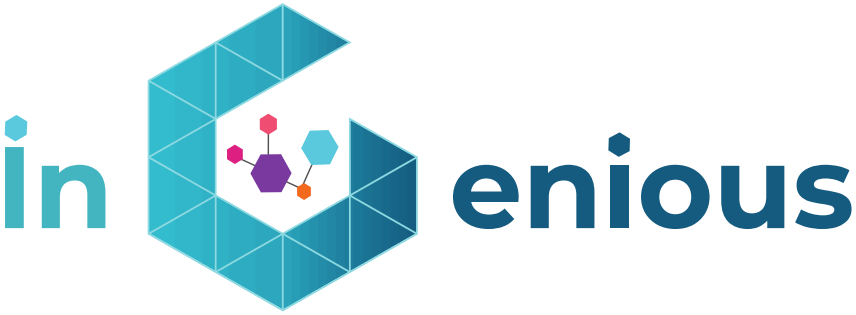
NEXT GENERATION INTERNET

Grant Agreement No.: 957216
Call: H2020-ICT-2018-2020

Topic: ICT-56-2020
Type of action: RIA



D4.2 Smart NR and NG-RAN IoT designs

Revision: v.1.0

Work package	WP4
Task	Task 4.1
Due date	31/01/2022
Submission date	31/01/2022
Deliverable lead	UPV
Version	1.0
Editors	Nuria Molner (UPV), Javier Renart (UPV)
Authors	Javier Renart (UPV), Nuria Molner (UPV), José Luis Cárcel (FV), Pietro Piscione (NXW), Giacomo Bernini (NXW), Christos Politis (SES), Juan Jose Garrido Serrato (SES), Fabian Buehrmann (SES), Miguel Cantero (5CMM), Manuel Fuentes (5CMM), David Martín-Sacristán (5CMM), Roberto Bomfin (TUD), Ivo Bizon (TUD), Ahmad Nimr (TUD), Cristina Escribano (NOK), Joe Cahill (iDR).
Reviewers	Carsten Weinhold (BI), Nuria Molner (UPV), Roberto Bomfin (TUD), Giacomo Bernini (NXW), Jose Costa-Requena (CMC)

Abstract	This document describes the smart RAN approach of INGENIOUS to develop a heterogeneous connectivity architecture by aggregating different existing and forthcoming IoT technologies. In short, this deliverable targets at explaining the technical developments of the main innovations and state of the art technologies applied to internet-of-things (IoT) networks and their relationship to the project's use cases.
Keywords	Radio Access Network, Smart NR, Internet-of-things, NG-RAN IoT

Document Revision History

Version	Date	Description of change	List of contributor(s)
V1.0	31/01/2022	EC version	See author list

Disclaimer

This iNGENIOUS D4.2 deliverable is not yet approved nor rejected, neither financially nor content-wise by the European Commission. The approval/rejection decision of work and resources will take place at the Mid-Term Review Meeting planned in June 2022, after the monitoring process involving experts has come to an end.

The information, documentation and figures available in this deliverable are written by the "Next-Generation IoT solutions for the universal supply chain" (iNGENIOUS) project's consortium under EC grant agreement 957216 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

Copyright notice

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:		**R***
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to iNGENIOUS project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Executive Summary

This document describes the technical work done on Smart New Radio (NR) and Next-Generation (NG) IoT designs, which is part of the heterogeneous connectivity solution that iNGENIOUS is developing. This deliverable focuses on the main Radio Access Technology (RAT) innovations and state-of-the-art technologies being used and developed, mapping them to the corresponding use cases of iNGENIOUS.

These innovations cover Smart NR technologies, which include Flexible PHY/MAC to enable applications with diverging needs, and AI/ML for Radio Access Network (RAN) to improve the decision-making and performance of current open networks. NG-RAN IoT is also addressed, including a New Radio Modem focused on IoT devices and a Smart IoT Gateway, which supports multiple RATs on one device. Furthermore, Satellite connectivity is described to allow the connection of IoT devices to the network where terrestrial coverage is not possible as well as a mmW port deployment, focusing on the hardware components used.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Table of Contents

Co-funded by the Horizon 2020 Framework Programme of the European Union

List of figures

Co-funded by the Horizon 2020 Framework Programme of the European Union

Co-funded by the Horizon 2020
Framework Programme of the European Union

List of tables

Co-funded by the Horizon 2020
Framework Programme of the European Union

Abbreviations

5GC	Fifth Generation Core
AN	Access Network
API	Application Programming Interface
BS	Base Station
CN	Core Network
EPC	Evolved Packet Core
FIFO	First In First Out
FPGA	Field Programmable Gate Array
GFDM	Generalized Frequency Division Multiplexing
GW	GateWay
HMI	Human Machine Interface
M2M	Machine-To-Machine
MANO	Management and Network Orchestration
MCS	Modulation And Coding Scheme
NI	National Instruments
NN	Neural Networks
NR	New Radio
NRM	Network Resource Model
NSMF	Network Slice Management Function
NSSI	Network Sub Slice Instances
OFDM	Ortogonal Frequency-Division Multiplexing
QPSK	Phase-Shift Keying
RAN	Radio Access Network
RIC	Radio Intelligent Controller
SDR	Software-Defined Radio
TDD	Time Division Duplexing
TN	Transport Network
UE	User Equipment
WAN	Wide Area Network

Co-funded by the Horizon 2020 Framework Programme of the European Union

1 Introduction

1.1 Objective of this Deliverable

This deliverable aims to describe the main innovations and technical progress on the iNGENIOUS *Smart NR components and the NG-RAN IoT* within Task 4.1 on work package (WP) 4. Several use cases in iNGENIOUS have different network requirements; thus, several radio access technologies (RATs) are being developed or deployed to fulfil the requirements. In this case, D4.2 presents the progress on the RAN domain and part of the UE domain within the overall iNGENIOUS architecture in Figure 1. Six leading technologies are addressed including innovations and state-of-the-art technologies, i.e., Flexible PHY/MAC, AI/ML for RAN, NR Modem, Smart IoT Gateway, Satellite Backhaul and Direct Access technologies following both 3GPP and non-3GPP standards.

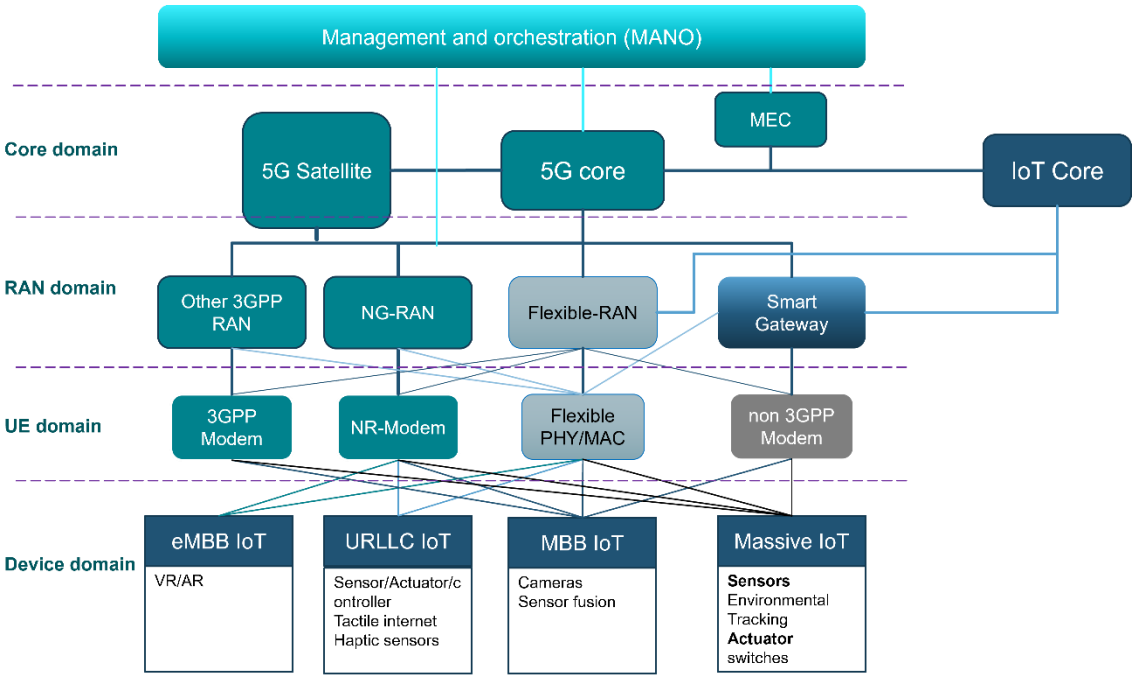


Figure 1: Overall iNGENIOUS network architecture

1.2 Role of T4.1 in iNGENIOUS

Task 4.1 aims at improving the network support of various radio technologies to be used in the next generation IoT networks and will support several use cases plotted in Figure 2 with diverse traffic requirements that are further detailed in Deliverable *D2.1 Use cases, KPIs and requirements* [1]

Co-funded by the Horizon 2020 Framework Programme of the European Union

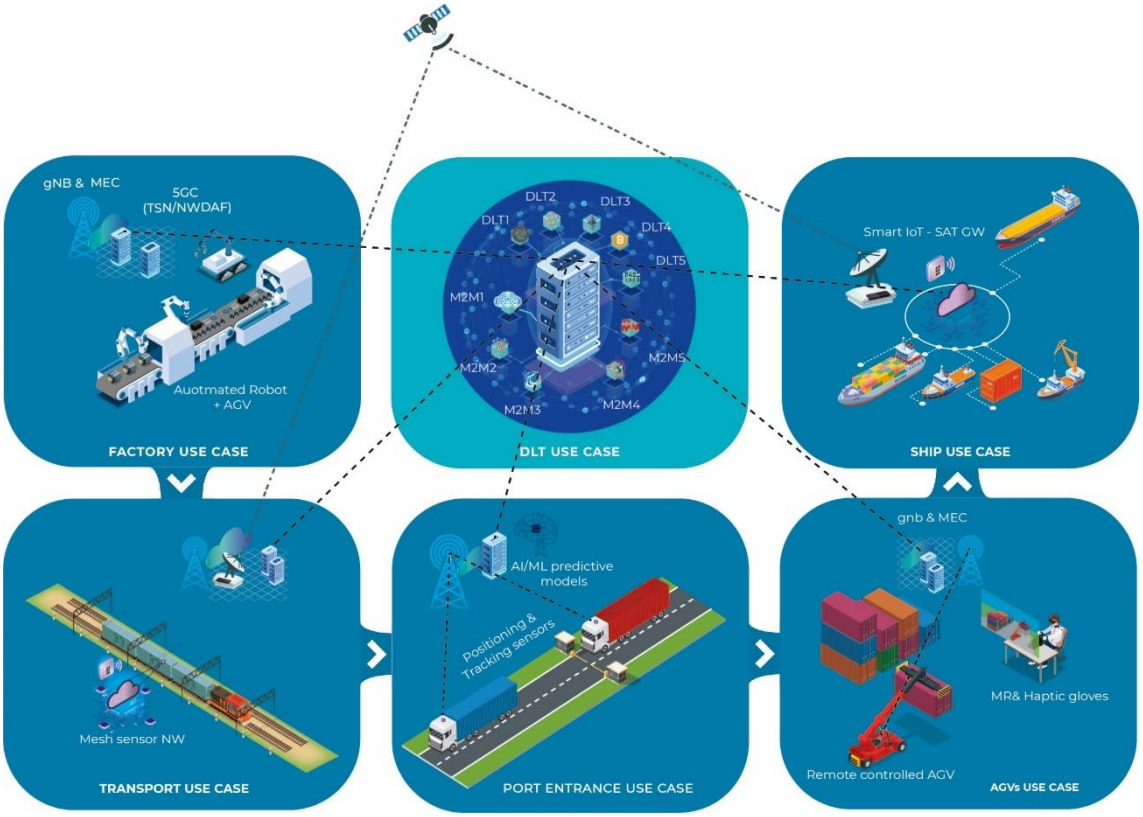


Figure 2: iNGENIOUS next generation supply chain use cases

T4.1 is the first task inside WP4, and it aims to achieve two of the main WP4 objectives:

- to design and develop a flexible RAN architecture to support the forthcoming IoT network requirements
- to design new functional blocks to support smart and semiautonomous decisions based on ML algorithms

These objectives include multi-technology aspects like NB-IoT, 5G or LoRaWAN via the smart IoT gateway and 5G Modem that are contributed by iNGENIOUS partners. Furthermore, T4.1 investigates efficient and flexible PHY/MAC schemes to develop dynamic network deployments with Software Defined Radio (SDR) devices. On the RAN side, AI/ML models are being introduced into RAN controllers to optimise resource management and increase the capabilities of next generation IoT networks. Real deployments are also being assessed, introducing as an innovation a mmW deployment in the port of Valencia, which leads the path to global deployments. Finally, coverage in remote areas is being addressed thanks to a Satellite Backhaul which will interconnect ships and containers when on the sea.

Co-funded by the Horizon 2020
Framework Programme of the European Union

2 Smart NR

The introduction of AI/ML models and flexible architectures in networks are a natural step in the development of next generation IoT networks. In this section, the technical development is detailed on Flexible PHY/MAC schemes to introduce dynamic network deployments with SDR devices. This section looks at AI/ML for RAN, focusing on the introduction of ML models in open networks, aiming towards an increased efficiency and performance in resource management.

2.1 Flexible PHY/MAC

In this section, we present the flexible PHY/MAC implementation of TUD with emphasis on the multiple access protocol and integration with the 5G higher layers. The goal of this implementation is to develop a dynamic network deployment with Software Defined Radio (SDR) devices, that enables applications with diverging needs. TUD's PHY implementation is based on the generalized frequency division multiplexing (GFDM) waveform, that can be flexibly configured depending on the users' needs. The implementation has been developed with the National Instruments (NI) SDR platforms and consists of a host implementation for control and a field programmable gate array (FPGA) implementation for real-time signal processing. Due to fast signal processing capability of the SDR platforms, this implementation enables low-latency applications such as remote control of automated guided vehicles (AGVs), it also enables applications with medium throughput such as video stream.

As described in the Deliverable *D4.1 Multi-technologies network for IoT* [2], the flexible PHY will be applied in the Factory use case (UC) where different applications will be deployed in a dynamic environment. For example, applications that require relatively high throughput, e.g., video stream will have more resources allocated than applications requiring less data; another example is the remote control of factory robots that will require low latency.

2.1.1 DYNAMIC MULTIPLE ACCESS PROTOCOL

The goals of the dynamic multiple access are twofold, namely, i) we target at designing a multiple access mechanism to be implemented with the SDR platforms available at the TUD's testbed, and ii) we aim at designing a multiple access protocol that can dynamically manage the wireless resources of different users depending on the application needs. We take advantage of the flexibility of the TUD's PHY that will be reported in deliverable *D3.2 Proposals for next generation of connected IoT modules* to accomplish these goals. For this purpose, our design approach consists of a centralized controlled access scheme based on a polling mechanism. This approach has been chosen because it allows the base station (BS) to have a full control of the traffic such that the PHY parameters can be adjusted according to the application needs, where the network can dynamically allocate resources in the time domain. To implement this MAC protocol, we have defined a control channel that transmits control data to coordinate the medium usage. In the Deliverable

Co-funded by the Horizon 2020 Framework Programme of the European Union

D3.2, we provide a detailed block diagram of the PHY aspects of this implementation.

Before explaining the details about our MAC implementation, we first need to define the frame structure in Figure 3. The frame is composed of three parts, namely, i) preamble, ii) control and iii) payload. The frame is explained below:

Preamble: This is a signal composed of two appended Zadoff-Chu sequences with 64 samples each. This signal is used for synchronization and channel estimation.

Control: From the multiuser perspective, what is important to know is that the BS uses the control channel to inform which user should decode the payload, and which user should transmit its data in the next uplink slot. More details about the control channel are given in section 2.1.2.

Payload: The application data is sent via the payload channel, which is configured depending on the application. The PHY configuration of the payload signal is sent via the control channel, as shown in section 2.1.2. Details on possible configurations of the payload will be reported in Deliverable D3.2.

Figure 3: Frame design

The multiple access scheme is depicted in Figure 4, where a scenario with 3 users is shown. Figure 4 shows that the control and payload channels are sent separately. On the left side of this figure, one can observe that all nodes decode the control channel to check if they are the recipient of the payload and/or they should transmit in the next payload time slot. In this case, user 1 (UE1) is the node that should use the payload slots. The right side of this figure depicts this situation over a timeline. The information flows as follows: 1) the BS sends the control information as depicted in the frame of Figure 4, and all users decode its content, 2) in this example, the payload has UE1 as destination, therefore, all users but UE1 ignore the payload, 3) the subsequent uplink time slot is also designated to UE1, and the remaining users wait until the next downlink signal and the cycle starts again. Thus, this system implements a time division duplexing (TDD) mechanism to transmit downlink and uplink data.

Co-funded by the Horizon 2020 Framework Programme of the European Union

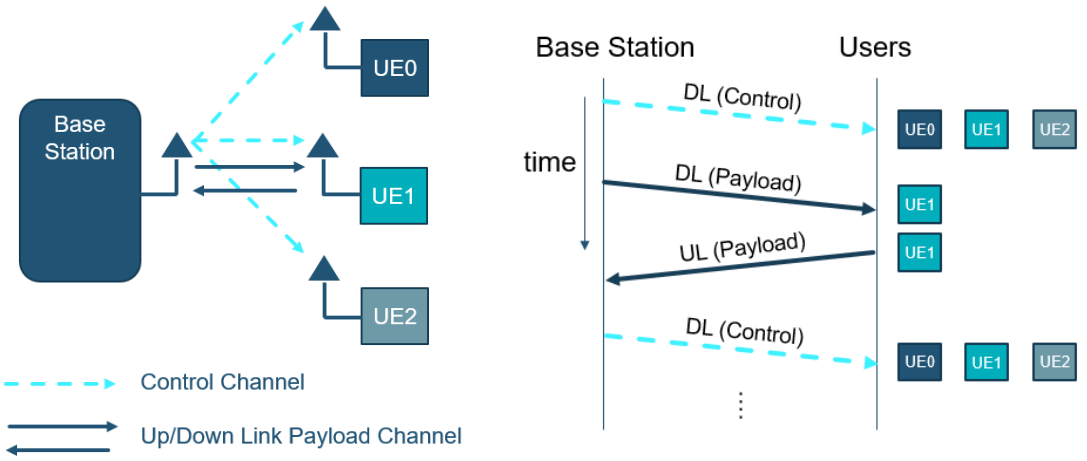


Figure 4: Multiuser overview

The hardware implementation of the MAC design is given in Figure 5. The PHY related blocks such as *scrambler*, *CRC*, *(de)encoder*, *symbol/resource (de)mapper*, *GFDM (de)modulator*, *channel estimation* and *channel equalization* will be explained in more details in the deliverable 3.2. This deliverable focuses on the MAC related aspects. As one can see, in both transmitter and receiver modules, there are two parallel chains implemented on the FPGA, one for the control channel on top and one for the payload below. As we already discussed in Figure 3, the control channel has a fixed configuration, meaning that the blocks are fixed in the implementation. On the other hand, the payload processing chain can be reconfigured at the runtime depending on the UE configuration. At the transmitter side, this configuration is done by the *TX MAC Ctrl*, that is responsible for configuring the PHY blocks. Additionally, the *MAC MUX* concatenates the control channel and payload signals according to the frame structure of Figure 3. At the receiver side, after the synchronization, the *MAC DEMUX* block separates the control channel and payload signals, where they are applied to their respective receiver chains. Firstly, the control data is decoded, which contains the user ID and the payload signal configuration. If the RX user ID matches the user which is processing this data, then the RX PHY is configured, and the payload is decoded. Lastly, to interface the incoming and outcoming data of the FPGA and host, the data is transferred using first-in, first-out (FIFO) elements, namely, the host to target (H2T) FIFO and target to host (T2H) FIFO. The FIFOs are used for the buffering and the flow control between the host software and the FPGA. The baseband signal processing is deployed on the FPGA.

Co-funded by the Horizon 2020 Framework Programme of the European Union

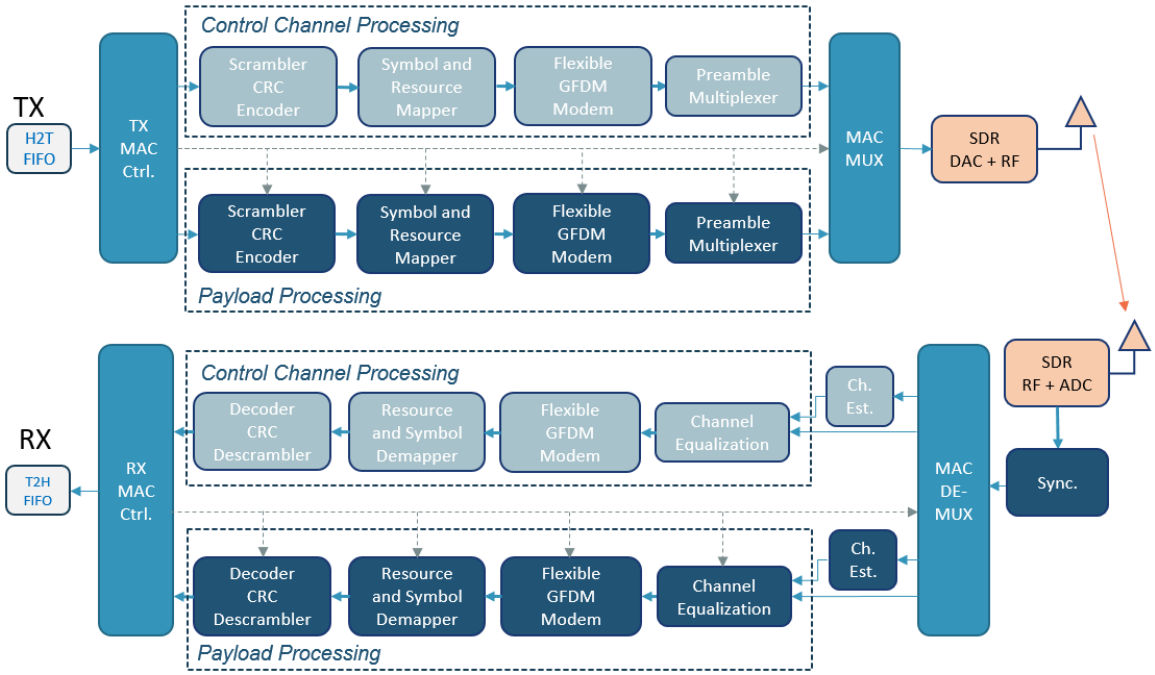


Figure 5: Block diagram MAC control implementation

2.1.2 CONTROL CHANNEL

The control channel is used only in the downlink for setting up the transmission parameters and to define the transmission schedule. Due to the fixed parametrization of the control channel, it has a separate processing chain compared to the payload processing as seen in Figure 5. Due to the fixed size of the operations, the hardware needed for the implementation is significantly reduced in comparison with the payload processing. Also, we highlight that the control channel implementation described in the following section is a particular design example, because this concept can be generalized. For instance, in general it is possible to support a larger number of users than the one specified, and the control channel can be made longer.

PHY Configuration of the control channel:

The PHY configuration of the control channel is given in Table 1. We utilize an orthogonal frequency-division multiplexing (OFDM) configuration with $N = 64$ subcarriers, where $N_{pilots} = 4$ carriers are used to transmit pilot symbols and $N_{data} = 48$ carriers are used to transmit data. For the data symbols, we employ the quaternary phase-shift keying (QPSK) because it is the most robust modulation. With this configuration, a number of 32 information bits can be transmitted. Notice that with QPSK (2 bits per symbol) and 48 data symbols, the control channel can transport 96 bits in total, which is calculated based on the amount of information bits as: [32 (inf. bits) + 8 (CRC bits)] x 2 (encoder)+ 16 (tail) = 96 bits.

Table 1: PHY configuration of control channel

Waveform	N	N_{on}	N_{pilots}	N_{data}	QAM Order	Information bits
OFDM	64	52	4	48	4 (QPSK)	32

Co-funded by the Horizon 2020 Framework Programme of the European Union

Implementation of PHY for control channel:

The PHY implementation of the control channel has multiple advantages in terms of complexity. Firstly, the size of the computational blocks, e.g. the FFTs, of the FPGA are reduced in size compared to the flexible blocks due to the fixed PHY configuration. Secondly, the length of the control channel was selected so that it matches the length of the chirps. Therefore, the channel estimation can be directly used for the equalization bypassing the conversion to the time domain for the zero-padding. Thirdly, OFDM is used as the waveform which can be implemented with relatively low complexity.

Structure of bits of the control channel:

The 32 bits of the control channel contain all the information that the user UE needs to receive as packets from the BS and packets to transmit to the BS. The first 12 bits of the control channel are used for the downlink control information and the following 12 bits for the uplink control information. The remaining 8 bits are left for future use.

The downlink and uplink control information contain four different parameters. The identifier (ID) of the targeted user or BS, the used modulation and coding scheme (MCS), the remaining configuration of the PHY and the number of frames which should be received or should be transmitted. The structure of the control channel is visualized in Table 2Figure 5. In the following section, the exact definitions of the different parameters of the control channel are provided.

Table 2: Structure of the control channel

	Parameter	Length [bits]	Comment
Downlink Control Information	User ID in DL	4	Identify users to receive the downlink data
	MCS DL	2	Identify MCS in downlink: (QPSK, 16QAM, 64QAM)
	PHY config DL	4	PHY configuration in downlink
	N payload DL	2	Number of continuous frames in downlink
Uplink Control Information	User ID in UL	4	Identify users in for uplink data
	MCS UL	2	Identify MCS in uplink: (QPSK, 16QAM, 64QAM)
	PHY config UL	4	PHY configuration in uplink
	N payload UL	2	Number of continuous frames in uplink
	Total	24	8 bits left for future use

User ID:

The user ID is used in the downlink such that a specific UE can be targeted by the BS as the recipient of the transmitted frame(s). Other way around, the UE can target the BS in the uplink. Four bits are used for the user ID meaning that 16 different IDs are available as shown in Table 3. A broadcast ID is defined as 1111 such that the frame(s) are transmitted to all recipients simultaneously.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Furthermore, the ID 0000 is reserved if no transmission should occur in the downlink or uplink. This enables a data stream in only one direction. The remaining 14 IDs are utilized for addressing the UEs.

Table 3: User ID

Bits	ID	Comment
0000	No transmission	e.g., there is no payload in the downlink, but the BS expect data in the subsequent uplink frame
0001-1110	UE Address	Can support up to 14 users
1111	Broadcast	Message designated to all users

Modulation and Coding Scheme (MCS):

Two bits are used to define the MCS of the PHY. However, only the modulation can be changed, and the code rate is fixed to $r = 0.5$. One of four different modulations can be selected as shown in Table 4. Currently, the 8PSK modulation is not implemented but could be included if needed.

Table 4: Modulation and Coding Scheme

Bits	Modulation	Code rate
00	QPSK	0.5
01	8PSK	0.5
10	16QAM	0.5
11	64QAM	0.5

PHY configuration:

16 configurations of the PHY can be defined with four bits. Several parameters such as the number of subcarriers, the number of active subcarriers, the CRC length, and others are combined into one configuration. In addition, 7 different PHY configurations have been tested with over the air transmission will be reported in the Deliverable 3.2, which are given in Table 5. Basically, the configurations with high N_{data} are suitable for transmissions with high payload size, and vice-versa. The K and M parameters are part of the GFDM configuration and denote number of subcarriers and subsymbols.

Table 5: Tested PHY configurations

N	K	M	N_{on}	N_{pilots}	N_{data}
2048	2048	1	1792	8	1784
2048	128	16	1680	8	1672
1024	1024	1	896	8	888
1024	64	16	810	8	802
512	512	1	448	8	440
512	32	16	360	8	352
64	64	1	52	4	48

N Payload:

Two bits are used for the number of continuous frames. Therefore, multiple frames can be transmitted without the additional control overhead. The configurations are shown in Table 6.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Table 6: Number of frames

Bits	Number of frames
00	1
01	2
10	4
11	16

2.1.3 INTEGRATION INTO HIGHER LAYERS

Another aspect of interest of our MAC implementation is its integration with higher layers such as the 5G Core (5GC) and the management and orchestration (MANO) layers, which were reported in the deliverable D4.1 [2]. In this manner, iNGENIOUS PHY/MAC implementation becomes 5G enabled on the network side, while the GFDM based PHY will be reported in deliverable 3.2 remains customized.

This activity is in its initial phase at the time of writing this deliverable. Therefore, in this section the planned approach for this integration is described. The idea is to use the *UERANSIM* [3] RAN simulator as the bridge between the flexible PHY/MAC and the 5G Core. The *UERANSIM* is an open-source state-of-the-art 5G UE and RAN (gNodeB) implementation, which can emulate a 5G UE and a BS. Since this emulator already has the interfaces with the 5G Core, the plan is to take advantage of this interface and route the user plane data through it. In this way, the UEs connected to the flexible PHY/MAC BS will be connected to the 5G network.

The planned integration is depicted in Figure 6. The flexible PHY/MAC is interfaced with the 5G Core via the *UERANSIM* emulator including the data plane and control plane. The MANO platform is on top of the diagram coordinating the management plane. The RAN NSSMF block of the MANO layer will be connected to the flexible PHY/MAC BS, where communication resources can be allocated depending on specific application metrics. The interconnection of 5G Core and MANO will be reported in detail in deliverables *D4.3 Core network automation design for 5G-IoT* and *D4.4 Service orchestration at the edge*, which respectively deal with these layers.

Co-funded by the Horizon 2020 Framework Programme of the European Union

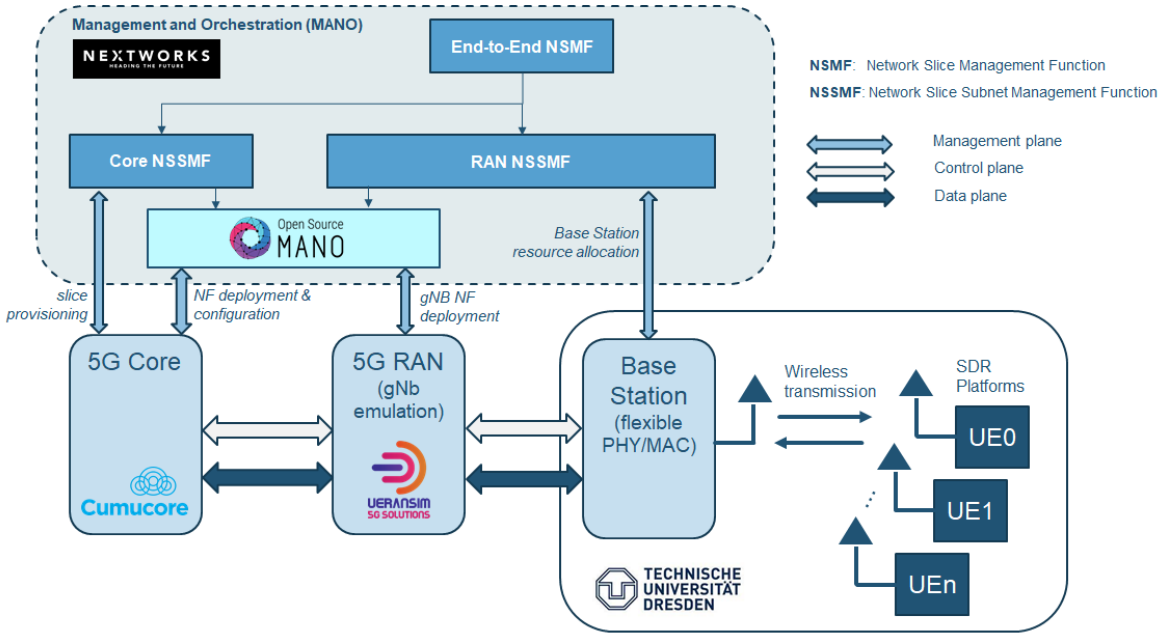


Figure 6: Planned integration of Flexible PHY/MAC with 5GCore and MANO layer

2.2 AI/ML for RAN

The RAN architecture is moving from a monolithic approach in traditional networks, to a flexible, interoperable, and virtualised architecture. The O-RAN Alliance is working together with hundreds of vendors and operators towards this approach, offering to the community open-source software to test and develop new functionalities of the O-RAN platform. The O-RAN alliance introduced the RAN Intelligent Controller (RIC), which is divided into the near-real-time and the non-real-time components. The RIC will be able to take intelligent decisions to improve the RAN's performance and efficiency, through dedicated applications called xApps in the near-RT RIC and rApps in the non-RT RIC. These applications will make use of the data gathered from UEs and BSs to analyse with machine learning (ML) tools the behaviour of the access network in order to be able to predict for example possible failures and take decisions in order to fix them in advance. Examples of this type of applications are given in the following sections. In particular, in iNGENIOUS project the near-real time RIC has been deployed with emphasis to the A1 interface and two current available applications have been tested using the data provided by the O-RAN open-source community [4].

2.2.1 ANOMALY DETECTION APPLICATION CASE

This application case has the objective of detecting and correcting anomalies in the UEs, using signal quality metrics and throughput values. Once an anomaly is detected, the corresponding xApps look for a neighbour cell that can provide a better service to the UE, triggering a handover if a better cell is found.

The full workflow of this application case can be found in Figure 7. This application case will use the E2Simulator, which will send Cell and UE reports to act as an O-RAN Data Unit (O-DU). In addition, the KPIMON xAPP shown in

Co-funded by the Horizon 2020 Framework Programme of the European Union

Figure 7 will retrieve this data and store it in the near-RT RIC database, ready for future consumption by other xApps.

Currently, the greyed-out components in Figure 7 are not implemented yet by the O-RAN community, but the application case can still be demonstrated using the available xApps, which are the main pillar in this scenario.

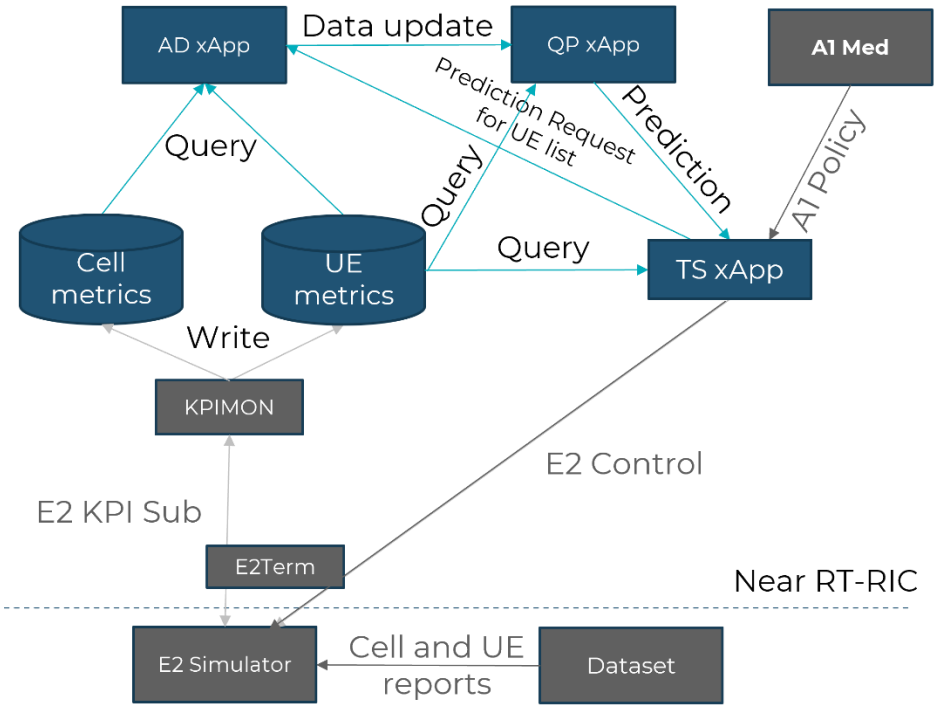


Figure 7: Anomaly Detection workflow

2.2.1.1 Anomaly Detection xApp (AD)

The Anomaly Detection (AD) xApp is in charge of two main tasks: populating the database (DB), which stores the User and Cell reports, and detecting anomalies. While the E2 Simulator is not available, this xApp populates the near-RT RIC database with the Cell and UE report classified dataset, which was simulated by the O-RAN alliance using the Viavi E2 simulator.

UEPDCPBytesDL	UEPDCPBytesUL	UEPRBUsageDL	UEPRBUsageUL	S_RSRP	S_RSRQ	S_SINR	N1_RSRP	N1_RSRQ
290000	122000	25	10	-48	-34	25	-53	-64
280000	123000	25	10	-44	-30	25	-49	-60
280000	123000	25	10	-45	-31	25	-50	-61

N1_SINR	N2_RSRP	N2_RSRQ	N2_SINR	UEID	ServingCellID	N1	N2	MeasTimestampRF
20	-68	-94	17	12345	555002	6E+05	6E+05	05/11/2020 13:44
20	-64	-90	17	12345	555002	6E+05	6E+05	05/11/2020 13:44
20	-65	-91	17	12345	555002	6E+05	6E+05	05/11/2020 13:44

Figure 8: Extract of UE report dataset.

The second step is to train an ML model using the dataset previously stored in the DB. In this case, the ML model used is a Random Forest Classifier, which provides simplicity and transparency to the decisions taken by the xApp.

Finally, the xApp starts consuming the data from the DB, simulating that new data is received every 10ms. All the reports go through the Random Forest Classifier model which finds the anomalous UEs and its degradation cause

Co-funded by the Horizon 2020 Framework Programme of the European Union

(RSRP, RSRQ, SINR, Throughput). Once an anomalous UE is found, the UE ID is sent to the Traffic Steering xApp, which will continue the process, as shown in Figure 9.

```
[INFO] Sending Anomalous UE to TS
[INFO] Message to TS: message sent Successfully
[INFO] Received acknowldgement from TS (TS_ANOMALY_ACK): {'payload': b'[{"du-id": 1001, "ue-id":
    "Car-4", "measTimeStampRf": 1639141800965, "Degradation": "RSRP RSSINR"}]', 'payload length':
    99, 'message type': 30004, 'subscription id': -1, 'transaction id': b'
    b24e136e59ba11eca725ba9788a94735', 'message state': 0, 'message status': 'RMR_OK', 'payload
    max size': 3136, 'meid': b'', 'message source': 'service-ricxapp-trafficxapp-rmr.ricxapp:4560'
    , 'errno': 0}
```

Figure 9: AD log file

2.2.1.2 Traffic Steering xApp (TS)

The Traffic Steering (TS) xApp acts as intermediary between the AD and QoS Prediction (QP) xAPP defined in Section 2.2.1.3. The TS xApp also takes handover decisions. First, TS receives an anomalous UE ID from the QP xApp. Then, TS xApp asks for a throughput prediction to the QP xApp, which will be explained in Section 2.2.1.3. When the prediction is received, the TS xApp checks if a neighbour cell can provide a better throughput to the UE and sends a handover request from the serving cell to the new cell. At this moment, this final handover message is sent but not received by any entity, as it is still not implemented in the application case. Figure 10 shows the execution of TS xApp.

```
[INFO] AD Callback got a message, type=30003, length=99
[INFO] Payload is [{"du-id": 1001, "ue-id": "Car-4", "measTimeStampRf": 1639141799935,
    "Degradation": "RSRP RSSINR"}]
[INFO] Prediction Request length=30, payload={"UEPredictionSet": ["Car-4"]}
[DEBUG] Prediction Request sent=
[DEBUG] Prediction Request sent=30000
[INFO] Prediction Callback got a message, type=30002, length=118
[INFO] Payload is {"Car-4": {"c1/B2": [2, 2], "c1/N77": [0, 0], "c4/B13": [3421, 3421], "c2/B13":
    [4118, 4118], "c3/B13": [1063, 1063]}}
[DEBUG] before for
[DEBUG] inside for
[DEBUG] curr_throughput=1063, highest_throughput=0
[DEBUG] inside for
[DEBUG] curr_throughput=4118, highest_throughput=1063
[DEBUG] inside for
[DEBUG] curr_throughput=3421, highest_throughput=4118
[DEBUG] inside for
[DEBUG] curr_throughput=2, highest_throughput=4118
[DEBUG] inside for
[DEBUG] curr_throughput=0, highest_throughput=4118
[DEBUG] before if highest_throughput=4118, serving_cell_throughput=2
[INFO] Sending a HandOff CONTROL message to "http://localhost:5000/api/echo"
[INFO] HandOff request is {
    "command": "HandOff",
    "seqNo": 2,
    "ue": "Car-4",
    "fromCell": "c1/B2",
    "toCell": "c2/B13",
    "timestamp": "Fri Dec 10 13:11:23 2021",
    "reason": "HandOff Control Request from TS xApp",
    "ttl": 10
}
```

Figure 10: TS log file

2.2.1.3 QoS Prediction xApp (QP)

The QoS Prediction (QP) xApp predicts the throughput based on signal quality parameters (RSRP, RSRQ, SINR) and cell load data. It receives a UE ID from the

Co-funded by the Horizon 2020 Framework Programme of the European Union

TS xApp, and then retrieves the cell and UE reports from the current serving cell and all the neighbour cells, which are seven cells in this scenario.

Currently, the throughput is forecasted using a Time Series Vector Autoregression model, which models the Cell load over time and considers past values to predict the result. The main goal here is to improve this model, introducing ML algorithms which will be trained with the cell and UE dataset, and will be further explained in the next subsection.


```
{"ts": 1639141882682, "crit": "DEBUG", "id": "ricxappframe.xapp_frame", "mdc": {}, "msg": "run:
    invoking msg handler on type 30000"}
{"ts": 1639141882682, "crit": "DEBUG", "id": "qp.main", "mdc": {}, "msg": "predict handler
    received payload b'{\"UEPredictionSet\": [\"Car-4\"]}'"}
{"ts": 1639141883873, "crit": "DEBUG", "id": "qp.main", "mdc": {}, "msg": "Sending message to ts
    : {\"Car-4\": {\"c1/B2\": [2, 2], \"c1/N77\": [0, 0], \"c4/B13\": [3421, 3421], \"c2/B13\": [
    4118, 4118], \"c3/B13\": [1063, 1063]}}"}
{"ts": 1639141883873, "crit": "DEBUG", "id": "qp.main", "mdc": {}, "msg": "predict handler: sent
    message successfully"}
```


Figure 11: QP log file

2.2.2 ML MODEL TESTING AND RESULTS

Viavi's dataset has been extracted from the AD xApp, and a Python environment has been prepared to work with it and test new predictive models. The initial approach was to use a Random Forest regression, as it is the most appropriate considering the lack complexity of the dataset. However, due to the efficiency of the model for this type of datasets and the fact that the dataset is simulated with low complexity models, we encountered that the Random Forest regression learn completely the model behind the simulated data and thus, gives an accuracy of 100% in the prediction. This confirms the model can easily predict the simulation behaviour, but it is not relevant to learn from the data in order to extract useful knowledge. Hence, we can learn that the low-complexity simulated data is not enough to extract fruitful knowledge since the model that produces the simulation can also be learnt with the appropriate ML tools.

However, datasets with complex simulations will be used in the near future, which will certainly require to add more sophisticated ML models, like neural networks (NN). Thus, with the ambition of using the models that will be more relevant for the more realistic datasets, we've also tested a two-layer neural network in the previously defined dataset, using the Pytorch framework, to compare the behaviour between this and the previous model, expecting to observe that NN does not achieve the 100% accuracy and a decrease of error occurs along iterations.

The metric used to measure the error is the root-mean-square error (RMSE), which is a measure of the difference between values predicted by a model and the real values, and it is the most common error metric used to measure the performance of NNs. The NN model has been trained for 100 epochs in order to have a more extended view of the evolution of the error behaviour, and the RMSE can be seen per epoch on Figure 12. The model behaves as expected, having higher RMSE values on the first epochs, and reducing it gradually while it is being trained, until reaching an RMSE of 0.008 on the training dataset, and of 0.013 on the validation dataset on epoch 100.

Co-funded by the Horizon 2020 Framework Programme of the European Union

From the analysis done we can conclude that using low-complexity models to simulate the data is not useful to predict the real behaviour of traffic, as they don't add the variability that happens in real environments and thus can be easily learnt with the proper ML tools. Also, the ML models that will be used for high-complexity simulated datasets have been tested over the low complexity dataset to analyse the evolution of error over learning epochs. Further analysis is required to extract more relevant knowledge from the QP xApp data.

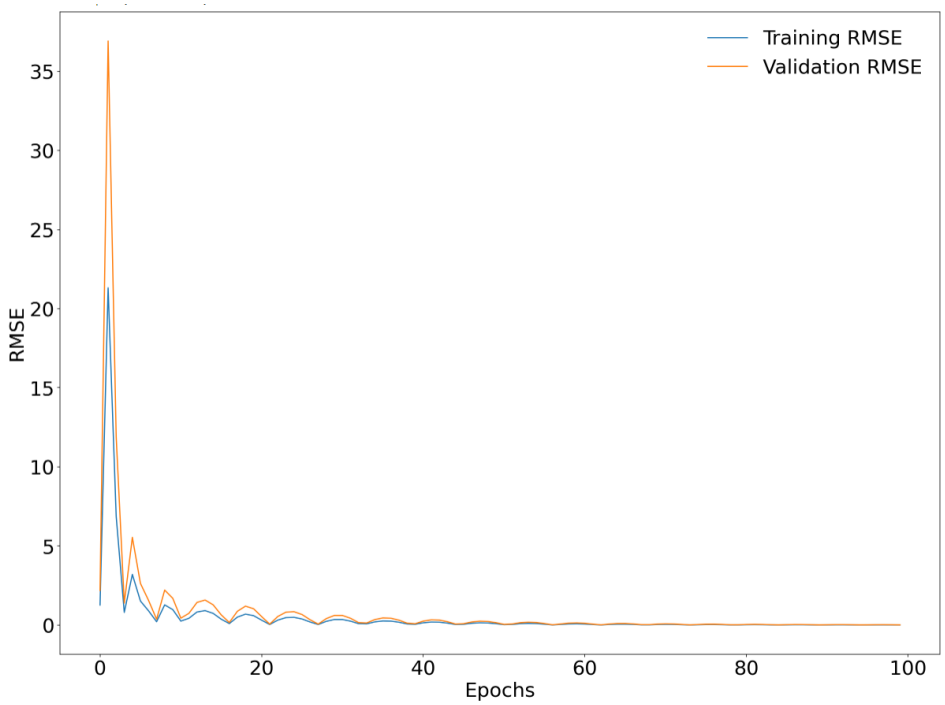


Figure 12: Training and validation error

2.2.3 INITIAL O-RAN INTEGRATION WITH MANO LAYER

Network slicing is one of the key features of the 5G network architecture. Multiple services with different requirements can be deployed simultaneously on top of the same physical infrastructure, providing isolation, security, transparency while guaranteeing specific per-service performances.

To achieve this, the high-level network slice requirements (e.g., in terms of quality-of-service constraints, coverage, density of users, etc.) must be translated into detailed and technical requirements across multiple and different domains of the network, such as the RAN, edge, 5G Core, transport network. In general, this means to translate, decompose, and manage the lifecycle of a network slice in multiple Network Sub Slice Instances (NSSI), each of them tailored for a specific domain of the network.

The cross-layer MANO designed and developed in iNGENIOUS follows an architectural approach capable of orchestrating network slices across heterogenous domains. Figure 13 depicts a high-level architectural diagram for the management of end-to-end network slices in the 5G network. This architectural diagram follows the management and orchestration architectural framework specification provided by 3GPP TS 28.533 in its Release 17 [5] Starting from the top, the Network Slice Management Function

Co-funded by the Horizon 2020 Framework Programme of the European Union

(NSMF) is responsible for decomposing the end-to-end Network Slice into different Network Sub-Slices. The NSMF indeed interacts with the Network Slice Sub Slice Management Functions (NSSMFs) for the lifecycle management of the Network Sub Slice Instances across different network domains: Access Network (AN), Core Network (CN), Transport Network (TN) and where needed specific private or public cloud domains where specific service applications are deployed.

Indeed, each NSSMF is thus customized for managing its technology or network domain and coordinating the provisioning of Network Sub Slice instances. Each NSSMF interacts with per-domain resource controllers that take care to allocate and configure slice resources (e.g., network functions, radio resources, virtual network links, etc.) in their own domain.

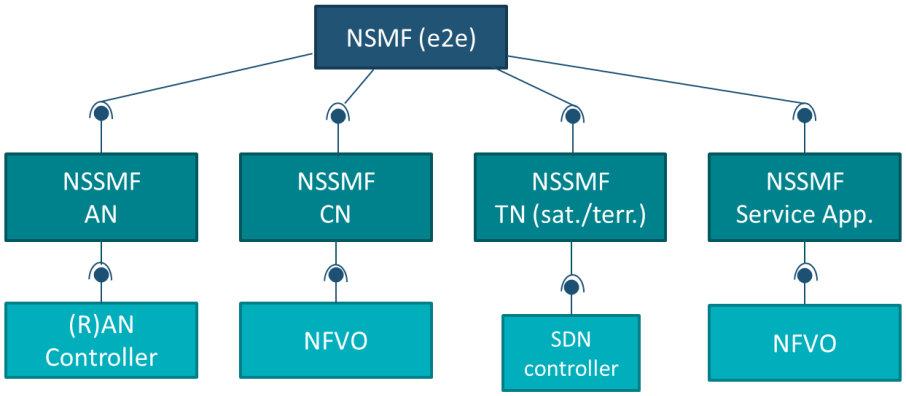


Figure 13: Architectural diagram of Network Slice Management in 5G network

Figure 14 depicts the high-level architectural diagram of a preliminary version of the NSSMF dedicated to the management of AN domains controlled by O-RAN near-RT RICs. It exposes dedicated NSSI Management APIs for creating, modifying and de-allocating NSSIs and, through a specific translation logic, it creates, modifies, and deletes O-RAN A1 QoS related policies into the Near RT RIC.

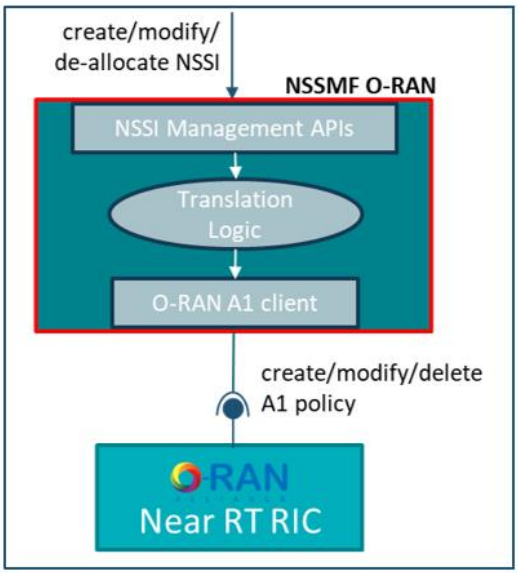


Figure 14: NSSMF O-RAN high-level architectural diagram

Co-funded by the Horizon 2020 Framework Programme of the European Union

The O-RAN Alliance and in particular its Working Group 2 has defined the O-RAN.WG2.A1GAP-v02.02 technical specification for the A1 interface policy general aspect and principles. In general, an A1 policy is composed of a scope and one or more policy statements: the former defines to which context the policy statement will be applied (e.g., UEs, QoS flow, or cells), the latter specifies the goals to the near-RT RIC. In this context, the O-RAN Alliance has defined also the O-RAN.WG2.A1AP-v03.01 and O-RAN.WG2.A1TD-v01.00 technical specifications for A1 interface. The former defines the REST API and thus the endpoints to be queried for managing the policies and policy types, while the latter defines the data models and type for the A1 interface. Both specifications have been widely used for designing, developing, and testing the preliminary version of the O-RAN NSSMF for interacting with the near-RT RIC, especially the O-RAN A1 client.

For what concerns the integration between the iNGENIOUS cross-layer MANO and O-RAN, some initial activities have been started. More specifically, an instance of the O-RAN NSSMF provided by NXW can interact with a O-RAN near-RT RIC provided by UPV through the A1 interface, sending the request to create a policy into the near-RT RIC. In this early integration phase, the O-RAN NSSMF exposes the NSSI Management API for the creation and instantiation of a Network Sub Slice at the O-RAN level. The Network Sub-Slice Instantiation workflow requires checking, processing, and eventually sending the request, through the A1 interface, to the near-RT RIC. Currently, the request to the NSSMF REST controller is manually issued and will be automated in the next steps of the integration and cross-layer MANO developments. More details on the cross-layer MANO perspective of this integration, e.g. in terms of the interaction with the NSMF for providing the end-to-end network slices, will be reported in deliverable D4.4.

Figure 15 depicts the workflow implemented by the O-RAN NSSMF for creating and instantiating a Network Sub Slice. As an initial step, an NSSI identifier is created within the NSSMF and returned as a response. Then, the instantiation of the NSSI is triggered. The instantiation consists of translating the Network Sub Slice requirements, i.e., the Slice Profiles aligned with the 3GPP 5G Network Resource Model (NRM) into one or more A1 QoS policy creation requests sent to the near-RT RIC.

Co-funded by the Horizon 2020 Framework Programme of the European Union

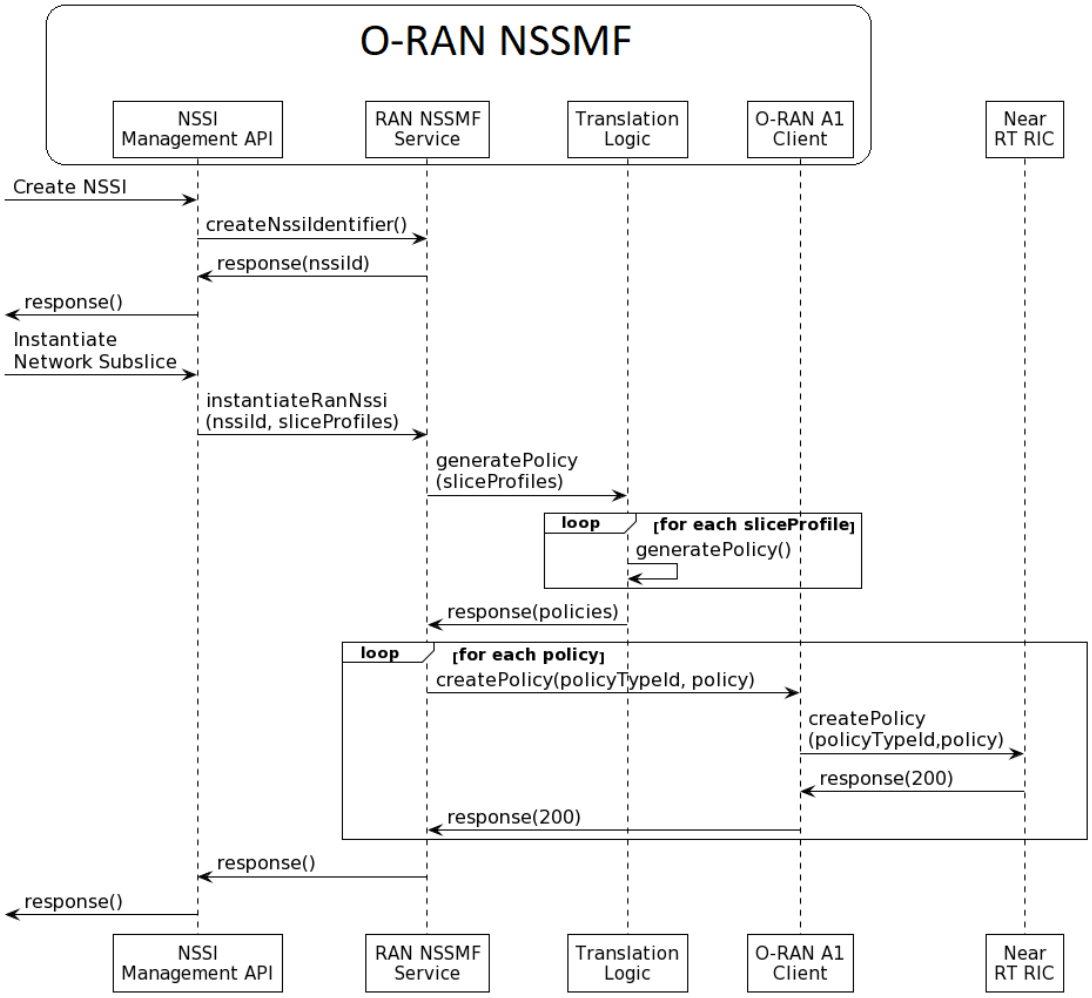


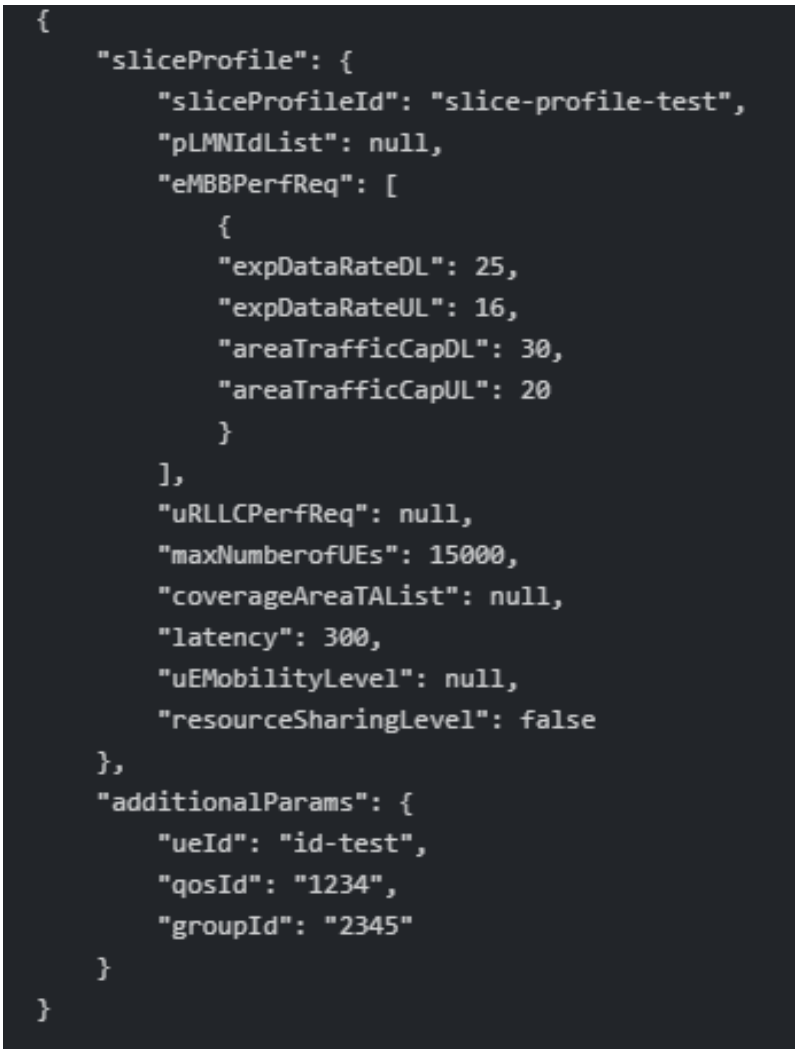
Figure 15: Workflow of on Network Sub-Slice Instantiation O-RAN

Figure 16 depicts an example of a Network Sub-Slice instantiation request payload. The Network Sub Slice contains only one slice profile, specifying the expected bitrate in upload and download and the maximum number of UEs.

The slice profile is then translated into a corresponding policy that contains the following information about the single A1 policy QoS objective:

- Guaranteed flow bit rate (gfbr) and Maximum flow bitrate (mfbr): the average bitrate in a fixed time window and the maximum bitrate, respectively.
- Priority level: the lowest Priority Level value corresponds to the highest priority.
- Packet delay budget (pdb): the maximum allowed latency in milliseconds.

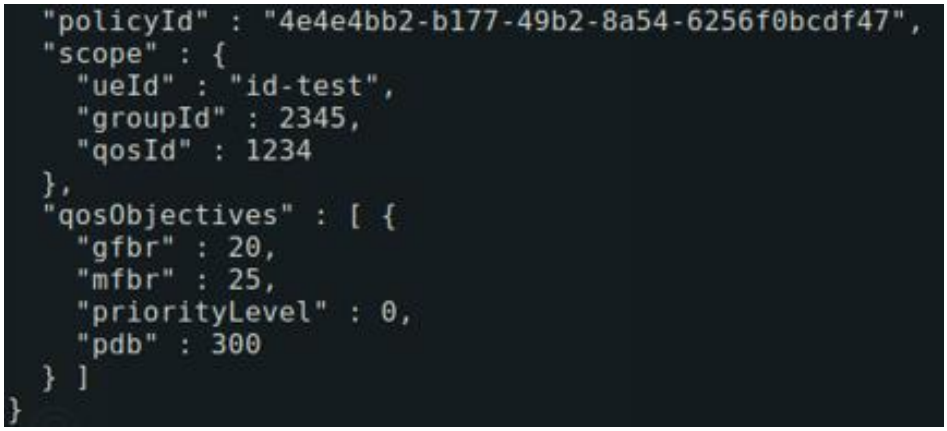
Co-funded by the Horizon 2020 Framework Programme of the European Union



```
{
    "sliceProfile": {
        "sliceProfileId": "slice-profile-test",
        "pLMNIdList": null,
        "eMBBPerfReq": [
            {
            "expDataRateDL": 25,
            "expDataRateUL": 16,
            "areaTrafficCapDL": 30,
            "areaTrafficCapUL": 20
            }
        ],
        "uRLLCPerfReq": null,
        "maxNumberofUEs": 15000,
        "coverageAreaTAList": null,
        "latency": 300,
        "uEMobilityLevel": null,
        "resourceSharingLevel": false
    },
    "additionalParams": {
        "ueId": "id-test",
        "qosId": "1234",
        "groupId": "2345"
    }
}
```


Figure 16: Example of slice profile into Network Sub-Slice instantiation request

The translated policy is depicted in Figure 17.



```
"policyId" : "4e4e4bb2-b177-49b2-8a54-6256f0bcdf47",
"scope" : {
  "ueId" : "id-test",
  "groupId" : 2345,
  "qosId" : 1234
},
"qosObjectives" : [ {
  "gfbr" : 20,
  "mfbr" : 25,
  "priorityLevel" : 0,
  "pdb" : 300
} ]
}
```


Figure 17: QoS Objective Policy as result of translation of slice requirement

Figure 18 shows the logs of the O-RAN NSSMF: the first two rows refer to the creation of a new NSSI. Then, a request to instantiate the network sub slice is received. This request is translated into the related slice profile and then sent to the near-RT RIC. Finally, the policy is correctly created within the near-RT RIC, which answers with a 202 HTTP response code.

Co-funded by the Horizon 2020 Framework Programme of the European Union

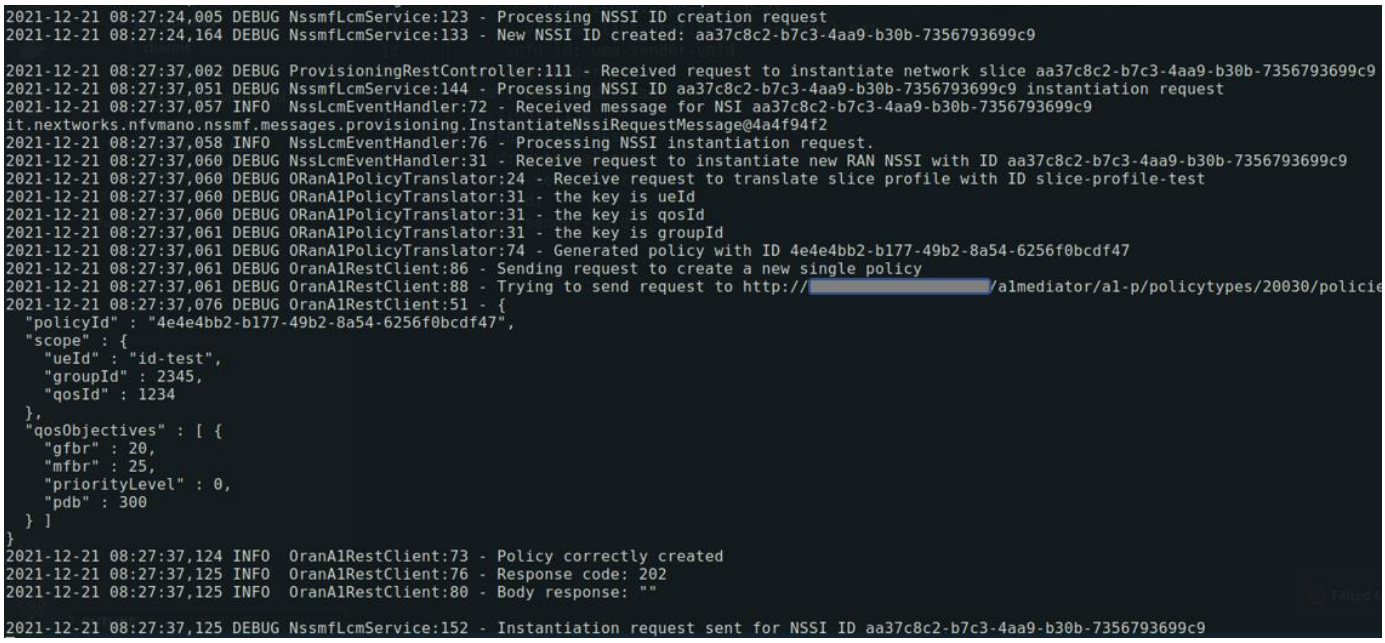


Figure 18: O-RAN NSSMF Logs containing the Network Sub Slice creation and instantiation request

At this point, the O-RAN NSSMF has a NSSI identified as *aa37c8c2-b7c3-4aa9-b30b-7356793699c9* which corresponds to the policy on the near-RT RIC with identifier *4e4e4bb2-b177-49b2-8a54-6256f0bcdf7*.

Figure 19 depicts the information retrieved from the near-RT RIC by querying the specific O-RAN A1 policy object endpoint through an HTTP GET request. This means that the policy has been successfully created into the near-RT RIC.

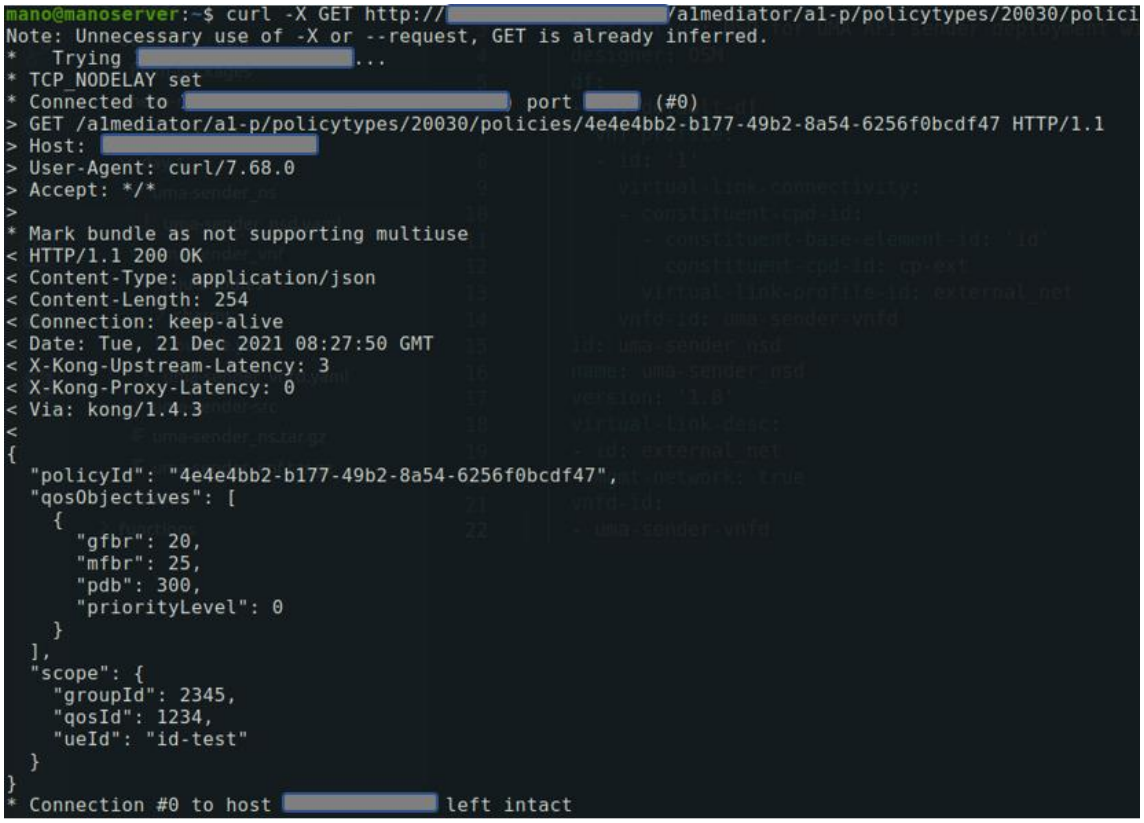


Figure 19: Policy manually retrieved from Near RT RIC

Co-funded by the Horizon 2020 Framework Programme of the European Union

During the integration experiments, it was found that the A1 standard policy type used for QoS objectives was not actually available in the opensource O-RAN near-RT RIC deployed at UPV. To overcome this issue, the policy type was first created manually to enable the O-RAN near-RT RIC to accept policy objects based on such policy type. This manual process will be automated as part of the future work in the context of a full integration of the cross-layer MANO with O-RAN. Moreover, as part of the next steps, further tests will be carried out to validate the use of different Slice Profiles and NSSI requests at the O-RAN NSSMF. Additional policy types beyond the currently used QoS related one will be also investigated.

Co-funded by the Horizon 2020
Framework Programme of the European Union

3 NG-RAN IoT

The overall goal of iNGENIOUS is to develop the next generation IoT networks and devices. This section focuses on the next generation of RAN for IoT devices, introducing an NR modem developed to interconnect current and future IoT devices, a Smart IoT Gateway which supports different RAT technologies and IoT devices, as well as a satellite access on remote areas that leverages on a Satellite Backhaul.

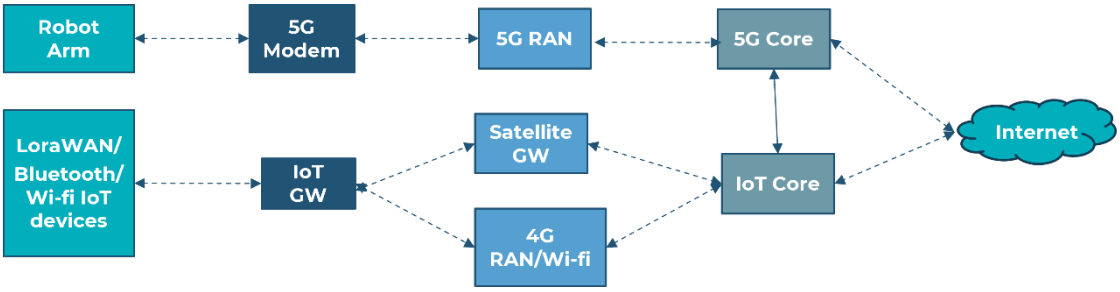


Figure 20: NG-RAN IoT scenario using the developed technologies.

Three main technologies are addressed on this section and shown on Figure 20, the first one being a NR modem for demanding scenarios like high quality video streaming on AGVs or Cranes. Second, a Smart IoT Gateway has been developed, and supports multiple RAT technologies to connect different IoT devices on a variety of applications. Finally, a satellite backhaul is presented, to provide internet access on the sea, where terrestrial networks can't offer coverage.

3.1 NR Modem

The 5G modem to be used in iNGENIOUS project is currently validated. In this section a description of the component and validation results in three different scenarios are shown. Future steps are focused on improving the modem with Rel-16 implementation (currently it works with Rel-15) and hardware upgrades such as reducing the size and adding the SIM card insertion functionality.

3.1.1 COMPONENT DESCRIPTION

5G modems are User Equipment (UE) devices that are used to connect specific vertical components, such as robots, sensors, cameras, or AGVs to the 5G network. 5G modems permit these devices to be connected via Ethernet and to communicate with the network wirelessly. Figure 21 shows a simplified scheme of a 5G architecture, where modems are included as part of the end user devices.

Co-funded by the Horizon 2020 Framework Programme of the European Union

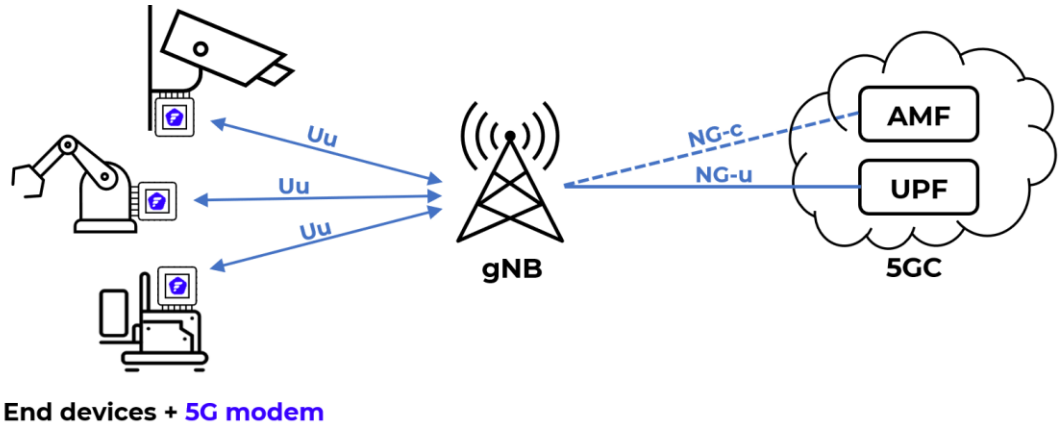


Figure 21: Role of 5G modems in the network architecture

Within the context of iNGENIOUS, 5G modems provided by Fivecomm (5CMM) will be integrated into the complete end-to-end system. These modems are both 5G NSA (non-standalone) and SA (standalone) compatible and work in mid-band frequencies. The modem is a compact and flexible solution, whose characteristics will be described in the future deliverable D3.2 of iNGENIOUS. Figure 22 shows a picture of the current version of the 5G modem. The modem is formed by a 5G chipset module connected to a Raspberry Pi 4 via USB. The Raspberry Pi 4 provides routing, connecting the device via Ethernet to the 5G module, which is then used to communicate with the 5G network and receive/transmit data.

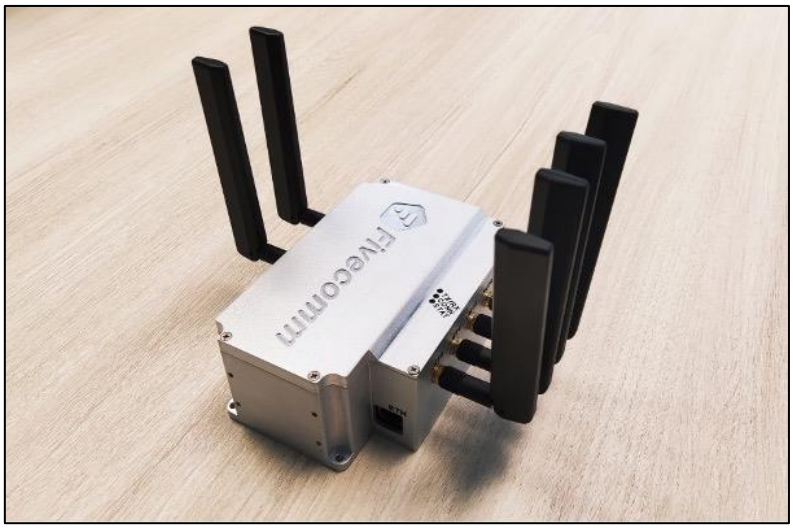


Figure 22: Fivecomm's 5G modem

A first prototype is available for demos and trials. This modem has been already integrated and validated in several testbed facilities, including Cumucore premises as part of the work towards the trials of iNGENIOUS. More details are provided in the following section.

3.1.2 INTEGRATION AND VALIDATION ACTIVITIES

The Fivecomm 5G modem has been validated against several 5G networks, testing its performance in different conditions and under various scenarios. Some of the most relevant results obtained are gathered in this section.

Co-funded by the Horizon 2020 Framework Programme of the European Union

3.1.2.1 5G NPN scenario at UPV

This first activities for the modem validation were performed in 5CMM premises. The gNB was the Amarisoft Callbox Ultimate from UPV. 5G SA was evaluated in the n78 band and with a SINR of 20 dB. The results obtained were a DL throughput of 250 Mbps, UL throughput of 40-50 Mbps and a 5G Latency of 15 ms on average.

It is worth to notice that, throughput values tested with Amarisoft could be lower than expected due to the bad quality of the channel.

Table 7: Validation in 5G NPN scenario at UPV (Amarisoft Callbox Ultimate)

Scenario	Characteristics	5G latency	DL throughput	UL throughput
Amarisoft (Fivecomm premises)	SA n78 band BW = 100 MHz	~15 ms	~250 Mbps	~40-50 Mbps

3.1.2.2 5G commercial network

The second scenario was in UPV campus, using the commercial 5G NSA network from Orange, also using the n78 band. Figure 23 shows the location of the 5G antenna and the modem (Fivecomm premises) inside the campus.

Figure 23: 5G commercial network and modem location at UPV campus

The throughput measurements were carried out in two different ways: with an external server, and between 2 modems. The results obtained were 400-500 Mbps in DL and 110 Mbps in UL for the first case. 110-115 Mbps were obtained in the second case, where the communications between two modems is limited by the UL. Also, a RTT latency of ~20 ms was measured. Different antenna configurations have been tested. The results are summarized in Table 8.

Tests with the 5G commercial network are NSA, what could make carrier aggregation with LTE, improving the performance. The bandwidth described in this case corresponds to the 5G bandwidth.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Table 8: Tests results with the commercial network

Antenna configuration	Speed test (Mbps)		UE to UE TCP (Mbps)	Network latency (ms)
	DL	UL		
Config 1	515.14	111.5	115	20
Config 2	412.67	109.59	110	22
Config 3	426.07	110.63	109	21
Config 4	426.07	110.63	110	21
Config 5	472.54	-	115	19
Config 6	472.54	110.76	115	19
Config 7	447.66	110.6	115	20
Config 8	405.21	110.41	115	20
Config 9	375.29	111.64	115	21

Table 9: Validation in 5G commercial network

Scenario	Characteristics	5G latency	DL throughput	UL throughput
5G commercial network (UPV campus)	NSA n78 band BW = 50 MHz	~20 ms	~400-500 Mbps	~115 Mbps

3.1.2.3 Cumucore premises

Once the modem was validated in Valencia, the third integration and validation activity took place in CMC premises in Finland. This setup used the 5G modem and the 5G network using CMC 5G Core. For the tests, n78 band with a bandwidth of 100 MHz and a nrCellType of 4DL2UL was used. Two different tests were evaluated, obtaining a DL throughput of 420 Mbps in one, and a DL throughput of 343 Mbps, UL throughput of 133 Mbps and a ping of 10 ms in the other one. These tests are shown in Figure 24.

Co-funded by the Horizon 2020 Framework Programme of the European Union

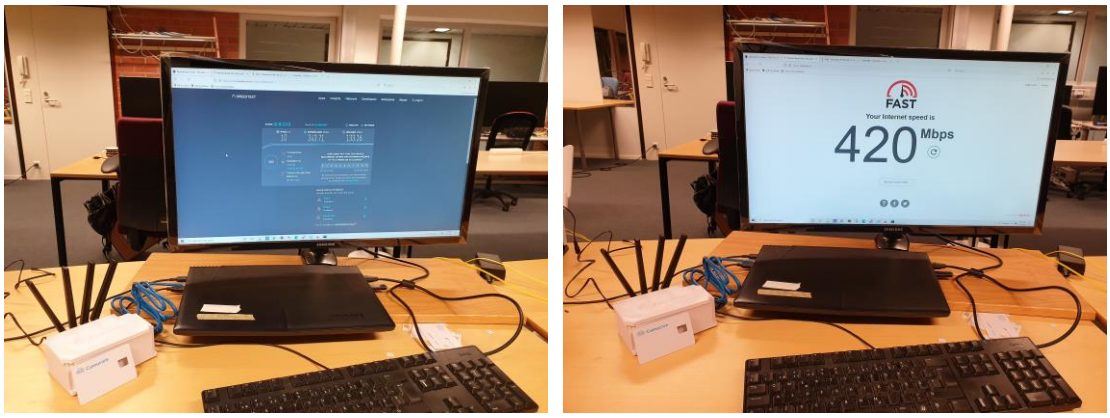


Figure 24: Latency and throughput performance tests done at Cumucore premises

Table 10: Validation in Cumucore premises

Scenario	Characteristics	5G latency	DL throughput	UL throughput
CMC premises	SA n78 band BW = 100 MHz 4DL2UL	~10 ms	~343-420 Mbps	~133 Mbps

3.2 Smart IoT Gateway

The Smart IoT Gateway (GW) is the system element responsible for the appropriate routing and sorting of sensor data, coming from one or more sensor networks to higher layer data consolidation services and machine-to-machine (M2M) platforms. To achieve this, the Smart IoT GW has been divided into functional blocks with an implementation that isolates them as independent containers, following a philosophy quite similar to micro-services architecture. The blocks are presented in Figure 25.

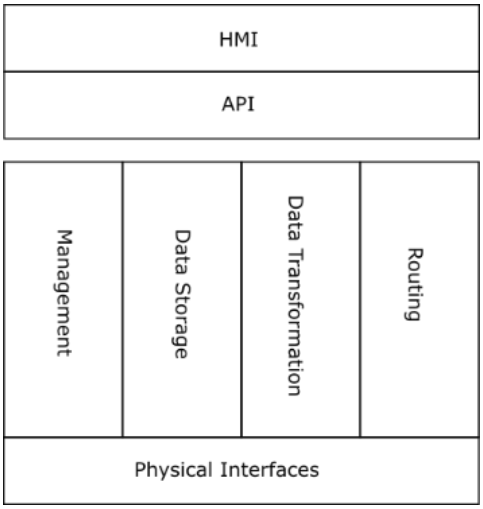


Figure 25: Smart IoT GW block decomposition

Co-funded by the Horizon 2020 Framework Programme of the European Union

Connected to the different physical interfaces the following services are available in the Smart IoT GW:

- *Management*: Management of the GW that includes the current status of the device, internal configuration, local logs and firmware update. It will include also some limited control over the subsystems.
- *Data storage*: Databases (time series and relational). All data coming from the sensor space is time stamped and stored in a time series database that has the capability of performing some operations such as data aggregation, statistical calculations, and time consolidation (i.e., decimate values and down-sample the data).
- *Data transformation*: Message translation between formats and protocols. This does not include just direct or indirect mapping of fields, but also this service will be responsible of encapsulating related messages that came from independent sources or to perform compression when possible.
- *Routing*: Incoming message routing through the gateway to a different interface, following predefined rules, respecting priorities in a secure way.

An Application Programming Interface (API) is available for external interoperability and automation. A simple Human-Machine Interface (HMI) is provided for local data monitoring.

The Smart IoT GW is designed and developed across both WP4 and WP5, hence, we have to mention that in this document, we will not further describe the Smart IoT GW data transformation and routing, neither how the Smart IoT GW will interface with the M2M space. They will be described in the *D5.2 Baseline iNGENIOUS data management framework*.

At the time of writing this document, the Smart IoT GW is in a state in which only minor adjustments for the final integration and end-to-end tests are required before being ready for the first demonstration. These include establishing compatibility with the actual sensor equipment being used for the demonstration as well as integration into the network infrastructure of the M2M platform, which is currently being worked on. Once this work has been finalized, end-to-end tests will be performed with potential refinements and the final assembly of the physical device being implemented afterwards. The development activities can be briefly summarized as follows:

- Software component layout designed and implemented in lab-instance:
    - Implementation of core gateway components (Data Transformation, Routing Engine, Rule Engine).
    - Integration and interoperability of individual parts of the system.
- Implementation of simple HMI instances for monitoring and control:
    - Monitoring dashboards for gateway status and sensor data.
    - Control dashboard for core gateway components.
- Testing LoRa (WAN) integration, using Arduino-based sensor devices.
- Preparation of the interface to the M2M-platform.

Co-funded by the Horizon 2020 Framework Programme of the European Union

The following work is currently ongoing / to be finalized:

- Implementation of the network infrastructure for the M2M platform.
- Finalization of the gateway integration to the M2M platform.
- Compatibility with actual sensor devices for the demonstration.
- End-to-end testing of the complete system and final assembly of the physical device.

3.2.1 SMART IOT GW INTERFACES – SENSOR SPACE INTERFACES

This section describes how the Smart IoT Gateway will interface with the sensors. As mentioned earlier, the interfaces of the Smart IoT GW with the M2M space will be described in D5.2.

The smart IoT Gateway currently deployed in the SES Lab provides a single, unified **MQTT broker**, to which all types of sensors must connect in order to have their data processed. The way the individual sensors communicate with this MQTT broker depends on the specific sensor type and its physical communication interfaces. In the case of the lab instance, the **LoRa**-based sensor communication has been selected, using a special extension module to mount a *LoRa* antenna onto an *Arduino*. The gateway itself features a *LoRa transceiver module* mounted as a *shield* onto the *Raspberry Pi*. Due to regional specifications of the LoRa protocol, the module used only supports LoRa communication on the **EU863-870** frequency band (863-870 MHz).

Any LoRa signals received are decoded by a specific LoRa *concentrator* software, which in turn forwards the decoded messages for further processing to the gateway's **Chirpstack** software stack. This software stack consists of multiple components, implementing a full ready-to-use **LoRaWAN Network Server**, required to build a *LoRaWAN* network. While *LoRa* represents the *physical-layer* protocol, *LoRaWAN* adds multiple layers on top, managing critical aspects of a *LoRa* RF communication, like *duty cycles*, *frequency plans*, *adaptive data rate* and, additionally, it serves as a security component, by enabling **authenticated** and **encrypted** traffic between the gateway and the end-devices. When receiving a message from a registered sensor device, the Chirpstack application server publishes the payload extracted from the message to the MQTT broker mentioned above, ready to be picked up and processed by the gateway's main routing engine.

In addition to *LoRa(WAN)* based sensor devices, the Smart IoT Gateway also supports devices being connected via *Wi-Fi* (IEEE 802.11ac), *Bluetooth 5.0* (incl. BLE devices) as well as serial connections via the board's *GPIO* connectors. Any of these device types must publish their messages to the initially introduced MQTT broker to be picked up by the gateway's software. For exceptional cases, in which a translation to MQTT is impossible, the devices may as well communicate directly with the routing engine, without the intermediate step via the MQTT broker.

In the diagram below, the components highlighted in yellow feature custom implementation to serve the following purposes:

Co-funded by the Horizon 2020 Framework Programme of the European Union

Component	Description
Data Transformation	Collect incoming sensor messages from the MQTT broker and transform them to be further processed by the rule engine and stored in the local database
Rule Engine	Based on message priorities and route status parameters (will be defined in D5.2), select the most appropriate target route for a message.
Routing Engine	Prepare or schedule transmission of sensor messages via the respective route selected previously

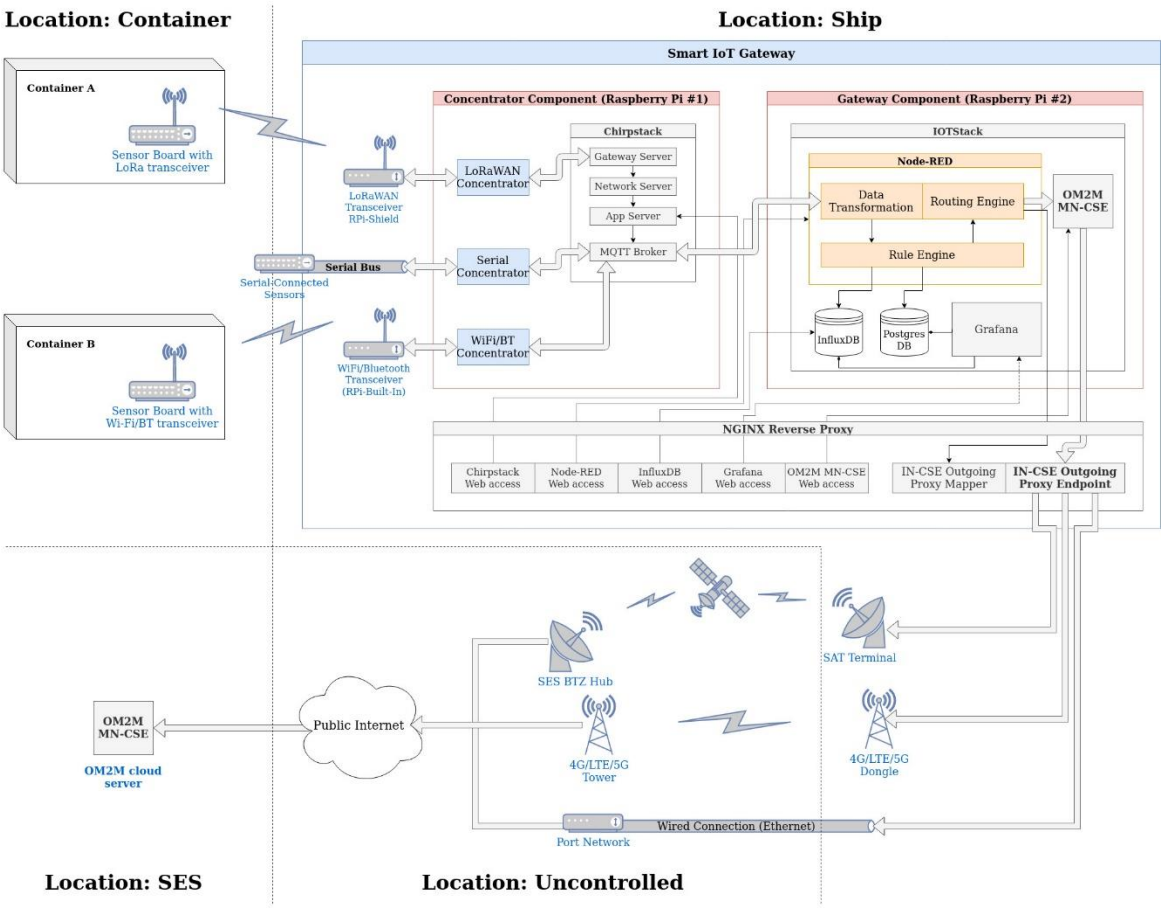


Figure 26: The overall system architecture. The description of the Sensor space interfaces primarily corresponds to the Concentrator Component seen in this diagram

3.2.2 SMART IOT GW DATA STORAGE

The Smart IoT Gateway features two core database components – besides software exclusive database instances – one being a **relational database** using **PostgreSQL 14** and the second one being a **timeseries database** based on **InfluxDB 1.8**. The PostgreSQL database stores (mostly) static information, including configuration data for the routing engine, as well as periodical snapshot values for route metrics and connection statistics for sensor devices. The InfluxDB on the other hand is the core storage instance for **sensor data**, stored in *timeseries* format. Additionally, the full history of statistics related to the Smart IoT Gateway is stored in this database. In the following pages, the individual tables and measurements/timeseries are specified.

Co-funded by the Horizon 2020 Framework Programme of the European Union

PostgreSQL: Relational database

The PostgreSQL database in Figure 27 contains three main groups of tables for different purposes:

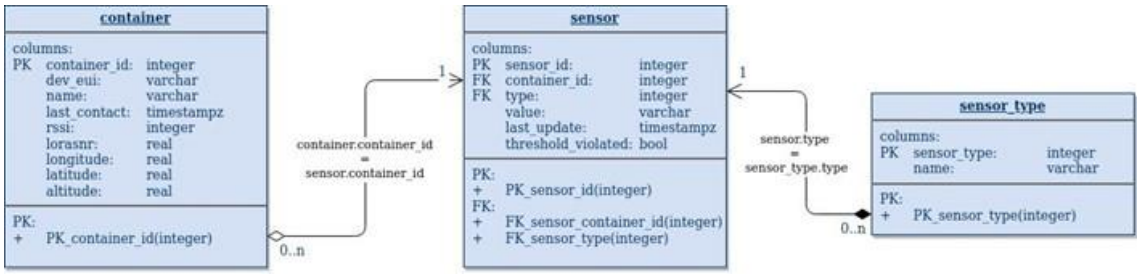


Figure 27: Table group container-sensor-sensor_type

The first group of tables consists of the tables: **container**, **sensor** and **sensor_type**. The latter holds pre-defined textual representation of the available sensor types ("GPS", "Temperature/Humidity", "Accelerometer", etc.). The *container* table holds information about each connected container. The LoRa specific fields *rssi* and *lorasnr* are only populated, when at least one LoRa-based sensor is connected to this container. The GPS fields *longitude*, *latitude* and *altitude* are only populated, if at least one GPS tracker is connected to this container and sending data to the Smart IoT Gateway. The field *last_contact* depicts the last time any of the related sensors sent a message to the Smart IoT Gateway. The field *dev_eui* is named in the scope of LoRaWAN sensors, however, it will be used as a secondary identifier for all types of sensor devices.

The sensor table holds information about each sensor sending data through the Smart IoT Gateway. Its field *container_id* references the *container_id* field in the *container* table as a foreign-key relation. Each container can have zero or more sensors and each sensor can only be related to zero or one containers. The *type* field references the *sensor_type* field in the *sensor_type* table as a foreign-key relation. Each sensor has exactly one sensor type and each sensor_type can be related to zero or more sensors. The *value* field contains a string representation of the last value received from the sensor with the timestamp being stored in the *last_update* field. Additionally, the *threshold_violated* field can be used to identify, whether a certain threshold has been violated by the last value.

Table	Column	Description
container	container_id	Database ID of the container object
	dev_eui	Secondary identifier (Part of LoRaWAN scope)
	name	Name/Label of the container object
	last_contact	Timestamp of the last message from a container
	rssi	Received Signal Strength Indicator (LoRaWAN)
	lorasnr	Signal-to-noise ratio (LoRaWAN)
	longitude	Last recorded longitude value of the container
	latitude	Last recorded latitude value of the container
	altitude	Last recorded altitude value of the container
sensor	sensor_id	Database ID of the sensor object
	container_id	Database ID of the container the sensor is part of
	type	Numeric representation of the sensor type

Co-funded by the Horizon 2020 Framework Programme of the European Union

	value	String representation of the last sensor value
	last_update	Timestamp of this sensor's last value update
	threshold_ violated	Boolean, true if a given sensor threshold is violated, false otherwise
sensor_type	sensor_type	Numeric representation of a sensor type
	name	Textual representation of a sensor type

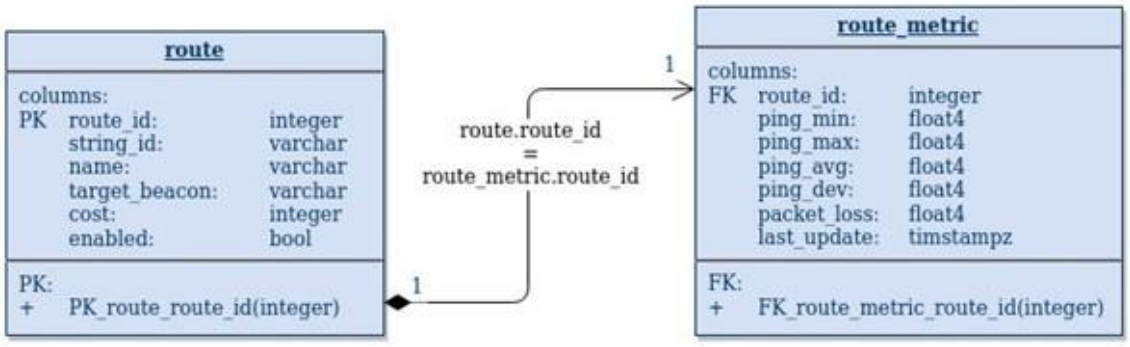


Figure 28: Table group route_metric-route

The second group consists of the tables: **route** and **route_metric**. The *route* table holds information about the network routes configured at the Smart IoT Gateway. The *target_beacon* is supposed to store an IP-address (or optionally a hostname) which is used to collect monitoring metrics to determine the quality of the route. *Cost* is an arbitrary number, defining the relative cost of a route compared to the other configured routes. The *enabled* boolean defines, whether a route is available for transmission or not (due to outages or forced). Each *route* object is related to exactly one *route_metric* object, holding the latest data required to determine the current rating of this route for a potential message to be forwarded, including a *last_update* timestamp to approximate the validity of this information.

Table	Column	Description
route	route_id	Database ID of the route object
	string_id	Textual identifier of the route object
	name	Name/Label of the route object
	target_beacon	Hostname/IP to target measurements at
	cost	Numeric value depicting the route-cost
	enabled	Boolean, depicting if a route is enabled
route_metric	route_id	Database ID of the related route object
	ping_min	Minimum value of the last ping measurement
	ping_max	Maximum value of the last ping measurement
	ping_avg	Average value of the last ping measurement
	ping_dev	Deviation of the last ping measurement
	packet_loss	Packet loss during the last measurement
	last_update	Timestamp of the last measurement

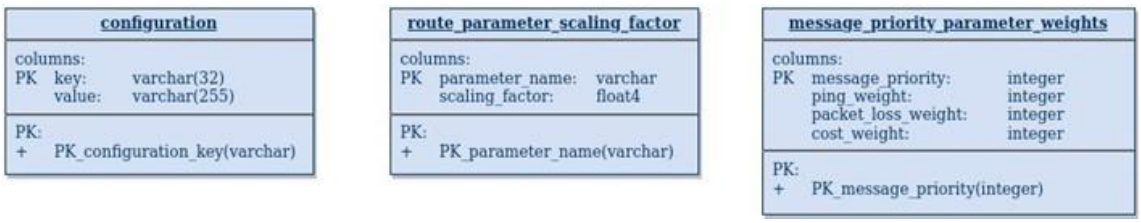


Figure 29: Configuration table group

Co-funded by the Horizon 2020 Framework Programme of the European Union

The remaining tables are non-related tables used to store configuration data. The **configuration** table allows to store free-form text as global configuration data, resembling a key-value mapping. The tables **route_parameter_scaling_factor** and **message_priority_parameter_weights** are used by the routing engine of the Smart IoT Gateway, which is described in the upcoming section of the document.

Table	Column	Description
configuration	key	Textual ID of the configuration entry
	value	Textual representation of the config. Value
route_parameter_scaling_factor	parameter_name	Textual identifier of the parameter
	scaling_factor	Decimal value of the scaling factor for the parameter
message_priority_parameter_weights	message_priority	Numeric representation of the priority value of a given message
	ping_weight	Weight of the ping-parameter
	packet_loss_weight	Weight of the packet-loss
	cost_weigth	Weight of the cost value

InfluxDB: Timeseries database

The local InfluxDB instance is used to store all historical sensor data received from all sensors attached to the Smart IoT Gateway.

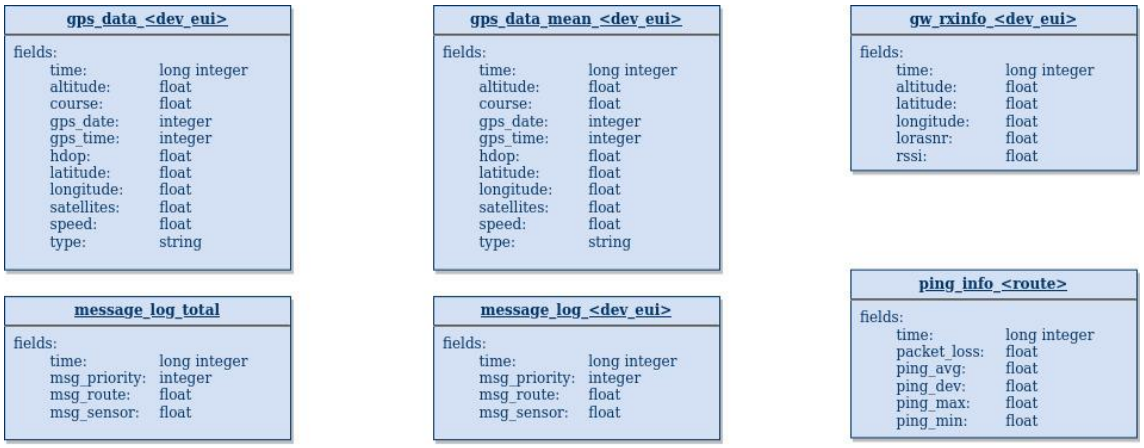


Figure 30: Overview of currently used InfluxDB measurements

Figure 30 displays all measurement tables used with the current test environment having only a GPS tracker attached as a sensor device. For each container (identified by the *dev_EUI* of the LoRaWAN configuration) the sensor data from individual sensor types are stored in their own measurements. These measurement tables vary in their set of *fields*, depending on the data fields provided by the sensor type. Each *<type>_data_<dev_eui>* measurement comes with a *<type>_data_mean_<dev_eui>* measurement table, storing accumulated mean values for that particular sensor device.

Additionally, for each container device (per dev_EUI) a measurement *gw_rxinfo_<dev_eui>* stores LoRa signal and GPS information received from the LoRaWAN gateway during transmissions. The *message_log_<dev_eui>*

Co-funded by the Horizon 2020 Framework Programme of the European Union

measurement keeps message routing statistics for a specific container – *message_log_total for the whole Smart IoT Gateway, respectively.*

Finally, ping_info_<route> stores ping and packet loss data for each route of the Smart IoT Gateway.

As at the current stage, the expected amount of retrieved sensor data is fairly low, the current strategy is considered sufficiently resilient. If in the future, the requirement for scalability to a larger amount of sensor data arise, the system can be easily adapted, using load balancing across multiple instances of individual Docker containers, or even the full IOTstack.

3.2.3 SMART IOT GW MANAGEMENT SERVICE

The core software components of the Smart IoT Gateway are running in individual **Docker** environments. The use of **Portainer**[1] allows straight forward control over the individual *Docker Containers* via a feature-rich and intuitive graphical user interface.

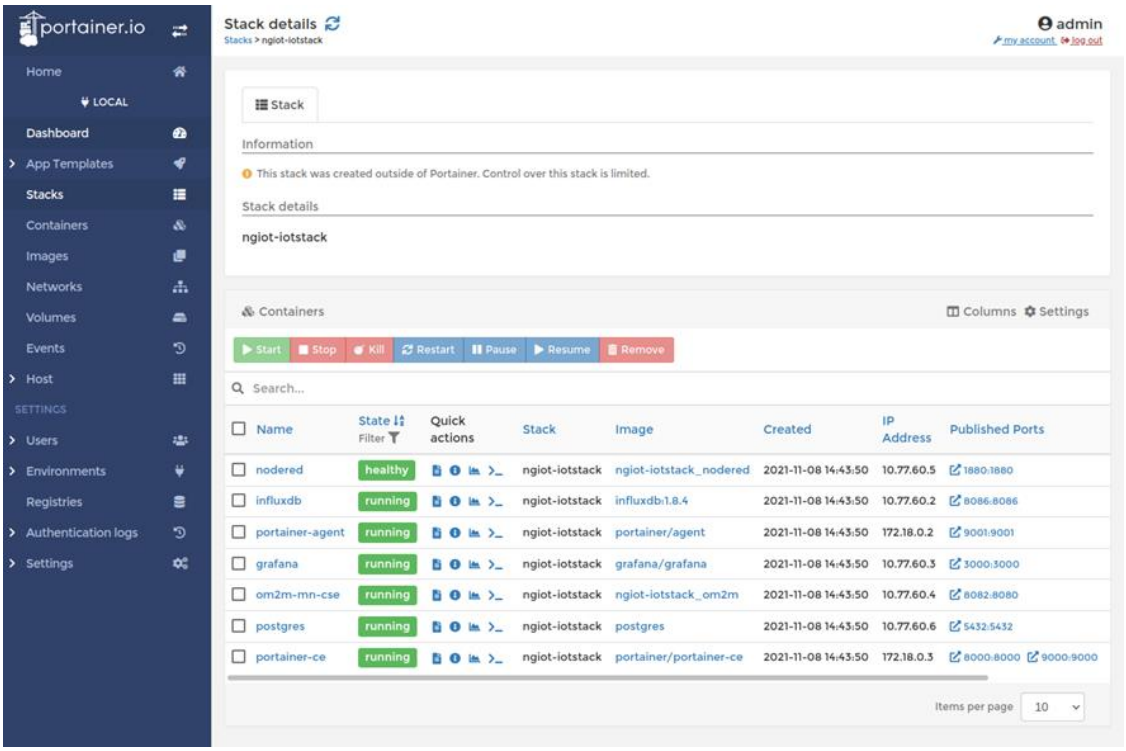


Figure 31: Portainer – IoT stack overview

Figure 31 and Figure 32 display the Docker Container stack overview for the IoT stack and Chirpstack of the Smart IoT Gateway. Via the side menu all Portainer features can be accessed – a detailed user guide can be found on the official documentation webpage[2].

[1] Portainer: Reference at https://www.portainer.io/
[2] Portainer: Documentation available at https://docs.portainer.io/

Co-funded by the Horizon 2020 Framework Programme of the European Union

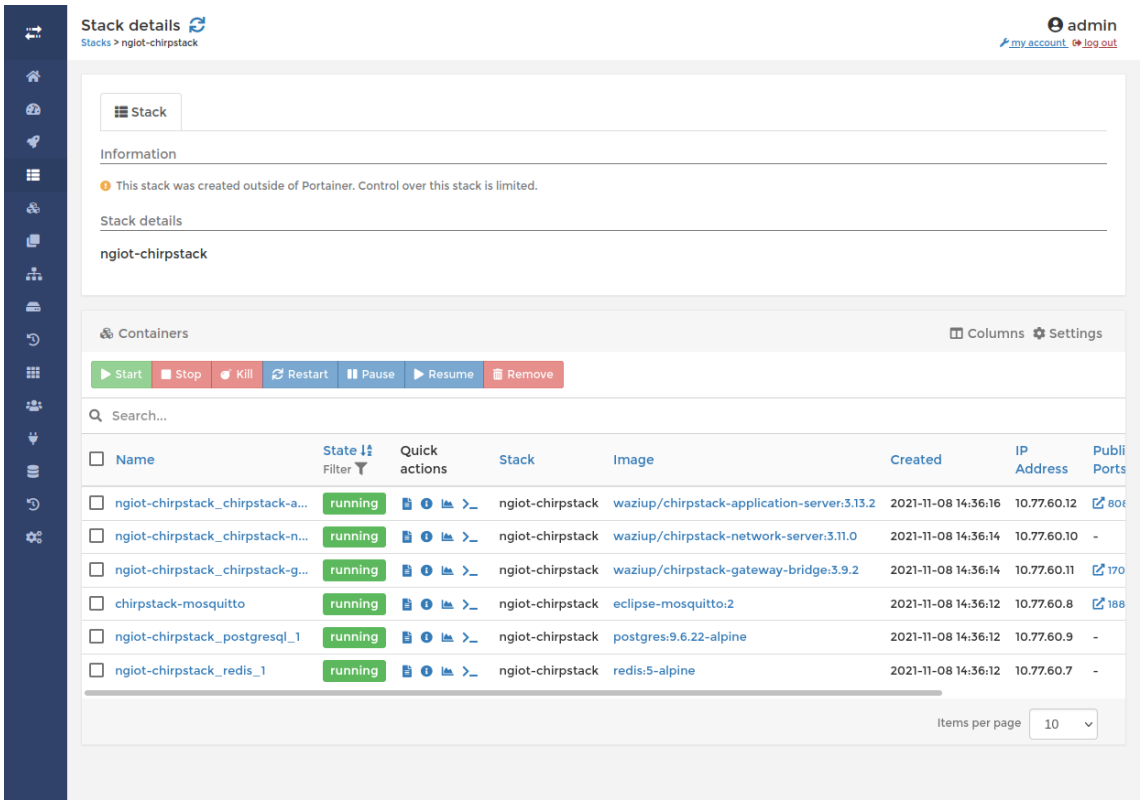


Figure 32: Portainer - Chirpstack overview

3.2.4 SMART IOT GW HMI & API

To observe the status of the Smart IoT Gateway including its connected sensor devices, several dashboards are set up using the **Grafana** dashboard engine interfacing with the previously mentioned databases. Additionally, **Node-RED** dashboards will be used to change core settings related to the routing and rule engine of the Smart IoT Gateway. Any screenshots shown within this section only show the current dashboard layouts used in the testbed environment – Future iterations will most probably vary.

The *Gateway Overview* dashboard in Figure 33 displays a histogram and summary of the target routes selected for the messages sent by all related containers. Furthermore, the last ping values for the configured routes and general information about the gateway hardware will be available here.

Co-funded by the Horizon 2020 Framework Programme of the European Union

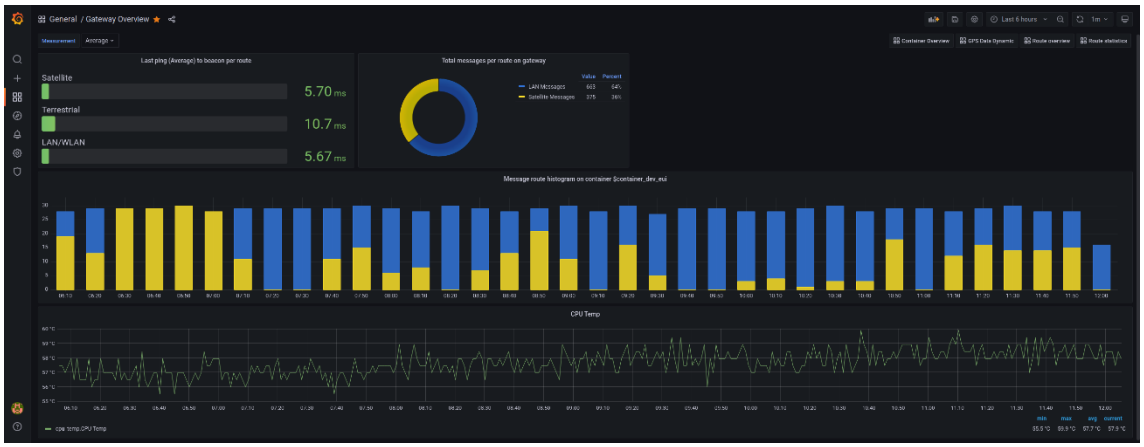


Figure 33: Gateway overview dashboard

The *Container Overview* dashboard in Figure 34 displays similar information regarding the message histogram as the previous *Gateway Overview*, however, only for the messages emitted by the specified container/sensor-device, which can be selected using the dropdown list in the top-left corner of the dashboard. Additional information includes a visualization of the LoRa signal quality as well as a summary table for the latest sensor values.

Figure 34: Container overview dashboard

The *GPS data* dashboard shown in Figure 35 displays current and historical information about the GPS data received from the selected container's GPS trackers. A similar dashboard may as well be introduced for the Smart IoT Gateway itself, should it be equipped with its own GPS tracker. Both the *Horizontal Delusion of Precision (HDOP)* and *Number of Satellites* are shown as timeseries, histogram and current value displays. Additionally, a **map view** uses the GPS coordinates to display the transmitted locations for the selected timeframe, using color coded HDOP to visualize the accuracy of the individual GPS data.

Co-funded by the Horizon 2020 Framework Programme of the European Union

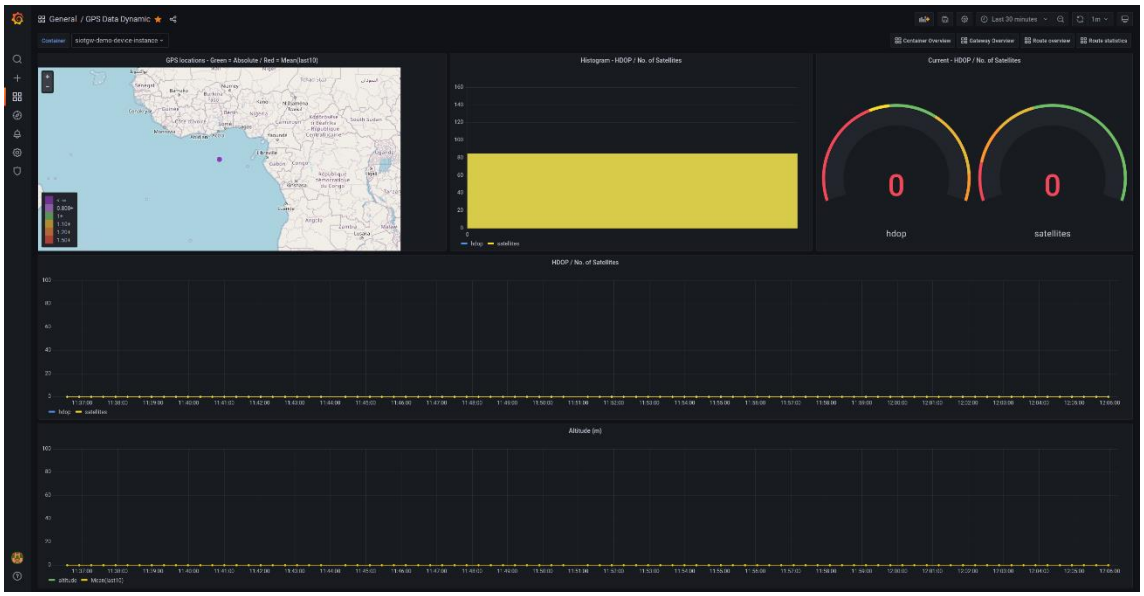


Figure 35: GPS data

The *Route overview* dashboard in Figure 36 displays the configuration of the available network routes as well as the historical trend of the routes' *ping* and *packet loss* measurements. Using the top-left dropdown selector, it is possible to switch the *ping* measurement between *average*, *maximum*, *minimum* and *mean-deviation*.

Figure 36: Route overview dashboard

The *Rule configuration* dashboard in Figure 37 allows to configure the most important settings for the rule- and routing engine, which will as well be specified in D5.2. These configurations are forwarded to the PostgreSQL database, from which they are retrieved for display by the respective Grafana dashboards. This can cause slight delays between the update of the configuration and the update of the displayed configurations in Grafana.

Co-funded by the Horizon 2020 Framework Programme of the European Union

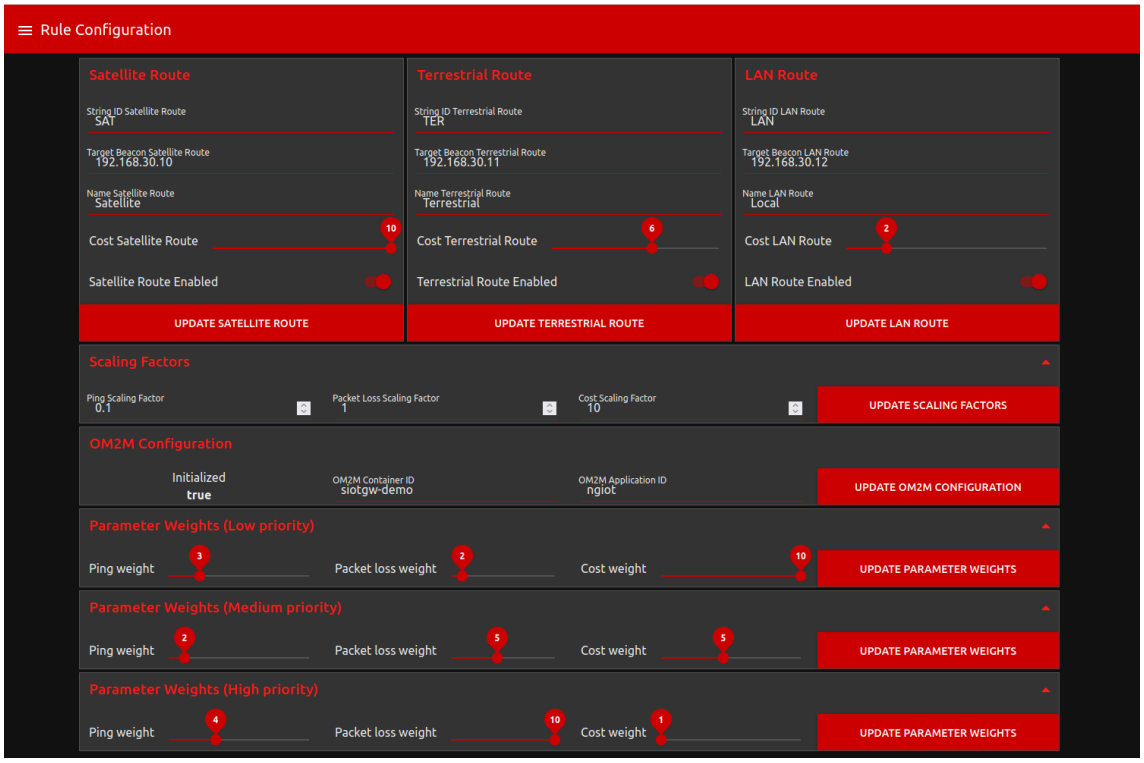


Figure 37: Rule configuration dashboard (Node-RED)

3.3 Satellite

The role of satellite in NG-RAN IoT can be broadly considered in two areas: satellite backhaul and direct access to devices over satellite.

Satellite backhaul use cases are where the satellite network is used to backhaul the IoT traffic from IoT devices connected to a satellite edge node. Satellite direct access is where IoT devices are connected directly to the satellite network. iNGENIOUS considers both cases to various degrees and both are described here in more detail and will be researched further during the project.

From a demonstration perspective, this project focuses on the GEO satellite constellations which is primarily used for satellite backhaul of IoT traffic and therefore the use cases requirements, research and solutions are focused therein. The project is also undertaking research in satellite direct access but as the use cases are the primary focus at the moment, the research in this area is not discussed in detail for this deliverable.

3.3.1 SATELLITE BACKHAUL

A satellite backhaul connectivity deployment includes UEs (IoT devices) connected to the edge node which is connected to, or integrated with, a satellite terminal. The satellite terminal communicates with the central node over a satellite link. The satellite backhaul is seen as a transport layer for the messages between the edge and the central node. Because of this, the backhaul should be as transparent as possible, while at the same time being

Co-funded by the Horizon 2020 Framework Programme of the European Union

able to assure a guaranteed communication quality depending on the requirements of the use case.

IoT devices can then use the satellite backhaul to send regular status updates via an IoT GW which processes and optimises the data before forwarding to the IoT cloud/data centre as shown in Figure 38.

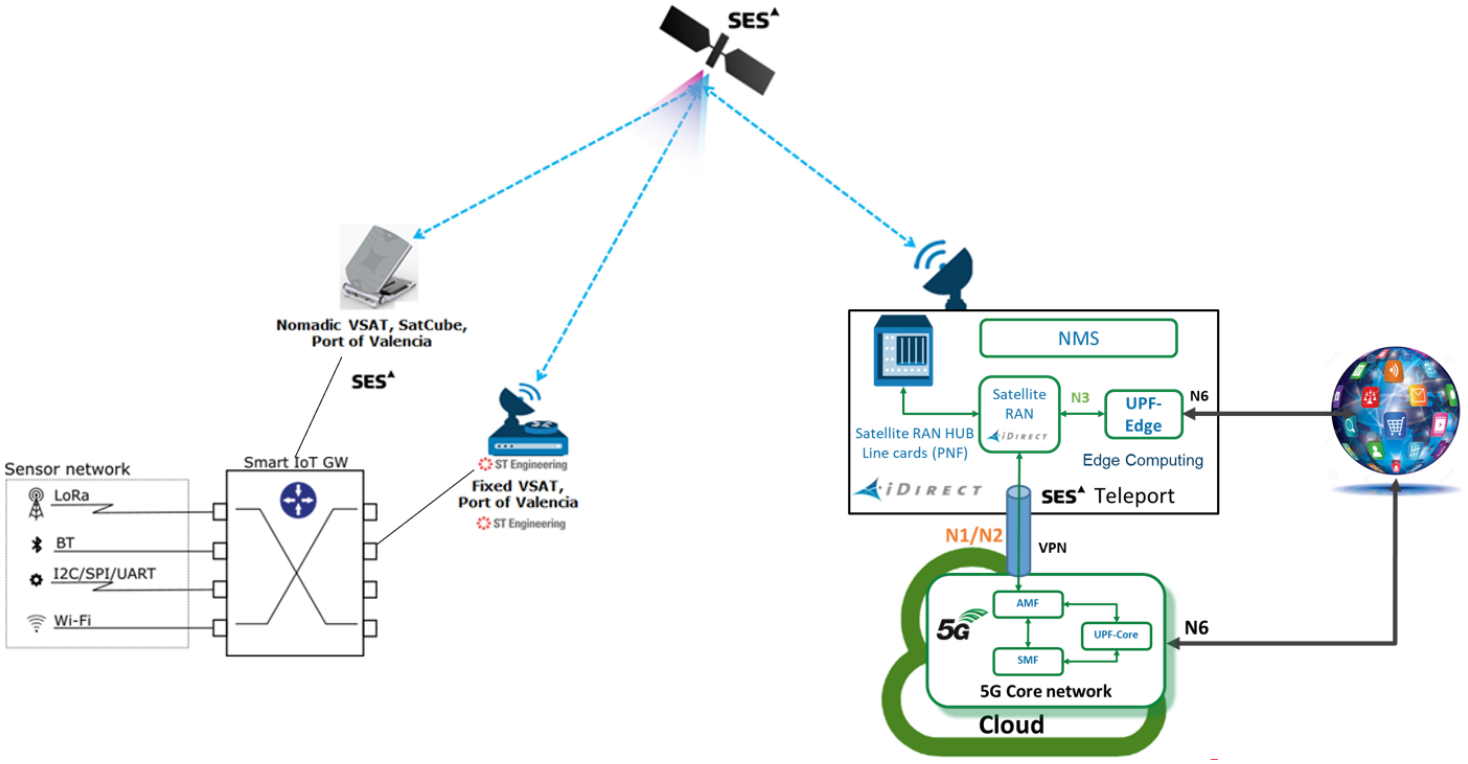


Figure 38: Satellite backhaul connectivity architecture

As illustrated in Figure 38 satellite network architecture used in iNGENIOUS also incorporates major concepts and components of the 5G architecture to provide the end-to-end satellite connectivity. Firstly, the space segment is provided by SES' multi-orbit and multi-band transparent (bent-pipe) satellite fleet which provides connectivity between the satellite remotes and the hub platform located at the SES' teleport in Betzdorf, Luxembourg. The satellite network deployed at the SES Teleport is built using IDR's 5G-enabled Velocity™ Intelligent Gateway (IGW) system and uses satellite capacity provided by SES's Ku Band satellite. This architecture is captured by ETSI SES in ETSI SES - *DTR/SES-00405 - TR 103 611: Satellite Earth Stations and Systems (SES); Seamless integration of satellite and/or HAPS (High Altitude Platform Station) systems into 5G systems* specifically "Scenario A3 - Indirect mixed 3GPP NTN access with bent-pipe payload".

Co-funded by the Horizon 2020
Framework Programme of the European Union

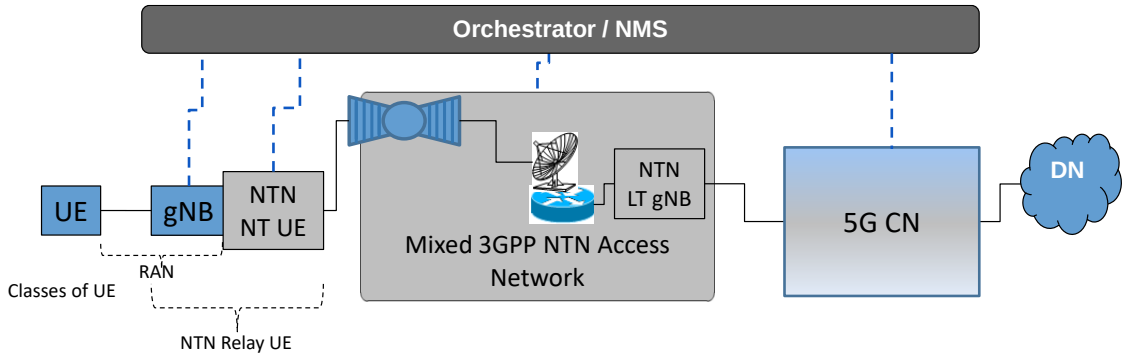


Figure 39: Scenario A3 - Indirect mixed 3GPP NTN access with bent-pipe payload

In this case the Non-terrestrial Network (NTN) terminal and modified gNB are using standard interfaces to connect to the 5G core network while continuing to use the satellite radio network interface over the air.

Within the satellite domain, a standard 3GPP 5G next generation core network (5GCN) is integrated to operate the control and user plane functions of the satellite network. Once this is in place, the hub-side of the existing satellite network is modified to comply with standard 3GPP interfaces, using N2 and N3 interfaces to communicate with the 5GCN. The satellite terminal is also modified to communicate with the 5GCN via the 5G standard N1 interface. This results in a 5G-enabled satellite/NTN terminal presenting itself as a 5G UE to the NGCN and the satellite hub-side network presenting itself as a standard gNB. This new satellite RAN (SatRAN) connects with the NTN 5GCN on the network side, presenting as a gNB, while continuing to use the existing satellite radio over the satellite. It is important to highlight the standard 3GPP 5G core network is unmodified and all modifications are incorporated in the 5G-enabled satellite network. Together, the modified satellite/NTN terminal, SatRAN, and 5GCN provide the NTN satellite network connectivity.

The 5G-enabled satellite system incorporates a cloud hosted 5GCN which is connected over VPN with the SatRAN installed at the SES Teleport in Betzdorf. The Betzdorf SatRAN control plane is integrated with the cloud hosted 5GCN. A 5G UPF instance is deployed at the SES Betzdorf teleport site and integrated with the 5GCN to support the local breakout of user plane traffic at the SES Betzdorf teleport, to facilitate edge computing. At the remote site, the edge node is connected to the satellite modem on the network side and the IoT devices on the end user side.

The optimization of the IoT traffic over the satellite link is a critical part of deploying satellite edge IoT devices. Further information on the optimization of edge IoT traffic is described earlier in Section 3.2 Smart IoT Gateway.

The Transport and Ship use cases both use satellite backhaul to connect the IoT gateway to the IoT cloud/data center. The role of satellite in both use cases is described further in Section 4.2.5.

Co-funded by the Horizon 2020
Framework Programme of the European Union

3.3.2 DIRECT ACCESS

Satellite direct access for IoT devices is where IoT devices are connected directly to the satellite network which in turn connects to the IoT cloud/data centre. This allows delivery of the IoT content in a more efficient and cost-effective manner by utilising a Direct-to-Satellite approach rather than simply using the satellite to backhaul IoT traffic.

Currently, pre-existing narrow-band and some broadband satellite services are being marketed as satellite direct access IoT services. Relatively new entrants to this market have identified the gaps in the Direct-To-Satellite IoT market (in terms of message size, connectivity, service cost, etc.) and are providing alternative products that utilise existing and new satellite constellations. These alternative products use a variety of communication media between the IoT device and the satellite, including both a 3GPP and a non-3GPP based physical layer.

Direct access of IoT devices over satellite can be categorized in the follow three areas.

Non-3GPP IoT access - Direct access of IoT devices over satellite using proprietary non-3GPP access technologies is already supported by many industry partners today. Within iNGENIOUS, IDR are researching the use of their own proprietary access technologies to determine if they are suitable for connecting IoT devices over satellite.

3GPP 5G NR-NTN - 5G NR NTN support is a new feature being added in 3GPP Rel 17. This offers the capability to use a standard 5G NR waveform over satellite links. This could offer new opportunities for both the satellite backhaul architecture and the direct access use cases and will also be researched further during this project as the 3GPP standards are released and products developed to support them.

3GPP NB-IoT NTN - 3GPP are also considering making changes to NB-IoT to support NTN. These changes are being studied in Release 16 and Release 17. In general, the changes are the same or similar to the changes outlined for 5G-NT NTN but tailored for NB-IoT.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4 Port deployment and Relation to the iNGENIOUS Use Cases

iNGENIOUS will implement and deploy smart NR and NG-RAN IoT designs in real environments for demonstrating specific project use cases in different supply chain scenarios. Among the variety of scenarios, maritime ports will be one of the environments where several use cases will be demonstrated. This section is devoted to the description of the networking deployment performed in the port as well as the technical work realized in the different use cases involving it. This allows to integrate and test the work done in the different use cases in a relevant real environment as it is the port of Valencia.

4.1 Port deployment study

The port of Valencia will host the demonstration of the use case on Improved Driver's Safety with MR and Haptic Solutions. For meeting the connectivity, bandwidth, and latency requirements of this use case, 5G technology was selected as the best candidate option for enabling wireless communications. Considering the absence of a 5G private network at the port of Valencia, a first study was performed to identify the potential coverage of commercial 5G networks inside the port.

Phase 0 – Port Area and 5G Connectivity: During the initial stage, FV discussed internally with the Port Authority of Valencia about the areas available for performing the demonstration. As a result of this query, the area shown in blue in Figure 40 was selected as the area of demonstration, since it is a wide-open area located next to cruise berths where AGVs could be driven without interfering the port operative. Considering this area as the demonstration zone, initial field tests were performed for measuring the signal received by commercial 5G nodes. These tests concluded that no 5G signal is received by any of the existing commercial networks at that area, and, as a consequence, a 5G mmW NSA node will be deployed and installed at the Valencia Port facilities. To carry out this deployment, FV and NOK performed a preliminary planning and study to define the location, the available infrastructure and the equipment needed for installing the new 5G node.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Figure 40: Commercial 5G Coverage at the Port of Valencia

Phase 1 – Antenna Location: After deciding the port area where the demonstration will be performed, the specific location and infrastructure required for installing the 5G mmW node was selected. In particular, the area of demonstration will be the one shown in Figure 41.

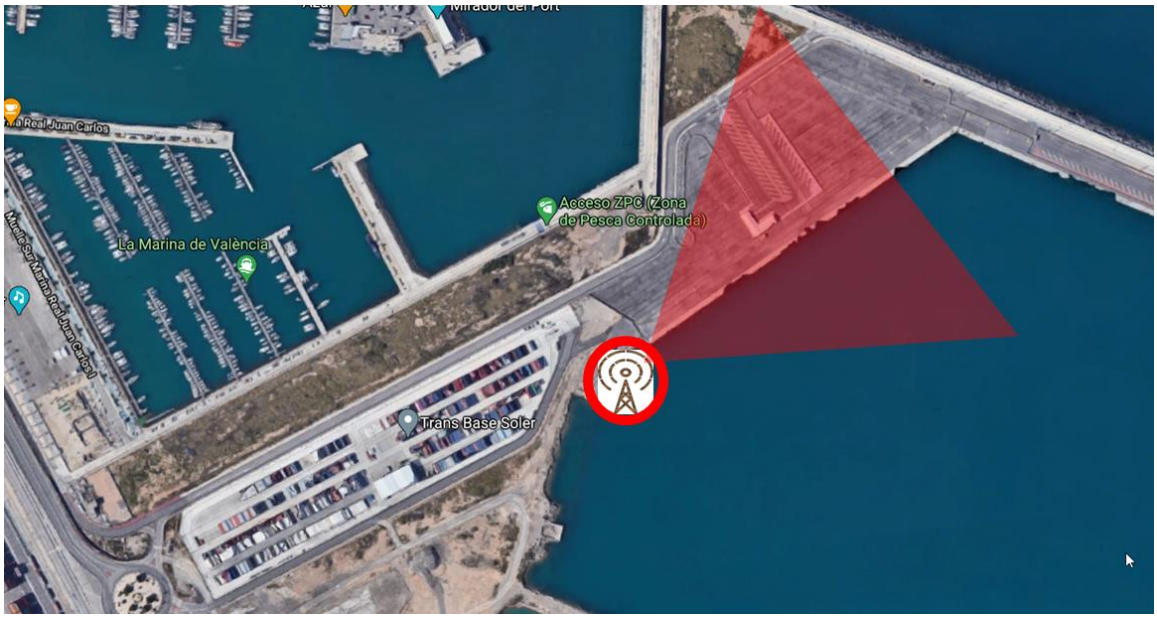


Figure 41: Area of demonstration and deployment of 5G mmW node

In that area, an outside cabin was identified as the location where the baseband, core and MEC equipment will be placed in a rack. The outside cabin offers power supply and fiber optic connectivity to Internet. To place the 4G and 5G antennas, a 20 m high pole located next to the outside cabin was selected. On this pole, both antennas will be installed at 17-18 m high approximately to cover the area shown below.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Figure 42: Area of demonstration

Figure 43: Outside cabin

Figure 44: Inside cabin

Figure 45: Pole

Phase 2 - Equipment and configuration: The configuration used in this deployment is Non-Stand Alone: NSA 3X. It is based on E-UTRAN New Radio - Dual Connectivity (EN-DC), a technology that enables introduction of 5G services and data rates in a predominantly 4G network. UEs supporting EN-DC can connect simultaneously to LTE Master Node eNB and 5G NR Secondary Node gNB. This approach makes it possible to deploy 5G services without the expense of a full scale 5G Core Network.

An EN-DC enabled UE first registers for service with the 4G EPC. The UE also starts reporting measurements on 5G frequencies. If the signal quality for the UE supports a 5G service, the LTE eNB communicates with the gNB to assign resources for a 5G bearer. The 5G-NR resource assignment is then signalled to the UE and once the RRC Connection Reconfiguration procedure is completed, the UE simultaneously connects to the 4G and 5G networks. In this configuration, user data traffic directly flows to the 5G gNB part of the base station.

Co-funded by the Horizon 2020 Framework Programme of the European Union

This deployment will be done using two radio units: AHHA and AWEUD.

- 4G antenna: AHHA to radiate 5 MHz in the 2600 MHz FDD band (B7) for the anchor.
- 5G antenna: AWEUD to radiate 8 carriers of 100 MHz each in the 24Ghz TDD for the millimetre wave band (n258).

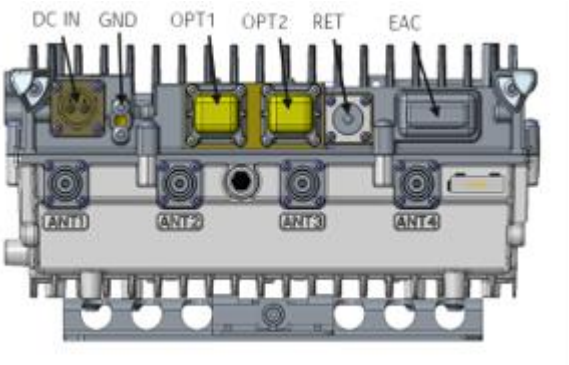


Figure 46: AHHA

Figure 47: AWEUD

Both antennas are outdoor antennas and can therefore be placed outside, on the pole.

As we are using AWEUD antenna, this deployment uses the 5G FR2 technology frequency range which goes from 24.25 GHz to 52.6 GHz. The 4G anchor band has been provided by Telefonica. And we have decided which is the millimetre band in which we are going to radiate using 5G in the 2600 MHz frequency band.

The following scheme represents a summary of the equipment installation.

Co-funded by the Horizon 2020 Framework Programme of the European Union

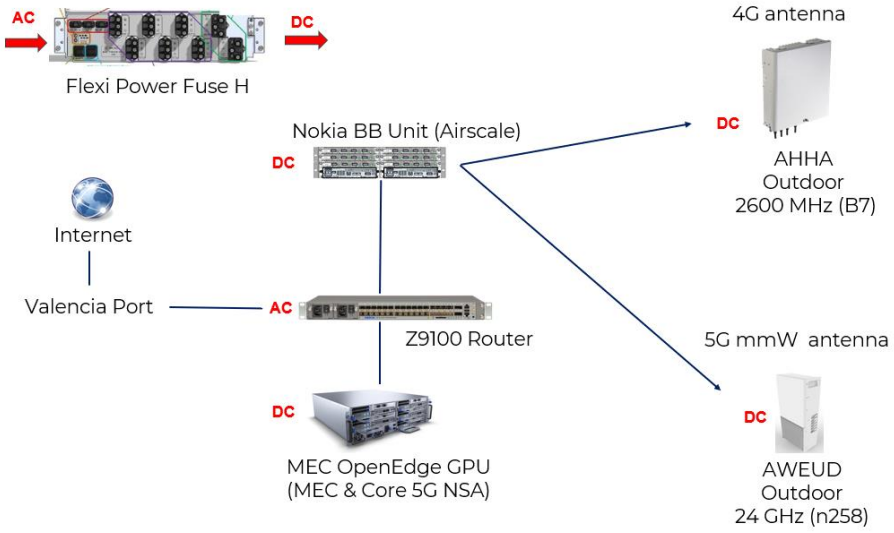


Figure 48: Antenna installation scheme

As explained, two antennas will be used, one for LTE and another one for 5G.

These antennas will be connected to Nokia Base Band Unit, known as Airscale. This base band is composed of the following modules that allow the control of the antennas:

- 4G antenna control modules: 1 ABIA and 1 ASIA for 4G base band
- 5G antenna control modules: 1 ABIL and 1 ASIK for 5G base band

The 4G antenna will be connected to ABIA module and the 5G antenna will be connected to ABIL module. Modules ASIA and ASIK are interconnected, it means that the 4G radio control is connected to the 5G radio control.

This baseband is connected to Z9100 switch which is in charge of connecting the baseband to the MEC. This Multi access edge computing is the Open Edge server, and it is in charge of the processing tasks near the radio and performs the CORE function.

A virtual machine is installed in the MEC to have here the installation of the CORE emulator (EPC). In addition, the switch will be connected to the Port of Valencia network which has internet access.

IP addresses will be fixed, and the remote access will be via VPN. Also, NAT will be used between Nokia outgoing traffic and Fundación Valencia IP address. Therefore, it is possible to have remote access to the radio to control it. As well, a 5G mmW modem is installed and remotely monitored in order to take measurements without the necessity of being physically at the port of Valencia.

In terms of power, AC is available inside the cabin. The router and the rectifier (Flexi Power Fuse) will be connected to the AC. The rectifier transforms AC power to DC power in order to feed all remaining equipment.

4.2 Relation to UCs

Co-funded by the Horizon 2020
Framework Programme of the European Union

This section refers to the relationship of the innovations and state-of-the-art Radio Access Network technologies mentioned above, with the corresponding use cases in iNGENIOUS. Table 11 was already introduced in *D4.1 Multi-technologies network for IoT* [2], even though a slight change is being introduced in this deliverable. The Flexible PHY/MAC changes from being an innovation concept to an innovation demo which will happen on a Factory UC trial.

Table 11: NG-IoT RAN functionalities mapped to UCs

Functionality	Factory UC	Transport UC	AGVs UC	Ship UC
5G Modem	Innovation demo		Innovation demo	
Flexible PHY/MAC	Innovation demo			
AI/ML for RAN	Innovation concept			
mmW deployment			Innovation demo	
Satellite Backhaul		SoA		SoA
Smart IoT Gateway				Innovation demo

4.2.1 FLEXIBLE PHY/MAC

The flexible PHY/MAC innovation presented in Section 2.1 will be applied in the Factory UC, which will be demonstrated in the TUD premises. The Factory UC has been designed to allow diverse applications in an industrial environment such as automated robot control, different types of sensors, actuators, and parallel control loops for connecting machines and humans. The flexible PHY/MAC innovation will be employed in the Factory UC to enable the different applications in the same network with its necessary PHY resources, that will be allocated dynamically. In particular, the TUD testbed is equipped with several SDR devices that will be used by both UEs and BSs. As an exemplary scenario, if there are two applications with different requirements such as video stream and robot control, the BS will allocate more time transmission to the video stream application since it requires more data rate. Additionally, we will have more UEs connected to the network by emulating the applications, such that the network can be a testbed in a dynamic environment. Moreover, the 5G Core and MANO layers will manage the UE connection to the 5G network and will orchestrate the resources.

4.2.2 AI/ML FOR RAN

The introduction of AI/ML for RAN presented in Section 2.2 will serve as an innovation concept for the Factory UC. ML models will be used to improve the decision making of O-RAN entities, increasing the performance of the network, and guaranteeing better service through the introduction of slices and Anomaly Detection and Traffic Steering loops. When fully developed, these models could be used at ASTI's factory, guaranteeing the needed QoS of different AGVs, cameras or robot arms.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4.2.3 MMW DEPLOYMENT

The AGV UC at iNGENIOUS aims to improve driver's safety at Valencia Port to avoid a potentially hazardous environment given by the presence of AGVs and machinery involved in the transportation of goods at the port.

In order to remove the AGV operator from the scene, an indoor immersive cockpit is designed to remotely control the vehicle. To accomplish the network requirements for the Tele-operation Driving (ToD) implementation, in terms of latency, bandwidth and throughput, a 5G coverage is needed.

The innovation of the AGV UC comes from the recreation of the AGV's point of view into the immersive cockpit used by the operator. The implementation of the architecture of this ToD is based on the network capabilities. Low latency at the maximum bandwidth is required, not only in DL, but also in UL, to have a proper Tele-operation that fulfils the given KPIs.

As defined in Section 4.1, the mmW antenna deployed at Valencia Port gives 5G connectivity over the AGV operating area. The AGV is equipped with an Askey 5G mmW modem that enables vehicle connectivity to the MEC. Once this connectivity is stable and robust, the information flux required for the ToD is settled, allowing an innovative, remote, and safe driving through 5G network, working with IoT, mixed reality and haptic devices.

4.2.4 SMART IOT GATEWAY

The Ship UC aims at providing E2E asset tracking enabling real-time/periodic monitoring of predetermined parameters (temperature, humidity, accelerometer, etc.) of shipping containers when they are sailing on the sea, or when the ship approaches at the port. IoT tracking devices will be installed on the shipping containers transported by ships and trucks on both segments. During the trip, depending on the service level required by the owner of the container and the supply chain associated, the heterogeneous IoT devices send regular status updates and a Smart IoT GW, defined in Section 3.2, on the ship/truck gathers and processes the data and the connectivity with the IoT cloud/data centre is obtained through satellite backhaul or terrestrial access network.

4.2.5 SATELLITE BACKHAUL

For both the Transport UC and the Ship UC, the satellite network provides the backhaul connection, defined in Section 3.3, between the edge node and cloud to ensure the edge node remains connected to the network even when there is no terrestrial or other means of connecting the edge node to the cloud.

A good example of this is the real time monitor of the shipping container onboard ships which are out at sea with no access to a terrestrial network. The satellite connectivity allows the operator to continue to monitor the edge node and connected IoT devices while at sea. This allows for active intervention once the container reaches the shore.

Co-funded by the Horizon 2020 Framework Programme of the European Union

5 Conclusions

This deliverable has detailed all the technical innovations and state-of-the-art of next generation RAN and smart NR in iNGENIOUS. More so, a relationship with use cases has been made when applicable, showing the importance of the innovations in real-world scenarios. The two focus areas of this deliverable have been addressed, being Smart NR and NG-RAN IoT, and an analysis of a real environment implementation and deployment has been included.

Regarding Smart NR, a Flexible PHY/MAC implementation has been introduced, with emphasis on the multiple access protocol and integration with 5G higher layers. Furthermore, use of AI/ML aims to improve the networks of the future by making them smarter, and an anomaly detection use case has been presented showing the great potential of these type of implementations, opening a wide range of possibilities to improve current networks algorithms.

The NG-RAN IoT will be used in a variety of verticals, having to focus on the interoperability between technologies to interconnect different devices, but also on the performance in high-demanding scenarios. An NR modem has been introduced, focused on connecting IoT devices to the 5G network. We obtained very promising results on the first validation scenarios. Furthermore, a Smart IoT Gateway was presented, which supports multiple Radio Access Technologies, opening the possibility of managing devices with different demands on different networks from the same gateway.

The port deployment has been successfully done, were a mmW private network will enable tele-operation driving thanks to cutting-edge throughput and latency. Finally, the relation between each innovation and state-of-the-art technology has been presented, showcasing the importance of the results achieved towards the project's use cases.

Co-funded by the Horizon 2020 Framework Programme of the European Union

References

[1] iNGENIOUS, “Deliverable 2.1 Use Cases, KPIs and Requirements,” [Online]. Available: https://zenodo.org/record/4683717/#.YHmlh-gzZaR.

[2] iNGENIOUS , “Deliverable 4.1 Multi-technologies network for IoT,” [Online]. Available: https://zenodo.org/record/4836191/#.YeFMwP7MIQ9.

[3] UERANSIM , [Online]. Available: https://github.com/aligungr/UERANSIM. [Accessed 14 01 2022].

[4] O-RAN, “Open Source Community,” [Online]. Available: https://wiki.o-ran-sc.org/.

[5] 3GPP, “Management and orchestration; Architecture framework,” [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128500_128599/128533/15.00.00_60/ts_128533v150000p.pdf.

[6] iNGENIOUS, “Deliverable 3.2 Proposals for Next generation of connected IoT modules,” [Online].

Co-funded by the Horizon 2020 Framework Programme of the European Union