
Simulated UE Placement Configuration	/ue-placement/{ue-id}	POST	Set/update simulated UE placement considering its ID
Simulated UE Radio Communications Configuration	/ue-radio-config/{ue-id}	GET	Retrieve simulated UE radio communications configuration considering its ID
Simulated UE Radio Communications Update	/ue-radio-config/{ue-id}	POST	Set/update simulated UE radio communications configuration considering its ID
Propagation Loss Model Configuration	/simulation-setup/{simulation-id}/propagation-loss-model	GET	Get propagation loss model configuration
Propagation Loss Model Update	/simulation-setup/propagation-loss-model	POST	Set propagation loss model configuration
Traffic Generation Configuration	/simulation-setup/{simulation-id}//traffic-generation	GET	Get traffic generation parameters
Traffic Generation Update	/simulation-setup/traffic-generation	POST	Set traffic generation parameters
Run Simulation	/simulation-run	POST	Run simulation with a given duration and number of iterations
Simulation Status	/simulation-status/{simulation-id}	GET	Get simulation status considering its ID
Stop Simulation	/simulation-stop/{simulation-id}	POST	Stop simulation considering its ID
Performance Results	/performance-results/{simulation-id}	GET	Retrieve performance results considering simulation ID
Radio Traces	/radio-traces/{simulation-id}	GET	Retrieve radio traces considering simulation ID
Events and Logs	/logging/{simulation-id}	GET	Retrieve events and logs considering simulation ID

Table 44 - Simulated gNB placement attributes

Attribute name	**Data type**	**Description**	**Unit**
gNB ID	Integer	Unique identifier for the simulated gNB	N.A.
Coordinates	String	3D Placement coordinates for the simulated gNB	m

Table 45 - Simulated gNB radio communications attributes

Attribute name	Data type	Description	Unit
gNB ID	Integer	Unique identifier for the simulated gNB	N.A.
Frequency band	Float	Carrier frequency of the simulated gNB	Hz
Transmission power	Integer	Transmission power of the simulated gNB	dBm

Table 46 - Simulated UE placement attributes

Attribute name	Data type	Description	Unit
UE ID	Integer	Unique identifier for the simulated UE	N.A.
Coordinates	String	3D Placement coordinates for the simulated UE	m

Table 47 - Simulated UE radio communications attributes

Attribute name	Data type	Description	Unit
UE ID	Integer	Unique identifier for the simulated UE	N.A.
Frequency band	Float	Carrier frequency of the simulated UE	Hz
Transmission power	Integer	Transmission power of the simulated UE	dBm

Table 48 - Propagation loss model attributes

Attribute name	Data type	Description	Unit
Model type	String	Name of the propagation loss model to use (e.g., Friis, Two-Ray, etc.)	N.A.

Table 49 - Traffic generation configuration attributes

Attribute name	Data type	Description	Unit
Traffic type	String	Type of traffic to generate (e.g., TCP or UDP)	N.A.
Bitrate	Float	Bitrate of the traffic to generate	bit/s
Packet size	Integer	Size of the packets composing the traffic	Byte

Table 50 - Simulation run attributes

Attribute name	Data type	Description	Unit
Simulation ID	Integer	Unique identifier for the simulation	N.A.
Duration	Float	Duration of the simulation	s
Iterations	Integer	Number of simulation runs	N.A.

Table 51 - Performance results attributes

Attribute name	Data type	Description	Unit
Simulation ID	Integer	Unique identifier for the simulation	N.A.
Throughput	Float	Throughput achieved during simulation	bit/s
Packet Loss Ratio	Float	Packet loss ratio achieved during simulation	N.A.

Delay	Float	Packet delay achieved during simulation	s
Jitter	Float	Jitter achieved during simulation	s

Table 52 - Radio traces attributes

Attribute name	Data type	Description	Unit
Simulation ID	Integer	Unique identifier for the simulation	N.A.
Radio Traces	Binary	File with the radio traces captured during a given simulation	N.A.

Table 53 - Logging attributes

Attribute name	Data type	Description	Unit
Simulation ID	Integer	Unique identifier for the simulation	N.A.
Events and logs	Binary	File with the events and logs generated during a given simulation	N.A.

5.10 Nvrsf API

CVRSCF: CONVERGE Vision Radio Simulator Control Function

The CVRSCF is a critical element of the CONVERGE high-level architecture, serving as a key component responsible for managing and orchestrating simulation experiments within the CONVERGE anechoic chamber. The VR-S API is part of the CONVERGE simulator building block, which consists of a ray-tracing based simulation tool that employs wave reflections models from 3.5 GHz to 28 GHz on environmental surfaces. Both 3D simulated models and real-world measurement-based scenes are supported. The simulator uses 3D models of a real experimental chamber featuring configurations that the user can modify. The simulator also supports configurable models of radio transmitters and receivers, as well as imaging equipment such as visible light, infrared cameras, and 3D scanners. Therefore, graphical rendering tools can be used to create rich visualizations in the electromagnetic domain, in addition to simulated camera views and/or real camera views augmented with measurement data.

Key responsibilities of the CVRSCF interface

The configuration interface of the vision-radio simulator will allow to take advantage of both planned and exploratory modes:

1. Planned: the simulator allows for setting known trajectories and speeds of the obstacles (including a simulated person), radios and cameras. This allows for producing simulated synchronised video and radio channel data in predetermined dynamic situations. At selected points the trajectories and radio parameters are made to match with and are calibrated against the prior measurement traces from the real-world chamber experiments. The results provide data for the automated learning of view-to-communicate algorithms that infer environment information from 2D views, 3D sensor or 360° video.
2. Exploratory modes: the environment is static, while simulated mobile radios and cameras are controlled by a machine learning system. The aim is to gather information of the environment in a communicate-to-view mode, using multipath radio signal propagation data in order to direct cameras to acquire environmental information. This mode requires calibration data from measurements in the experimental chamber.

High level workflow of the CVRSCF interface

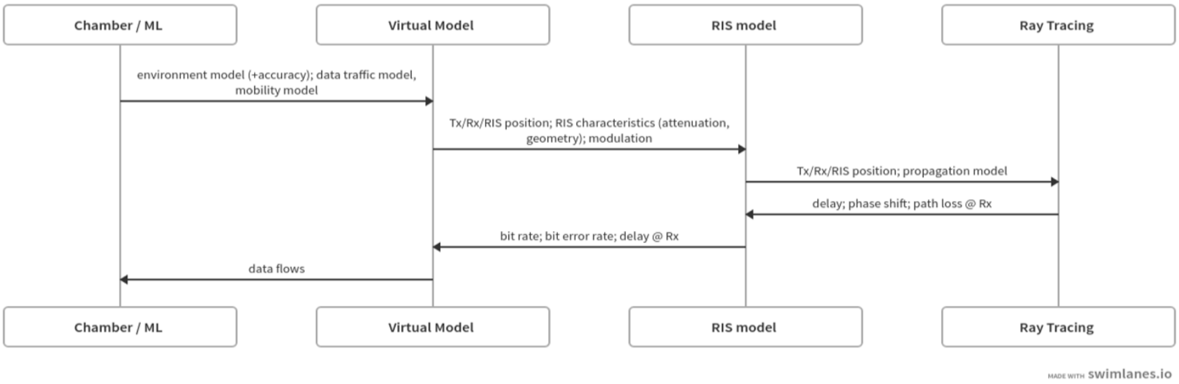

Figure 10 – High level workflow of the VR-S API.

Services operations

In the following we provide a summary of the resources made available by this service.

Table 54 - Resources made available by the CVRSCF service

Resource Name	Resource URI	HTTP Method	Description
3D Model Environment	/experiment-setup/3dmodel/{experiment-id}	POST	Set the 3D model of the environment
Data Traffic Model	/experiment-setup/datatrafficmodel/{experiment-id}	POST	Set the data traffic model
Antenna Characteristic Mobility model	/experiment-setup/antennamobility/{experiment-id}	POST	Set antenna characteristic and mobility model
Channel Measurement	/experiment-status/channel/{experiment-id}	GET	Get delay, phase shift and path loss
Data Flows	/experiment-status/flows/{experiment-id}	GET	Get bit rate, bit error rate and delay

Table 55 - List of attributes of CVRSCF

Attribute Name	Data Type	Description
Bit rate	Float	The amount of transmitted data in a given amount of time. It refers to the speed at which data is transmitted between the UE and the gNodeB, measured in terms of Mbps.
Bit error rate	Float	The number of bit errors per unit time. It represents the probability of errors occurring during data transmission.
Delay	Float	The average latency in milliseconds for traffic.
RIS position	Float	The specific location and orientations of the RIS

Type of propagation model	Float	The type of the mathematical or empirical representations used to describe how EM waves propagate through the environment. It predicts the behaviour of wireless signals as they travel from a transmitter to a receiver.
Tx position	Float	The location of the transmitter
Rx position	Float	The location of the receiver
Phase	Float	The current adjusted direction, focusing, and amplitude of the reflected or transmitted signals.
Ray tracing	Struct	The used propagation model. It simulates the behaviour of EM waves as they interact with objects and surfaces in a 3D environment.
Distance	Float	The current distance between transmitter and receiver, and the distance between the RIS and transmitter and receiver respectively.
Modulation	Struct	Modulation OFDM with 64 QAM with RIS as a used modulation technique.

5.11 N3dsf API

C3DFCF: CONVERGE 3D Simulator Control Function

The CONVERGE 3D Simulator is a central component of the CONVERGE high-level architecture, dedicated to generating electromagnetics augmented 3D environment models specifically for the vision-radio simulator. Nested within the CONVERGE simulator framework, the 3D-S tool specializes in the creation of detailed point-cloud models (and alternatively 3D triangle mesh models), representing both the environment and associated radio behaviour. The 3D simulator efficiently maps radio signals and their propagation models within this geometric space, providing the raytraced paths of electromagnetic signals as the interact with the environment in terms of reflection, diffraction, and optionally refraction and diffuse scattering. The simulator is also equipped to model data flows, sourcing inputs from a variety of sensors such as cameras, radios, and lidars, and then integrating them into its 3D geometry construction processes in an asynchronous manner. In this context, the simulator provides an accurate digital twin of the physical environment, allowing for swift reconfigurations based on real-world data flows, including those from installed cameras and communication equipment. Moreover, digital twin can be enhanced with selected 360° live or recorded camera views with simulated or measured sensor data.

Key Responsibilities of the C3DFCF interface

1. Model Creation: The simulator creates point-cloud models. It uses both direct 3D methods like LiDar and indirect ones such as Camera-based reconstructions. The 3D model of the chamber provides a complete view of it from any position, and has limited support for configuring moving objects and trajectories. The geometrical model incorporates labels for each point and object that considers its characteristics in terms of reflection, diffraction, and absorption, and it offers limited support for refraction and diffuse scattering.
2. Mapping Radio Signals: The simulator uses parallel processing with platforms to compute approximate radio path propagations, based on the location of the receiver and transmitter. It uses different models based on resolution needs, from triangle counts to ray launches, and can model different beamforming approaches.
3. Sensor Data Flow Modelling: The simulator re-models the environment using real data flows and measurements from the physical chamber, using the available camera network and the vision-radio tools (UE, BS, LIS).

Services operations

Experiment Configuration Session: *From Vision Radio Simulator to 3D Environment Modeler*

Table 56 - Resources made available by the C3DFCF service

Resource Name	Resource URI	HTTP Method	Description
Device Proprieties	/device/proprieties/{device-id}	POST	Post number and type of devices, objects cameras and proprieties of the materials
Configuration Status	/configuration-status/{experiment-id}	GET	Get the status of devices, objects, camera configurations and material proprieties
Send Data Streams	/data-session/{experiment-id}	POST	Send camera image/video and radio signals/parameters
Data Streams Status	/data-session/{experiment-id}	GET	Get camera image/video and radio signals/parameter status
Model Session	/model-session/{experiment-id}	POST	Send radio-augmented 3D model and mapped paths in the environment
3D Model Session Status	/model-session/{experiment-id}	GET	Get status of 3D model mapped paths
Visualization Session	/visualization-session/{experiment-id}	POST	3D Model point(s) of view and radio paths
Visualization Session Status	/visualization-session/{experiment-id}	GET	3D Model reception status, Viewpoint setup status and radio path status

Table 57 - List of C3DFCF Attributes

Attribute Name	Data Type	Description
Tx/Rx Locations	Array [Float3]	The 3D location of the transmitters and receivers
Tx/Rx angles	Array [Float4]	The valid angles of transmission and reception “Azimuth, elevation”, the first value corresponding to starting angle, the second to the end angle in counter-clockwise fashion (optional parameter)
3D Environment geometry model (static)	Array [Struct]	A static 3D model of the chamber in “PLY” format containing an array of points representing the basic 3D geometry and visual information. Each point contains as attributes (Float3, 3D_location, int3 RGB_ value).
3D Environment vision-radio model (static)	Array [Struct]	A static 3D model of the chamber in “PLY” format containing an array of points. representing a radio-enhanced 3D model. Each point contains as attributes (Float3, 3D_location, int3 RGB_ value, float3 normal_vector, int label_material, int label_hash, int label_diffraction
3D environment vision-radio model (current)	Array [Struct]	A current 3D model of the chamber in “PLY” format containing an array of points, based on realistic data flows from the chamber. Each point contains as attributes (Float3, 3D_location, int3 RGB_ value, float3 normal_vector, int label_material, int label_hash, int label_diffraction, float time_t)
Paths ready	Int	Expresses when the path computations are ready and possible error codes
Mapped Paths	Array [struct]	Paths in the environment, used signal propagation. It can be stored as a file in CSV format containing and array with the traced paths in the 3D environment. Each path

	[float3]	contains a set of 3D points from each path interaction with the environment objects, expressing the location of the interaction. The first point is the Tx and the last point is the Rx
Data Flow Inputs	Array [Struct]	Inputs from various data sources, detailing how they influence the environment model.
Number of Radio Devices	Integer	The total count of devices utilized in the simulation.
Number of Cameras	Integer	The total count of cameras used to capture images/videos in the simulation.
Number of Objects	Integer	The total count of objects present in the 3D environment.
Material Properties	Array [Struct]	Characteristics of materials in the environment, including reflection and absorption rates for light and radio waves.
Camera Image	Binary/ Data stream	The visual data captured by cameras in the environment.
Radio Signals	Struct	The radio parameters, frequency, and modulation of signals in the environment.
3D Model Configuration	Struct	Details of the 3D environment path computation process, including points of view, discretization size, max number of interactions.
Points of View	Array [Float]	Specific vantage points in the 3D environment from which views or simulations can be observed.
Visualization Status	Enum	The current state of visualization, including setup and reception details.

5.12 Ncvf API

CVCF: CONVERGE Video Control Function

The CVCF is responsible for handling the video camera network that is used to support other CONVERGE tools. It should enable the researcher the possibility to configure and use the video component of the tools (the video cameras installed in the chamber as well as in the different tools: gNB, UE and LIS) and be as flexible as possible.

Key Responsibilities of the CVCF interface

The interface of the CVCF will enable to control the CONVERGE video cameras and real-time vision models in different aspects, comprising the following functionalities:

1. Management of the cameras, including its configuration. The configuration allows the user to associate labels/ids to each camera, define the parameters of its outputs (resolution, framerate, modality, etc), and enable/disable image compression. Additionally, the user can start/stop the streaming or recording (directly on a server) of each camera.

2. Management of the support vision models, which consist of ML models that can be used to extract high-level semantic information from the scene being captured by the cameras. Examples of such models are person/object tracking, 3D posture estimation and scene classification. The models are trained outside the scope of this API and are available as a service. The configuration of each support vision model defines the model endpoint, the supported inputs (e.g. camera/source and type) and the model outputs (type).

Service operations

Table 58 - Resources made available by the CVCF service.

Resource Name	Resource URI	HTTP Method	Description
Cameras start/stop	/control/cameras/{start/stop}/{list of camera_ID}	POST	Start/Stop streaming or recording of cameras based on a list of IDs
Camera configuration	/configuration/cameras/{camera_ID}	POST	Set camera configuration based on its ID
Camera status	/configuration/cameras/{camera_ID}/status	GET	Get the current camera configuration parameters
Support vision models start/stop	/control/vision-model/{start/stop}/{model_ID}	POST	Activate/Deactivate support vision model based on its ID
Support vision models configuration	/configuration/vision-model/{model_ID}	POST	Set support vision model configuration based on its ID
Support vision models status	/configuration/vision-model/{model_ID}/status	GET	Get support vision model configuration based on its ID

Table 59 - List of CVCF Attributes.

Attribute Name	Data Type	Description
CameraID	String	Camera identifier used to identify a given camera
CameraConfiguration	String	Configuration of a camera in JSON format
CameraStatus	Int	Indicator of a camera status (0-idle, 1-streaming, 2-recording)
SupportModelID	String	Support model identifier used to identify a given support model
SupportModelConfiguration	String	Configuration of a support model in JSON format
SuportModelStatus	Boolean	Indicator of the support model service status, which indicates if the service is activated

6 CONCLUSIONS AND FUTURE WORK

This deliverable reports a first definition of access modes and policies for the tools developed by the CONVERGE project following the best practices used in established RIs, such as those used in ESFRI SLICES-RI. Based on the example of the SLICES-RI, the aspects related to access types, access modes, and access policies for the CONVERGE tools were identified and discussed. Since SLICES-RI is still in its preparation phase, experimenter access APIs and policies are still being specified, although some general principles have already been defined such as the consideration of a RESTful API to control and interact with the infrastructure management system. These access policies will be tightly monitored by CONVERGE in order to align the interfaces and prepare a smooth integration of most of its RIs into SLICES-RI in the medium term. The University of Oulu is currently working on the integration of 5GTN RI into the SLICES-SC Web Portal, which will also enable the access to the CONVERGE tools, and this will be a valuable experience for ensuring the long-term integration of CONVERGE into the SLICES-RI.

The CONVERGE service-oriented architecture was presented, and its design aligned with a 5G Network architecture was justified. An example of a CONVERGE use case test session was described with the step-by-step flowchart and procedures required to startup and configure the CONVERGE chamber for an experiment. A set of twelve CONVERGE core APIs were also presented, with associated attributes and parameters, following the 3GPP design principles and guidelines.

The work presented in this document is an ongoing activity pertaining to tasks T1.3 and T1.4 and will continue until M18. The results of these tasks will be later reported in D1.3. The envisioned next steps until then include further work on the definition of the access policies as well as on the definition of the interface APIs, according to the following considerations:

- The access policies will be defined per service and per category of user, using the definition of the services and users produced by the CONVERGE use cases and further refined by each tool definition. Links with SLICES-RI will be further elaborated and exploited. The CONVERGE facilities will position themselves with respect to the SLICES access policies and guidelines.
- Each CONVERGE site (Porto, Oulu and Sophia Antipolis) will define the specific use cases to be implemented and tested in the context of WP3. Based on each use case, D1.3 will specify in detail the modules involved and required APIs. Interaction with WP3 is expected on the definition of the user interface, which at least will provide access to the functionality required to support each of the use cases. D1.3 will detail the interfaces identified in D1.2 using OpenAPI 3.0.0 and following the 3GPP design principles.

REFERENCES

[1] ESFRI Whitepaper 2020: Equipping Europe with infrastructures for ground-breaking research ESFRI's response to the new challenges lying ahead for Europe, 2020, [Online] https://www.esfri.eu/sites/default/files/White_paper_ESFRI-final.pdf

[2] European Commission, Directorate-General for Research and Innovation, European charter of access for research infrastructures – Principles and guidelines for access and related services, Publications Office, 2015, https://data.europa.eu/doi/10.2777/524573

[3] 3GPP TS 38.215, 5G NR Physical layer measurements, 2020, [Online]

https://www.etsi.org/deliver/etsi_ts/138200_138299/138215/16.02.00_60/ts_138215v160200p.pdf

[4] J. Goldsmith, C. Ramsay, D. Northcote, K. W. Barlee, L. H. Crockett and R. W. Stewart, "Control and Visualisation of a Software Defined Radio System on the Xilinx RFSoC Platform Using the PYNQ Framework," in IEEE Access, vol. 8, pp. 129012-129031, 2020, doi: 10.1109/ACCESS.2020.3008954

[5] O-RAN White Paper: “O-RAN: Towards an Open and Smart RAN”, 2018 [Online]

https://assets-global.website-files.com/60b4ffd4ca081979751b5ed2/60e5afb502810a0947b3b9d0_O-RAN%2BWP%2BFInal%2B181017.pdf

ANNEX 1 – Ncgnbf OpenAPI specification

```
openapi: 3.0.0
info:
  title: CONVERGE gNB Control Function
  version: 1.0.0
  description: CONVERGE gNB Control Function
servers:
  - url: 'https://cgnbcf.converge.inesctec.pt/cgnbcf/v1'
security:
  - oAuth2ClientCredentials: []
tags:
  - name: 'Configuration'
  - name: 'Control'
  - name: 'Telemetry'
paths:
  /control/start/{gNB_ID}:
    get:
      description: start CgNB
      operationId: startCgNB
      parameters:
       - name: gNB_ID
         in: path
         schema:
           type: string
           format: uuid
         required: true
       tags:
         -  Control
       responses:
         '200':
           description: 'gNB successfully started'
         '400':
           $ref: 'TS29571_CommonData.yaml#/components/responses/400'
         '401':
           $ref: 'TS29571_CommonData.yaml#/components/responses/401'
         '403':
           $ref: 'TS29571_CommonData.yaml#/components/responses/403'
         '404':
           $ref: 'TS29571_CommonData.yaml#/components/responses/404'
         '500':
           $ref: 'TS29571_CommonData.yaml#/components/responses/500'
    /control/stop/{gNB_ID}:
      get:
        description: stop CgNB
        operationId: stopCgNB
        parameters:
          - name: gNB_ID
            in: path
            schema:
              type: string
              format: uuid
            required: true
        tags:
```

```
        -  Control
      responses:
        '200':
          description: 'gNB successfully stopped'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
/configuration/{gNB_ID}:
  get:
    description: get current configuration applied to gNB
    operationId: getCgNBConfiguration
    parameters:
      - name: gNB_ID
        in: path
        schema:
          type: string
          format: uuid
        required: true
    tags:
      -  Configuration
    responses:
      '200':
        description: 'Returns gNB current runtime configuration'
        content:
          'application/json':
            schema:
              $ref: '#/components/schemas/CgNBConfiguration'
      '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
  post:
    summary: change current configuration applied to gNB
    operationId: writeCgNBConfiguration
    parameters:
      - name: gNB_ID
        in: path
        schema:
          type: string
          format: uuid
        required: true
```

```
  tags:
    - Configuration
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/CgNBConfiguration'
  responses:
    '201':
      description: configuration successfully writen, returns the updated gNB configuration
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/CgNBConfiguration'
    '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
    '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
    '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
    '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
    '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
/telemetry/{gNB_ID}:
  get:
    description: retrieve telemetry for gNB with gNB_ID
    operationId: getCgNBTelemetry
    parameters:
      - name: gNB_ID
        in: path
        schema:
          type: string
          format: uuid
        required: true
    tags:
      -  Telemetry
    responses:
      '200':
        description: 'Returns gNB telemetry'
        content:
          'application/json':
            schema:
              $ref: '#/components/schemas/CgNBTelemetry'
      '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
```

```
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'

components:
  schemas:
    securitySchemes:
      oAuth2ClientCredentials:
        type: oauth2
        flows:
          clientCredentials:
            tokenUrl: 'https://auth.converge.inesctec.pt//oauth2/token'
    responses:
      CgNBConfiguration:
        type: object
        properties:
          O_CU:
            $ref: '#/components/schemas/O_CU'
          O_DU:
            $ref: '#/components/schemas/O_DU'
          O_RU:
            $ref: '#/components/schemas/O_RU'

      RU:
        type: object
        properties:
          bandconfiguration:
            $ref: '#/components/schemas/BandConfiguration'
          bandwidthMHz:
            type: integer
          tddConfiguration:
            type: string
          logicalAntRx:
            type: integer
          logicalAntTx:
            type: integer

      BandConfiguration:
        type: object
        properties:
          band:
            type: integer
          ssbArfcn:
            type: integer
          pointAArfcn:
            type: integer

      CgNBTelemetry:
        type: object
        properties:
          dlCarrierFreqHz:
            type: integer
          ulCarrierFreqHz:
            type: integer
          ueTelemetry:
            type: array
```

```
          items:
            $ref: '#/components/schemas/UeTelemetry'

    UeTelemetry:
      type: object
      properties:
        cqi:
          type: integer
        dlBytes:
          type: integer
        ulBytes:
          type: integer
        dlBler:
          type: number
          format: float
        ulBler:
          type: number
          format: float
        dlMcs:
          type: integer
        ulMcs:
          type: integer
        phr:
          type: integer
        ri:
          type: integer
        rnti:
          type: integer
        rsrp:
          type: number
          format: float
        rsrq:
          type: number
          format: float
        rssi:
          type: number
          format: float
        sinr:
          type: number
          format: float
```

ANNEX 2 – CUE Telemetry Attributes

Table 60 – Full list of CUE Telemetry Attributes.

Attribute Name	Data Type	Description
RSSI	integer	Received Signal Strength Indicator (RSSI) measures the linear average of the total received power observed only per configured OFDM symbol and in the measurement bandwidth, over NRB resource blocks (RB).
RSRP	integer	Reference Signal Received Power (RSRP) measures the linear average power of the resource elements that carry cell-specific reference signals within the considered measurement frequency bandwidth
RSRQ	integer	Reference Signal Received Quality (RSRQ) is used in 5G NR networks to determine the quality of the radio channel based on Synchronization Signals (SSs).
PHR	integer	Power Headroom Report (PHR) reporting refers to a mechanism that allows the gNB to assess the available power margin of a UE. Power Headroom reporting is important for efficient power management and resource allocation in the network.
SINR	integer	Signal to Interference plus Noise Ratio (SINR) is a quality measurement which represents the ratio of the wanted signal power to the interference plus noise power.
CQI	integer	Channel Quality Indicator (CQI) values allow a UE to quantify and report (using either the PUCCH or the PUSCH) its downlink radio channel conditions within a specific Bandwidth Part.
MCS	integer	The Modulation and Coding Scheme (MCS) corresponds to a row within the relevant MCS look-up table, and it is allocated by an algorithm belonging to the gNB.
BLER	float	The Block Error Rate (BLER) is defined as the number of erroneous received code blocks.
Bitrate	float	The bitrate is the total number of bits transferred per second.
RI	integer	The Rank Indicator (RI) is a feedback parameter that is sent by the UE to the gNB to indicate the number of independent spatial streams that can be supported by the MIMO radio channel.
SRS reference signal received power (SRS-RSRP)	float	The linear average of the power contributions (in [W]) of the resource elements carrying sounding reference signals (SRS). SRS RSRP shall be measured over the configured resource elements within the considered measurement frequency bandwidth in the configured measurement time occasions.
CLI Received signal strength indicator (CLI-RSSI)	float	The linear average of the total received power (in [W]) observed only in the configured OFDM symbols of the configured measurement time resource(s), in the configured measurement bandwidth from all sources, including co-channel serving and non-serving cells, adjacent channel interference, thermal noise etc.
PSBCH reference signal	float	The linear average over the power contributions (in [W]) of the resource elements that carry demodulation reference signals associated with physical sidelink broadcast channel (PSBCH).

received power (PSBCH-RSRP)		
PSSCH reference signal received power (PSSCH-RSRP)	float	The linear average over the power contributions (in [W]) of the resource elements of the antenna port(s) that carry demodulation reference signals associated with physical sidelink shared channel (PSSCH), summed over the antenna ports.
PSCCH reference signal received power (PSCCH-RSRP)	float	The linear average over the power contributions (in [W]) of the resource elements that carry demodulation reference signals associated with physical sidelink control channel (PSCCH).
Sidelink received signal strength indicator (SL RSSI)	float	The linear average of the total received power (in [W]) observed in the configured sub-channel in OFDM symbols of a slot configured for PSCCH and PSSCH, starting from the 2nd OFDM symbol.
DL PRS reference signal received power (DL PRS-RSRP)	float	The linear average over the power contributions (in [W]) of the resource elements that carry DL PRS reference signals configured for RSRP measurements within the considered measurement frequency bandwidth.
DL reference signal time difference (DL RSTD)	float	DL reference signal time difference (DL RSTD) is the DL relative timing difference between the Transmission Point (TP) [18] j and the reference TP i, defined as TSubframeRxj – TSubframeRxi, Where: - TSubframeRxj is the time when the UE receives the start of one subframe from TP j. - TSubframeRxi is the time when the UE receives the corresponding start of one subframe from TP I that is closest in time to the subframe received from TP j.
UE Rx – Tx time difference	float	The UE Rx – Tx time difference is defined as TUE-RX – TUE-TX Where: - TUE-RX is the UE received timing of downlink subframe #i from a Transmission Point (TP) defined by the first detected path in time. - TUE-TX is the UE transmit timing of uplink subframe #j that is closest in time to the subframe #i received from the TP.
SS reference signal antenna relative	float	The difference of the average phase of the receive signals on the resource elements that carry secondary synchronization signals (SS) received by the reference individual receiver branch (Rx0) and the average phase of the receive signals on the resource elements that carry secondary synchronization signals (SS)

phase (SS-RSARP)		received by one other individual receiver branch (Rx1 ... Rxn). The measurement time resource(s) for SS-RSARP are confined within SS/PBCH Block Measurement Time Configuration (SMTC) window duration.
DL PRS reference signal received path power (DL PRS-RSRPP)	float	The power of the linear average of the channel response at the i-th path delay of the resource elements that carry DL PRS signal configured for the measurement, where DL PRS-RSRPP for the 1st path delay is the power contribution corresponding to the first detected path in time.
Sidelink PRS reference signal received power (SL PRS-RSRP)	float	The linear average over the power contributions (in W) of the resource elements that carry SL PRS reference signals configured for RSRP measurements within the considered measurement frequency bandwidth.
Sidelink PRS reference signal received path power (SL PRS-RSRPP)	float	The power of the linear average of the channel response at the i-th path delay of the resource elements that carry SL PRS configured for the measurement, where SL PRS-RSRPP for the 1st path delay is the power contribution corresponding to the first detected path in time.
Sidelink relative time of arrival (TSL-RTOA)	float	The SL relative time of arrival ($T_{SL\text{-}RTOA}$) is defined as the beginning time of SL subframe # *i* containing SL PRS received from a UE, relative to the relative time of arrival (RTOA) reference time. The SL RTOA reference time is defined as , where - is the nominal beginning time of SFN 0 or DFN 0, provided by SFN and DFN initialization time, respectively. - where and are the SFN or DFN and the subframe number of the SL PRS, respectively.
Sidelink angle of arrival (SL AoA)	float	The estimated azimuth angle and vertical angle of a transmitting UE with respect to a reference direction, wherein the reference direction is defined: - In the global coordinate system (GCS), wherein estimated azimuth angle is measured relative to geographical North and is positive in a counter-clockwise direction and estimated vertical angle is measured relative to zenith and positive to horizontal direction - In the local coordinate system (LCS), wherein estimated azimuth angle is measured relative to x-axis of LCS and positive in a counter-clockwise direction and estimated vertical angle is measured relative to z-axis of LCS and positive to x-y plane direction. The bearing, downtilt and slant angles of LCS are defined according to TS 38.901.
Sidelink Rx – Tx	float	The SL Rx – Tx time difference at a UE is defined as $T_{UE\text{-}RX} - T_{UE\text{-}TX}$

time difference		Where: -$T_{UE\text{-}RX}$ is the UE received timing of sidelink subframe #*i* from a transmitting UE, defined by the first detected path in time. -If the UE reports the transmission timestamp of a SL PRS, $T_{UE\text{-}TX}$ is the transmit timing of the sidelink subframe #*j* of the SL PRS of the UE. Otherwise, $T_{UE\text{-}TX}$ is the transmit timing of the UE of sidelink subframe #*j* that is closest in time to the subframe #i received from the transmitting UE. -The same antenna reference point is used for receiver and transmitter for the Rx-Tx time difference measurement.
Sidelink reference signal time difference (SL RSTD)	float	The SL relative timing difference between the UE *j* and the reference UE *i*, defined as , where: - is the time when the UE receives the start of one subframe from UE *j* - is the time when the UE receives the corresponding start of one subframe from UE *i* that is closest in time to the subframe received from UE *j*
DL reference signal carrier phase (DL RSCP)	float	The phase of the channel response at the *i*-th path delay derived from the resource elements carrying DL PRS configured for the measurement.
DL reference signal carrier phase difference (DL RSCPD)	float	The difference of DL RSCPs measured from DL PRS transmitted in a DL PFL from the transmission point (TP) *j* and the reference TP *i*. If UE reports RSCPD measurements together with RSTD measurements in a measurement report element, the reference TP for RSCPD is the same as the reference TP reported for RSTD.
Sidelink PRS received signal strength indicator (SL PRS-RSSI)	float	The linear average of the total received power (in [W]) observed in OFDM symbols of SL-PRS and PSCCH of slots configured for PSCCH and SL-PRS.
Time domain channel property (TDCP)	float	The wideband normalized correlation between two CSI-RS transmission occasions, corresponding to CSI-RS resourece(s) from *NZP-CSI-RS-ResourceSet(s)* configured with higher layer parameter *trs-info*, that are separated by D_n symbols or slots, depending on the configuration, where D_n is the n-th delay configured value among [*placeHolderForRrcParameter-D*] configured delay values $\{D_1, \ldots, D_Y\}$ and Y is number of configured delay values.
UE Rx – Tx time difference offset	integer	The actual index difference between subframe #*j* and subframe #*i* of the subframes used for the UE Rx – Tx time difference measurement as defined in Clause 5.1.30, where uplink subframe #*j* is the closest in time to the DL subframe #*i* received from a transmission point (TP).

DL timing drift	float	The DL timing estimated to be shifted due to Doppler over the service link associated with the UE Rx-Tx time difference measurement period.