5TH GENERATION END-TO-END NETWORK, EXPERIMENTATION, SYSTEM INTEGRATION, AND SHOWCASING

[H2020 - Grant Agreement No. 815178]

Deliverable D4.12

The Surrey Platform (Release C)

Editor D. Triantafyllopoulou (UNIS)

Contributors D. Triantafyllopoulou, S. Vahid, K. Moessner, R. Yogaratnam, F. Carrez (UNIS), C. Sakkas, G. Theodoropoulos, A. Papaioannou, E. Vergis, V. Koumaras (INF), I. Etxebarria, I. Pretel, E. Saiz (FON), M. Rajiullah (KAU), J. McNamara (LMI)

Version 1.0

Date 31/08/2021

Distribution PUBLIC (PU)

List of Authors

UNIS	UNIVERSITY OF SURREY
D. Triantafyllopoulou, S. Vahid, K. Moessner, R. Yogaratnam, F. Carrez	
INF	INFOLYSiS
C. Sakkas, G. Theodoropoulos, A. Papaioannou, E. Vergi, V. Koumaras	
KAU	KARLSTAD UNIVERSITY
M. Rajiullah	
FON	FON TECHNOLOGY SL
I. Etxebarria, I. Pretel, E. Saiz	
LMI	L.M. ERICSSON LIMITED
J. McNamara	

Disclaimer

Copyright

The 5GENESIS Consortium consists of:

NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"	Greece
AIRBUS DS SLC	France
ATHONET SRL	Italy
ATOS SPAIN SA	Spain
AVANTI HYLAS 2 CYPRUS LIMITED	Cyprus
AYUNTAMIENTO DE MALAGA	Spain
COSMOTE KINITES TILEPIKOINONIES AE	Greece
EURECOM	France
FOGUS INNOVATIONS & SERVICES P.C.	Greece
FON TECHNOLOGY SL	Spain
FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.	Germany
IHP GMBH – INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ-INSTITUT FUER INNOVATIVE MIKROELEKTRONIK	Germany
INFOLYSIS P.C.	Greece
INSTITUTO DE TELECOMUNICACOES	Portugal
INTEL DEUTSCHLAND GMBH	Germany
KARLSTADS UNIVERSITET	Sweden
L.M. ERICSSON LIMITED	Ireland
MARAN (UK) LIMITED	UK
MUNICIPALITY OF EGALEO	Greece
NEMERGENT SOLUTIONS S.L.	Spain
ONEACCESS	France
PRIMETEL PLC	Cyprus
RUNEL NGMT LTD	Israel
SIMULA RESEARCH LABORATORY AS	Norway
SPACE HELLAS (CYPRUS) LTD	Cyprus
TELEFONICA INVESTIGACION Y DESARROLLO SA	Spain
UNIVERSIDAD DE MALAGA	Spain
UNIVERSITAT POLITECNICA DE VALENCIA	Spain
UNIVERSITY OF SURREY	UK

Version History

Rev. N	Description	Author	Date
1.0	Release of D4.12	D. Triantafyllopoulou (UNIS)	31/08/2021

List of Acronyms

Acronym	Meaning
3GPP	3rd Generation Partnership Project
5G & 6GIC	5G & 6G Innovation Centre
5GC	5G Core
AMF	Access and Mobility Management Function
AP	Access Point
APEX	Adaptive Policy EXecution
API	Application Programming Interface
CoAP	Constrained Application Protocol
COTS	Commercial Off The Shelf
CPE	Customer Premises Equipment
DWDM	Dense Wavelength Division Multiplexing
ELCM	Experiment LifeCycle Manager
eMBB	enhanced Mobile Broadband
EMS	Element Management System
EPC	Evolved Packet Core
EUTRAN	Evolved UMTS Terrestrial Radio Access Network
GEO	Geosynchronous Equatorial Orbit
HTTP	HyperText Transfer Protocol
ICT	Information and Communications Technology
IoT	Internet of Things
IoT-vGW	IoT virtual Gateway
IP	Internet Protocol
JANET	Joint Academic NETwork
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MAC	Medium Access Control
MEC	Multi-access Edge Computing
mMTC	massive Machine Type Communications
MQTT	MQ Telemetry Transport
N3IWF	Non-3GPP Interworking Function
NB-IoT	Narrow Band IoT

NFV	Network Function Virtualization
NFVI	NFV Infrastructure Management
NFVO	NFV Orchestrator
NMS	Network Management System
NR	New Radio
OSM	Open-Source MANO
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
REST	Representational State Transfer
SA	Standalone
SDN	Software-Defined Networking
SMF	Session Management Function
SNMP	Simple Network Management Protocol
TTN	TheThingsNetwork
UDP	User Datagram Protocol
UE	User Equipment
UPF	User Plane Function
VIM	Virtualization Infrastructure Manager
VLAN	Virtual Local Area Network
vLEO	very Low Earth Orbit
VM	Virtual Machine
VNFM	Virtual Network Function Manager
VPN	Virtual Private Network
WAN	Wide Area Network
WIM	WAN Infrastructure Manager
WP	Work Package
WSAM	WiFi Slice Analytics Monitor
WSC	WiFi Slice Controller
WSMP	WiFi Service Management Platform
XML	Extensible Markup Language

List of Figures

List of Tables

Executive Summary

This document provides an overview of the activities regarding the evolution of the 5GENESIS Surrey Platform. The main aim of the platform (as part of the 5GENESIS facility), which is hosted in the 5G & 6G Innovation Centre (5G & 6GIC) at the University of Surrey, UK, is to demonstrate the support of massive Internet of Things (IoT) and multimedia communications in a networking environment consisting of multiple (3GPP and non-3GPP) Radio Access Technologies (RATs).

More specifically, the 5GENESIS Surrey Platform comprises a multitude of mobile network technologies, both 3GPP and non-3GPP. During the third Phase of the 5GENESIS project, Commercial Off The Shelf (COTS) 5G New Radio (NR) solutions were integrated into the Radio Access Network (RAN), as part of a larger flexible 5G network infrastructure. The Surrey Platform RAN also supports Narrow Band IoT (NB-IoT) Rel.15, as well as WiFi (802.11ac), integrated using the Non-3GPP Interworking Function (N3IWF), and LoRA Wide Area Network (WAN) technologies. The 5G Core (5GC) developed is Rel.16 Standalone (SA) compliant. On the user side, the Platform uses the MONROE probes, which are extended with IoT capabilities, and tailored to the needs of the Surrey use case. Moreover, 5G Customer Premises Equipment (CPE) devices are also used. On the IoT side, the Surrey Platform uses micro-controller devices that feed the platform with data using different IoT protocols. These data are made interoperable by being mapped to User Datagram Protocol (UDP) data, using an IoT virtual Gateway (IoT-vGW).

This document is the is the final part of the series, describing the evolution of the 5GENESIS Surrey platform, which is an instantiation of the 5GENESIS platform blueprint. At the end of each integration Phase, a testing and validation cycle follows, providing a demonstration of vertical use cases allowing specific KPI evaluation based on relevance to the use case. For the Surrey platform, a specific massive Machine Type Communications (mMTC) related use case will be executed at the university campus premises.

The aim of Phase 1 was to deliver a pre-5G network infrastructure with Network Function Virtualization (NFV)/ Software-Defined Networking (SDN) capabilities, support for heterogeneous RATs and the provision of edge computing capabilities. To that end, Phase 1 focused on the deployment of the Rel. 15 4G core and radio components, i.e., Evolved Packet Core (EPC) and Evolved UMTS Terrestrial Radio Access Network (EUTRAN), as well as the mmWave backhaul network and the Open-Source MANO (OSM)-based Infrastructure.

The aim of Phase 2 was to complete the integration of the Release A of coordination layer components, coming from WPs 3 and 4, and to prepare for Phase 3 demonstration of the Surrey use case.

Following the approach adopted by the 5GENESIS project, the Surrey platform components and technologies (adopted in Phase 3) are described in this document. Coordination layer remains a common layer for all platforms and thus, related components have been integrated as they have become available through project partners. Management and Orchestration (the ETSI-compliant OSM) components were already implemented and integrated during Phase 1, while infrastructure layer comprises multiple components and covering different RATs, together with the 5GC that has been developed in-house, in Surrey, continue to be enhanced and upgraded.

More specifically, at the end of Phase 2, the following milestones have been achieved:

- Integration of the Release B of the 5GENESIS Facility components is successfully completed.
- Final testing of INFOLYSiS vGW under heavy load of data and performing the appropriate optimizations where needed.
- Extension of the INFOLYSiS vGW Application Programming Interface (API) to support more queries and variables in order to disseminate IoT data more efficiently.
- Deployment and configuration of all planned IoT sensor nodes and emulation of the remaining traffic virtually in order to execute the massive IoT and multimedia communication use case.
- Testing of an mMTC slice instantiation (in parallel with enhanced Mobile Broadband (eMBB) slice) is scheduled for October 2021.
- Indoor/Outdoor FON WiFi Access Point (AP) deployment and integration testing with Surrey 5GC - FON and Surrey 5GC support N3IWF related interfaces. The development of the N3IWF interfaces (on the User Equipment (UE) & core-network sides) by both FON and Surrey teams is completed. The deployment of the FON APs in the Surrey premises, and the integration testing of N3IWF interfaces are also completed.
- Deployment and testing of the Adaptive Policy Execution (APEX) policy management solution and analytics is re-scheduled for Phase 3.
- Final testing of the Surrey Platform use case is scheduled for October 2021.

The Surrey Platform will be used for the validation and testing of the main IoT showcasing event, and the report on the Key Performance Indicators (KPIs) measured will be available in deliverable D6.3.

Table of Contents

1 INTRODUCTION

1.1 Purpose of the document

This is the third and final deliverable documenting the activities regarding the preparation, upgrading and operation of the Surrey Platform. This work is undertaken within the context of 5GENESIS Task 4.5, which is part of Work Package 4 (WP4) "End-to-End Instantiations of the Facility". The objective of this work is achieved by following the specifications defined in WP2, integrating the Facility components developed in WP3 according to the methodology set in WP5, and realising the necessary use-case specific extensions, in order to allow for experimentation and validation of the KPIs in WP6.

In order to allow the document readers to familiarise with the Surrey Platform and appreciate its capabilities in view of the expected showcasing, an overview of the Platform is provided. This deliverable provides information on the overall Platform infrastructure, Management and Orchestration, as well as the 5GENESIS Coordination Layer. As the main focus of the Platform is on massive IoT communications, emphasis is given on the IoT network, providing a detail description of its data sources, i.e., the sensing devices deployed and configured, and the mapping of the resulting data through an IoT-vGW in order to make them unified and interoperable for further use. The key Surrey Platform infrastructure in the main and edge data centres is also overviewed, while its connectivity with other platforms, as well as the nation-wide fibre network is also discussed. The mobile network technologies employed in all parts of the network, resulting in a multi-RAT environment supporting massive IoT and multimedia communications are described in detail.

The evolution of the Surrey Platform is also discussed, describing the details of the instantiation of the 5GENESIS architecture, while reporting on the accomplishments in Phase 3 of the 5GENESIS project.

Currently, the 5GENESIS project has released four main documents as part of the WP2 deliverables that are used as guidelines in order to define the 5GENESIS testbed specifications, as depicted in Table 1.

Table 1: WP2 deliverables

id	Document title	Relevance
D2.1 (5GENESIS Consortium, 2018)	Requirements of the Facility	The document sets the ground for the first set of requirements related to supported features at the testbed for the facilitation of the use cases.
D2.2 (5GENESIS Consortium, 2018)	5GENESIS Overall Facility Design and Specifications	The 5GENESIS facility architecture is defined in this document. The list of functional components to be deployed in each testbed is defined.
D2.3 (5GENESIS Consortium, 2018)	Initial planning of tests and experimentation	Testing and experimentation specifications that influence the testbed definition, operation and maintenance are defined.

D2.4 (5GENESIS Consortium, 2020)	Final report on facility design and experimentation planning	The final version of the 5GENESIS facility architecture is defined in this document.

The Surrey Platform focuses on validating KPIs related to latency, reliability, and coverage. To this end, the respective use case will demonstrate the support of massive IoT with heterogeneous data protocols and multimedia communications for large-scale public events. This document provides a description of the target deployment as well as an overview of the use-case specific platform extensions.

1.2 Structure of the document

The structure of this deliverable is as follows:

Section 2 provides an overview of the Surrey Platform. Starting with a high-level description of the Platform site, the mobile network technologies employed are summarised. Emphasis is on the IoT network, describing the sensing devices used, as well as the operations regarding the mapping of the IoT data provided by these devices to UDP data. Then, the Management and Orchestration Layer, as well as the 5GENESIS Coordination Layer in the Surrey Platform are also described.

Section 3 focuses on the evolution of the Surrey Platform, discussing on the instantiation of the 5GENESIS architecture and reporting on the accomplishments of Phase 3.

Section 4 describes the Surrey Platform massive IoT use case, providing a high-level view of the respective Platform extensions and topology.

Finally, Section 5 provides concluding remarks.

1.3 Target audience

This deliverable is a public document reporting on the Surrey Platform evolution during Phase 3 of the 5GENESIS project. Its target audience includes the Information and Communications Technology (ICT) professionals or research projects who are interested in performing experimentations in the Surrey Platform, the European Commission, who can use this document as a means for the evaluation of the activities of the Platform with regards to the project objectives, as well as the 5GENESIS consortium, who can use it as a guide and reference regarding future activities.

2 Surrey Platform Overview

This section provides an overview of the Surrey Platform. Starting with a high-level description of the Platform site, emphasis is given on the target deployment, summarising the Platform infrastructure in terms of mobile network technologies, data centres, and transport network. Since the focus of the Platform is the support of massive IoT communication, a detailed description of the Platform IoT network is provided. Finally, the Management and Orchestration Layer, as well as the 5GENESIS Coordination Layer in the Surrey Platform are described.

2.1 Platform Sites Topology

The 5G and 6G Innovation Centre (5G & 6GIC) testbed on the University of Surrey Campus in Guildford, UK, is the test site hosting the "Surrey Platform". The current outdoor deployment at the Surrey site is shown in Figure 1. The RED square indicates the location of the 5G & 6GIC building housing the 4G/5G (core) infrastructure and the area within the BLACK square indicates the geographical area where the Surrey Platform use case will be executed. The aim of the Surrey Platform is to demonstrate the support of massive Internet of Things (IoT) and multimedia communications in a multi-RAT environment using WiFi, LoRa, and Narrowband IoT (NB-IoT) access technologies. The different components provided by the Platform partners are integrated into the Surrey site.

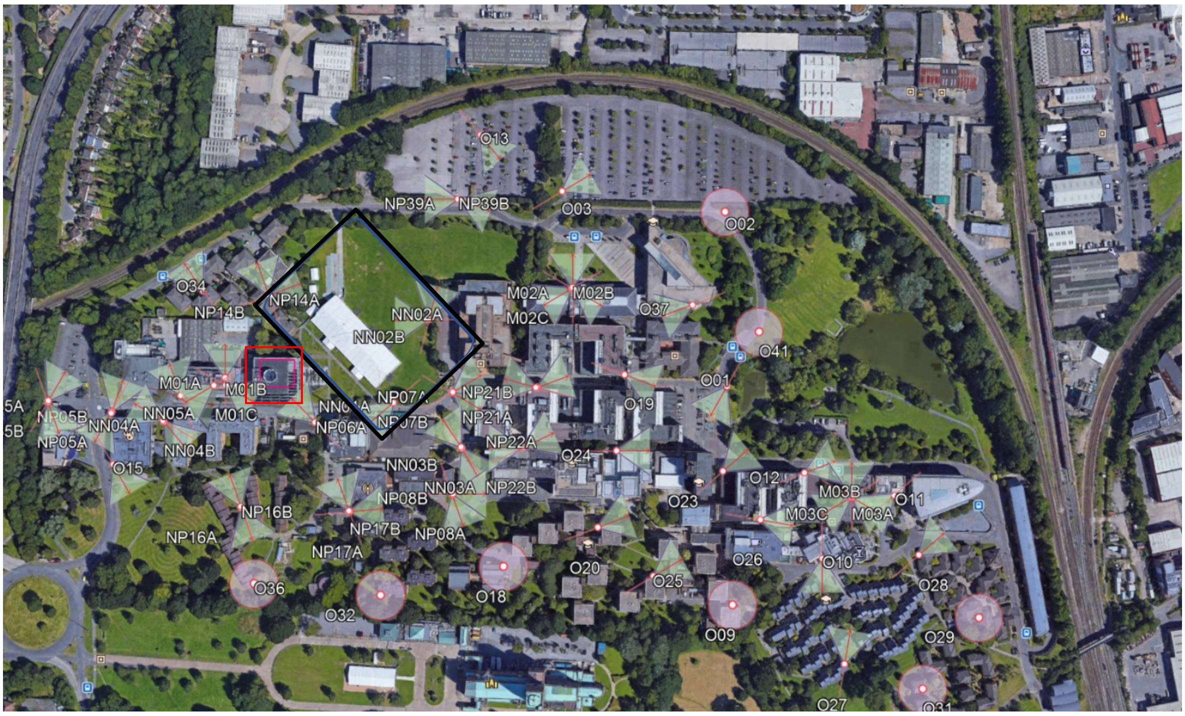


Figure 1: Network RAN deployment at the Surrey site

2.2 Platform Deployment Setups

The overall Surrey platform topology/deployment of the infrastructure components is illustrated in Figure 2:

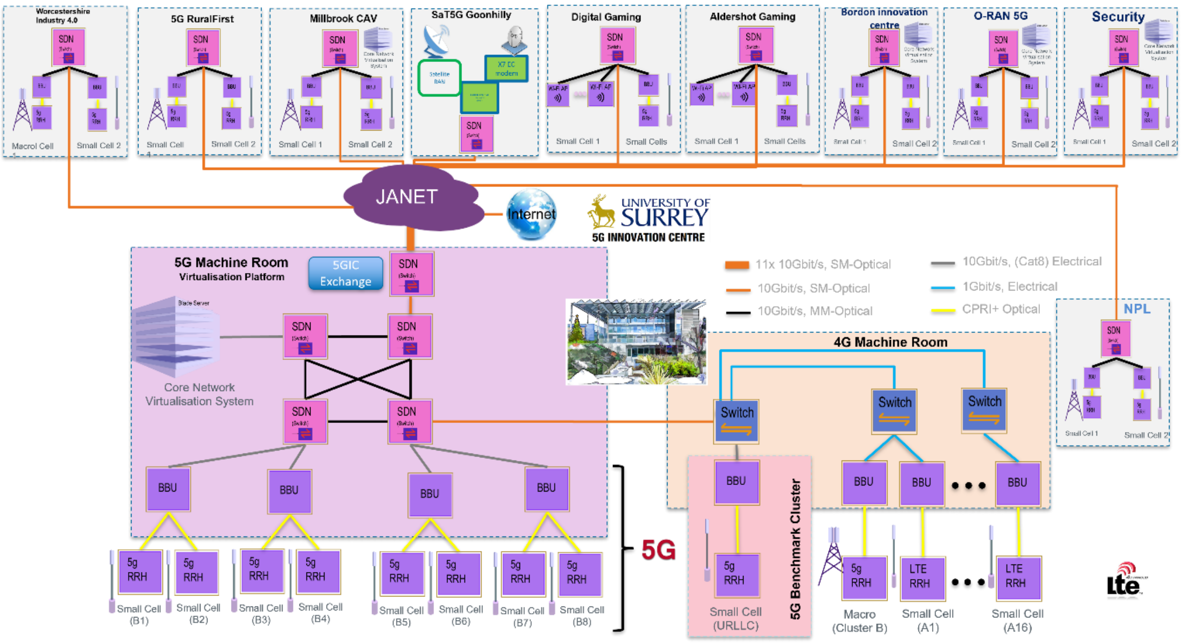


Figure 2: 5G & 6G IC Testbed UK National Network Connectivity Architecture

The Surrey platform is connected to the Joint Academic NETwork (JANET) network[1] with 10 x 10 Gbps aggregated fibre capacity and maintains connectivity to the data centre. Internally, the platform is comprised of a set of SDN switches to support dynamic traffic flow operations. This switching fabric connects the 5G RAN equipment to the virtualisation testbed, for virtual network services support to the users. Connectivity architecture is shown in Figure 2. The architecture includes connectivity between the newly developed 5G network data centre and the existing 4G data centre infrastructure, which will then make possible to showcase 4G-5G hand-over scenarios.

[1] https://www.jisc.ac.uk/janet

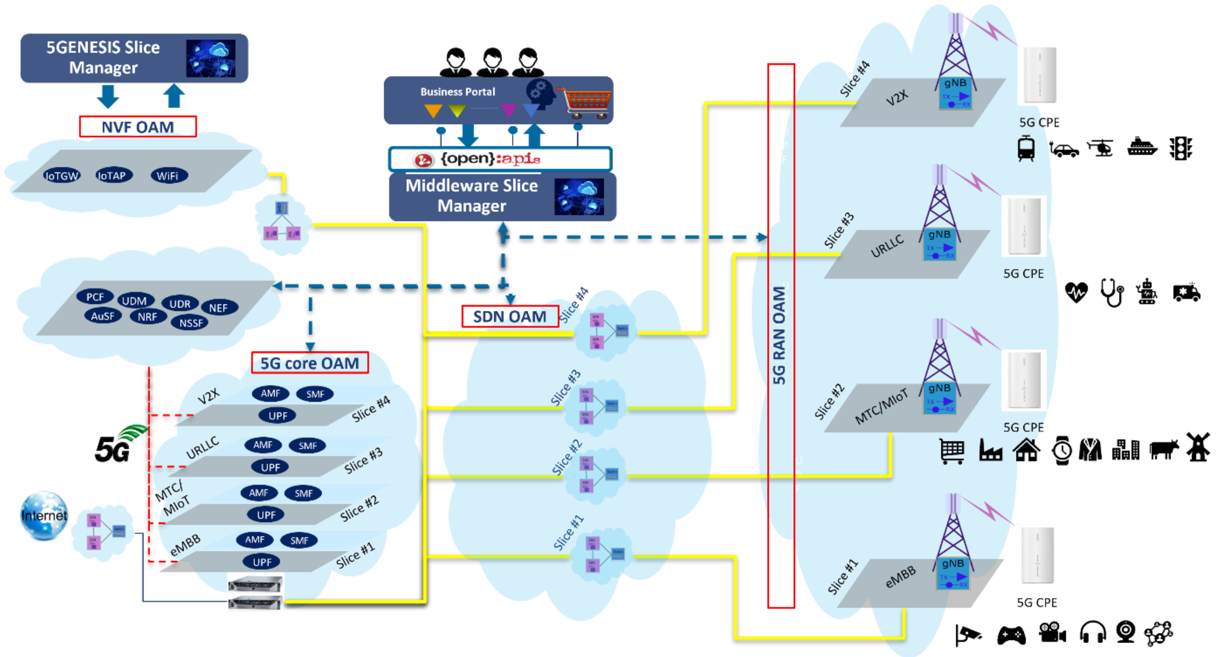


Figure 3: The Surrey platform "5GENESIS Island" – software architecture and components

Figure 3 shows the deployment strategy on multiple servers. The platform team has developed core network software, including but not limited to the following components: Access and Mobility Management Function (AMF), Session Management Function (SMF), and User Plane Function (UPF), and is closely following the developments in 3GPP, regarding the interface definitions and protocols to be used between the new 5G components.

2.2.1 The Surrey Platform IoT network

The Surrey Platform follows the 5GENESIS architecture. The Platform and main components are hosted in and around the premises of the 5G & 6G Innovation Centre at the University of Surrey, and within this area, connectivity will be via the 5GENESIS branch of the 5G & 6GIC carrier-grade testbed. The Surrey platform currently supports Rel. 16 5G SA. The workflow of the information within the platform is as follows: i) First, sensors deployed on site will collect sensing data including information about temperature, air pressure, acceleration, and other parameters. ii) These data are then collected and transmitted using one or more of the available air interfaces on campus (use-case designated arena) and iii) are then passed to the IoT-vGW that understands/translates the various incoming IoT protocols into UDP-over-Internet Protocol (IP) packets, and iv) forwards the data to the Surrey server.

The instantiation of the Surrey IoT solution requires the following two steps: i) Configuration and deployment of Surrey IoT sensors over the target demonstration venue and ii) Deployment and operation of INFOLYSiS IoT-vGW, followed by the setup of a dedicated mMTC slice on the 5GC.

2.2.1.1 IoT sensors

In order to feed sensor data towards the Surrey Platform (RAN + Core), Pycom sensor nodes are configured and used to support different protocols and radio interfaces. This section provides a description of those data sources, interfaces supported and details about how these IoT sensors produce and transmit data.

(i) Sensor data & device types

The FiPy/Pysense sensor allows capturing and sending the following information about its environment:

- boardID: Board Identifier such as: "uos-5gic-pysense20"
- entityType: "FiPy+Pysense"
- entityOwner: "UoS"
- 19atatype: "sensorData"
- data: the measurement payload, made of the following information:
    - Air Pressure
    - Air Temperature
    - Ambient light
    - Humidity
    - Accelerometer values (roll/pitch)

In addition, any packet sent over the air interface also include meta-data:

- Time stamp (synchronized with RTC server like http://time.google.com)
- Data_source : "WIFI", "LoRA" or "5G"

Table 2 describes the devices/nodes used. The format of data output is in JSON.

Table 2: Sensor devices & AP deployment in Surrey testbed (also see Table 3)

	Pycom Pysense	WiFi AP	CPE (5G)	IoT (LoRa) GWs
Planned (as in DoW)	30~50 sensors + load emulation	1	0	3
Actual	30	2 (5G & 6GIC and FON)	1	5

(ii) Pycom FiPy+Pysense board

The sensors used are manufactured by Pycom[2]. A sensor consists of an ultra-low power programming board <<**FiPy**>> (see Figure 4 below) equipped with an expansion shield <<**PySense**>> that provides a variety of classical sensors (see Figure 5 below):

- Ambient Light sensor (LTR-329ALS-01)
- Barometric pressure sensor (MLP3115A2)
- Humidity sensor (SI7006-A20)
- 3-axis 12 bits accelerometer (LIS2HH12)
- Temperature sensor (SI7006-A20)

[2] https://pycom.io/product/fipy/

Figure 4: <<FiPy>> programming board with radio

Figure 5: <<PySense>> expansion board

In addition, the board provides a USB port with serial access and features ultra-low power consumption (~1µA in deep sleep).

The boards have been programmed using the Pymakr plugin inside the Atom editor. The programming language used is micro-python. This dependency on micro-python introduced lots of technical burdens and instability as most of the libraries in current releases of micro-python (e.g., microcoapy.py, umqtt.py, urequests.py) are at an early stage of development with lots of remaining issues/bugs. However, the required bug-fixes have been developed and applied by the Surrey team and all sensor nodes are currently operational.

For these boards, the sampling rate is fixed and was set to 10s. Any other more suitable value can be set up. The data format used to convey the sensor data to the Surrey 5GENESIS platform follows the JavaScript Object Notation (JSON) data format.

In relation to radio interfaces, the Pycom <<FiPy>> module supports:

- WiFi: The communication is established -via a dedicated access point- to the 5G WiFi network;
- Dual LTE-M (NB-IoT & CAT-M1): The device supports NB-IoT release 13 while the Surrey Platform supports release 15 currently. In order to demonstrate the use of 5G, 5G CPEs were used, which from the board perspective acts as a WiFi AP. Consequently, despite

the IoT payloads actually "fly" to the CPE using WiFi, the same payload is then transferred to the 5GENESIS testbed using 5G;

- LoRA: A communication is established between the LoRA radio module of the PySense board and TheThingsNetwork (TTN) Company located in the Netherlands using LoRA gateways, deployed over Surrey campus. When the data has reached TTN, it is made available to any client accessing TTN with MQ Telemetry Transport (MQTT) subscriber using credentials like the application ID and application key (which were both already used in addition to the device key to send the data). When the communication is established, sockets are used to send the data;
- Bluetooth: not used/considered for the surrey use-case;
- Sigfox: a competitor to LoRa, also not used/considered for the surrey use-case;

The data format/structure used can be found in Appendix 2 of [1].

(iii) Protocol/Radio channel combinations

The collected data record (in JSON structure) is pushed to the servers using different protocols, and using different radio channels as follows:

- Sockets over LoRa
  The data is collected and sent to The Things Network (TTN) servers located in the Netherlands using LoRa gateways. The Surrey Platform server then will connect to TTN using MQTT in order to receive and handle locally the data (including storage).
  A preliminary mock-up was made during the previous phase and was fully documented in the previous version of this deliverable (please refer to Deliverable D4.11 Section 2.2.1 for more detail and illustrations [1]). This mock-up was built using 1/ IBM's nodeRed for querying data from TTN using MQTT, 2/ influxDB for data storage in the form of time series and finally 3/ Grafana for graphical display of the sensor reading time series, one for each piece of data in the payload.

HyperText Transfer Protocol (HTTP) Push over WiFi / 5G: data is pushed to the Surrey platform HTTP server using a HTTP POST using the micro-python urequests.py library. Only the WiFi part has been implemented at the time of the document writing.

- Constrained Application Protocol (CoAP) over WiFi / 5G: data is pushed using a client.post() call from the microcoapy.py library. Only the WiFi part has been implemented at the time of the document writing.
- MQTT over WiFi / 5G: Data is published to the Surrey Platform MQTT queue along with the entityID topic where entityID= "uos-5gic-pysense<nbr>".

Figure 6: External fixed LoRa gateway

Figure 7: Internal portable LoRa gateway

2.3 Platform Implementations

2.3.1 Platform Infrastructure Layer

2.3.1.1 Main Data Centre

The core cloud domain currently consists of the following rack servers: 1x R430 & 2x R330 & 1x R640 & 1x R920. OpenStack runs on a Dell R640 server (72Core / 512GB / 2TB; Ubuntu 18.04). A Corsa Software Defined Network (SDN) switch (Mellanox SN2100, 16 port switch) runs on one Dell R640 server. OpenFlow 1.3 protocol is supported. The supported controllers include

ODL, ONOS, FloodLight, RYU, etc. The switch throughput is 3.2Tb/s. There is also support for remote Virtual Private Network (VPN) access.

Figure 8: The 5G & 6GIC main data centre

2.3.1.2 Edge Data Centre

The 5GENESIS dedicated edge cloud has a data centre architecture, currently consisting of 4 Dell R640 servers, which are either based on bare metal machines or deployed within virtual machines (VMs). For security purposes, these servers are attached to different virtual local area networks (VLANs) under the Surrey Platform 4G/5G core networks.

2.3.1.3 Transport Network

The Surrey Platform backbone network comprises an SDN architecture, supporting network virtualization slicing and traffic engineered paths. Quality of Service (QoS) and bandwidth allocation are supported across the infrastructure.

(a) Inter-platform and inter-site connection via JANET/OpenVPN

The JANET backbone network combines high bandwidth and low latency features providing a wide range of services. A transport network capacity of 10 Gbps throughput is available for traffic to and from the Surrey Platform via JANET backbone.

(b) DWDM access to nation-wide fibre network

In addition to the terrestrial backhaul network link provided by JANET, 5G & 6GIC is connected to a Dense Wavelength Division Multiplexing (DWDM) connection over a UK-wide fibre core network. That allows the provision of high throughput connectivity between research and industry sites, and allows experimentations involving an operator-grade long-distance metropolitan network.

(c) mmWave-based backhaul network

The Bluwireless[3] mmWave carrier-grade backhaul (P2P mesh connectivity) solution for 4G/5G, already deployed at the Surrey campus, enables interference mitigation and seamless co-existence. The products are based on IEEE 802.11ad/ay standard. Bluwireless mmWave technology can deliver data rates of multiple Gb/s and ranges over 300m.

2.3.2 Mobile Network Technology

2.3.2.1 Radio Access

The radio access part of the Surrey Platform comprises different Radio Access Technologies both 3GPP and non-3GPP.

Specifically, Commercial off the Shelf (COTS) 5G New Radio (NR) solutions developed for 5G are integrated as part of a larger flexible 5G network infrastructure and will allow support for a wide range of 5G use cases empowered by network slicing in the scope of 5GENESIS. Moreover, Rel.15-compliant software upgrade (from HUAWEI) to support NB-IoT is already available and deployed at the Surrey site (campus-wide). The WiFi (802.11ac) deployment is based on a series of Ruckus APs interconnected to the Surrey Platform 5GC following the 3GPP Release 16.

The LoRa devices integrated and used in the Surrey Platform serve as both sensor nodes that can be connected via (5G Ues acting as) gateways, as well as another set of non-3GPP access technology exploiting unlicensed spectrum to support and facilitate operation and communication of non-mission critical IoT deployments. Using the LoRaWAN protocol (i.e., LoRaWAN is a Medium Access Control (MAC) protocol for wide area networks) will provide complementing coverage for Machine Type Communication (MTC) in dense urban area deployments.

The Radio Access Technologies deployed at the Surrey Platform are summarised in Table 3.

Table 3: Summary of Radio Access Technologies deployed at the Surrey Platform

Site Type	# Sites	# Cells	Access Type
Outdoor 2x Sector	36	58	LTE-A, 9 of which also support NB-IoT
Outdoor Omni	8	8	LTE-A
Indoor Lampsites	6	6	LTE-A
Outdoor 1x & 2x Sector	7	9	5G-eMBB
Outdoor 1 Sector	1	1	5G-URLLC
Outdoor Omni	1	1	700MHz – LTE-A
Indoor AP	6	6	Wi-Fi
Outdoor GW	3	3	LoRa GW

[3] https://www.bluwireless.co.uk/

Figure 9: 5G NR Coverage at the Surrey Platform

2.3.2.2 Mobile Core

Surrey's 5G Innovation Centre (5GIC) used the Massive MIMO 64x64 5GNR at 3.5GHz spectrum with 100 MHz bandwidth present around its campus and 5G Core that was wholly built in the UK by the 5GIC R&D team. The Surrey team used a commercially available 5G capable mobile phone to test the end-to-end performance of its 5GSA.

The in-house built 5G mobile core network is compliant to 3GPP Release 15 and Release 16 standards. The state-of-the-art core is a fully virtualised solution that can be used on any cloud-native platforms. It is also capable of network slicing, and was previously tested with Geosynchronous Equatorial Orbit (GEO) and very Low Earth Orbit (vLEO) satellite communications systems.

2.3.2.3 User Equipment (UE)

Two physical dual-node MONROE probes have been deployed as part of the Surrey Platform. In contrast to the original MONROE hardware probes [2], and in support of the Surrey use case, the physical probes in Surrey have been extended with IoT capabilities. Further details on the MONROE IoT extension for the Surrey Platform are provided in the Appendix.

Moreover, in order to allow the traffic of WiFi enabled devices (such as the MONROE probes, and IoT sensor nodes) to be routed through the Surrey 5G core network, two 5G CPE devices are also deployed in the 5G & 6GIC building.

2.3.3 Management & Orchestration Layer

An overview of the Management and Orchestration layer components and associated technologies deployed in the Surrey Platform is provided in Table 4.

Table 4: Management and Orchestration Layer components – Phase 3

Component	Product/Technology
Slice Manager	Developed within the frame of the project to support network slice deployment over different management domains, i.e., Radio (Edge/Core), Cloud (NFV, Multi-access Edge Computing – MEC) and Network (SDN). *NB. API access can be provided from platform SDN controller to the slice manager.*
NFV orchestrator	The platform supports OSM v5.
Network Management System (NMS)	NMS is a composition of different tools/managers: – WAN Infrastructure Manager (WIM)[4] – Element Management System (EMS) for the 4G/5G Radio and Core components
NFV Infrastructure Management	NFVI/VIM based on OpenStack (at the edge and the core)
WAN/VIM	SDN transport

The Management and Orchestration Layer of the 5GENESIS architecture contains three main components:

- NFV MANO
  The Surrey Platform implements the NFV MANO functionally via Open Source MANO (OSM) and OpenStack. Amongst the key components of the NFV MANO, i.e., the NFVO, the Virtual Network Function Manager (VNFM) and the Catalogue, there is one to one mapping to the main components of OSM. The Virtualization Infrastructure Manager (VIM) is provided by OpenStack, the standard de-facto VIM in the ETSI NFV specification. Additionally, it is worth mentioning that the communication between NFV Orchestrator (OSM) and the VIM (OpenStack) is realized by a VIM Driver, which makes the NFVO transparent to OpenStack and enables OSM to manage multiple OpenStack instances at the same time.
- NMS
  The NMS is a platform-specific network management system with direct access to physical resources as well as configuration interfaces. In the Surrey Platform, the NMS provides an overview of the physical resources and an interface to manage them. The management of the resources is provided by the Resource Manager through the network and the inventory repository. The EMS, included in the NMS and responsible for the management of a PNF/VNF, is provided by a component in OSM.

[4] In the reference architecture the VIM in located within the NFV Orchestrator. And the WIM is the one located in the NMS.

- Slice Manager
  The Slice Manager is a common component that is instantiated in all 5GENESIS platforms and its final release was developed within WP3, specifically in Task 3.2.

2.3.3.1 APEX

Adaptive Policy Execution (APEX), previously described in D4.11 [1], is a lightweight engine for the execution of APEX policies. APEX policies can be designed for the straightforward execution of a single task or expanded into a larger complex model of multiple tasks and states. APEX policies can be designed to self-adapt through the dynamic selection of tasks influenced through the onboard context information of the policy driven entity.

APEX policies are triggered by an incoming event on a supported interface. The incoming event is matched with a policy starting a sequence of executions to produce an outgoing event. This sequence of executions is influenced by context information store on the APEX engine. The events sent and received by APEX are in JSON or Extensible Markup Language (XML) format. APEX supports a range of deployment options and interfaces for communication.

2.3.3.1.1 Flexible Deployment and Clustering

APEX has a variety of deployment options available due to the design and flexibility of the engine. It can be deployed for an interface/class, application, component, as a service, in a control loop or on cloud compute nodes.

APEX also has a variety of clustering options such as single source/target with single executor, multiple sources/targets with single executor, single source/target with multiple executors and multiple sources/targets with multiple executor instances through multithreading. Variations of these approaches can then be clustered further to create multiple non-multithreaded policy engine clusters, multiple multithreaded policy engine clusters or multiple mixed policy engine clusters. Further clustering can be used for the intelligent routing or optimisation of events handled in the cluster.

2.3.3.1.2 Interfaces

The interfaces of APEX are referred to as carrier technologies. These carrier technologies define how APEX receives and sends events. They can be used in any combination and any number of carrier technologies can be used for incoming and outgoing interfaces. APEX supports a range of carrier technologies, the most prevalent being Kafka and Representational State Transfer (REST).

KAFKA IO

Kafka IO is supported by the APEX Kafka plugin. More information can be found in Kafka documentation for Kafka Consumer[5] and Producer[6] classes. The Kafka carrier technology requires the least manual configuration due to the Kafka messaging system handling the successful communication of messages.

REST IO

The REST carrier technology supports a number of different approaches to handling APEX events. All APEX events using the REST interface must use JSON formatting, XML is not

[5] https://kafka.apache.org/090/javadoc/org/apache/kafka/clients/consumer/KafkaConsumer.html

[6] https://kafka.apache.org/090/javadoc/org/apache/kafka/clients/producer/KafkaProducer.html

supported for this interface. The most common REST interfaces used by APEX are the REST Client, REST Server and REST Requester. Each of these fulfil a specific purpose for the triggering of APEX policies and the communication of generated responses.

The Rest Client can handle both incoming and outgoing events. Incoming events are retrieved through polling. A HTTP GET request is repeated requiring timing to be handled by the server in the form of a timeout. Successful polls are used to trigger an appropriate policy and unsuccessful polls are recorded as a failure. Outgoing events are sent using a HTTP POST request on the conclusion of a policy. The destination of the event is set in APEX configuration and can be adapted at runtime.

The REST Server allows for synchronous communication with external components. Incoming events are sent to the REST Server through a HTTP GET request. The body of this request contains the event which is then used to trigger the appropriate policy. This approach requests a consumer producer pair. The linking of the consumer with a producer means the response to the incoming event can be generated in the APEX policy. Multiple consumer-producer pairs can be defined in the APEX configuration. Outgoing events are generated at the conclusion of the triggered body and sent in the response body of the initial HTTP GET request.

The REST Requester allows APEX to send a REST request and receive the reply as an incoming event without tying up APEX resources. This approach allows APEX to trigger a new policy as a result of a previous execution and can be used to link policy executions together. The location of the outgoing event can be adapted through policy logic allowing for dynamic reconfiguration.

2.3.4 Coordination Layer

The Coordination Layer is common for all 5GENESIS platforms. The corresponding software components are solely instantiated for each platform, i.e., they run independently, but in terms of functionality they provide the same set of features and functions.

The Experiment LifeCycle Manager (ELCM) in each platform provides a project-wide interface to the common 5GENESIS coordination layer components and site-specific instantiations of the lower layers of the architecture.

The components of the Coordination Layer that are common for all platforms within 5GENESIS are described for all platforms in Deliverable 2.4 [3] and therefore not included in this document.

Release B of the 5GENESIS Coordination Layer has been integrated into the “5GENESIS Island” of the Surrey Platform.

3 SURREY PLATFORM EVOLUTION IN 5GENESIS

3.1 Evolution Timeline

This section reports on the third Phase of development and integration within the Surrey platform, however, it is expected that the structure of this report will be common for all three phases. Figure 10 depicts the per-phase instantiation of the 5GENESIS architectural blueprint in the Surrey platform. It shows the functional blocks implemented and integrated during Phases 1 & 2, as well as the functionalities planned for integration in Phase 3.

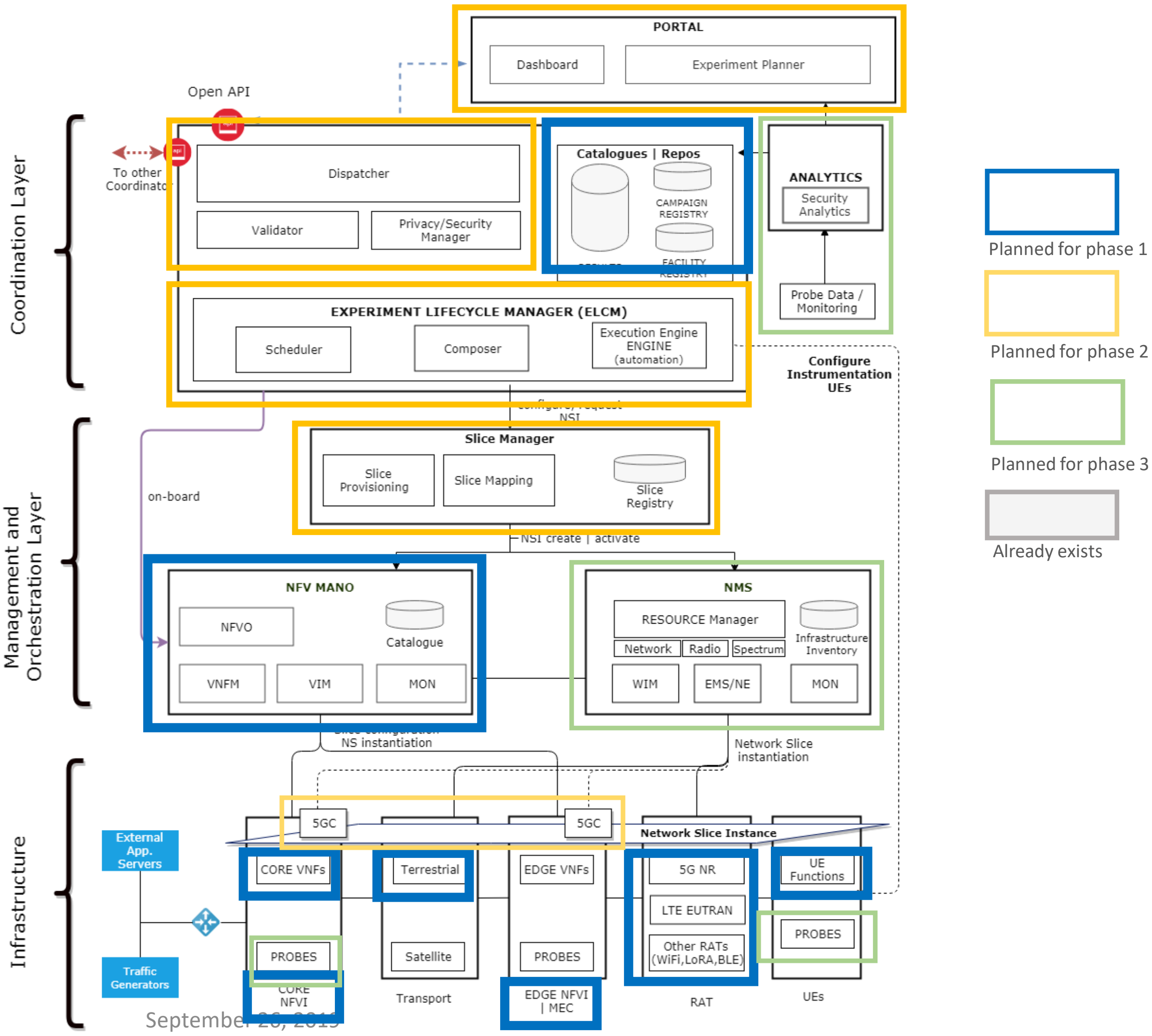


Figure 10: Per-phase instantiation of the 5GENESIS Architecture for the Surrey platform

During Phase 1, a first version of the IoT-vGW and two MONROE nodes were deployed (as well as early releases of a gNBs and a 5G enabled UE). As part of Phase 2, initial versions of the Slice Manager, ELCM, TAP tool, APIs, etc., have been integrated as well as the monitoring and analytics components. Moreover, experimentation with an initial/reduced-scope Surrey IoT use case has been successfully conducted.

The aim of Phase 2 was to deliver an end-to-end 5G (Rel.15 compliant) network which is NFV/SDN capable, also featuring edge computing capabilities. To that end, Phase 2 enhancements included: deployment of 5GC and radio components as well as the mmWave network and the NFV MANO and other infrastructure components – including the edge computing platform.

During Phase 3, the final releases of the Coordination layer components and the Portal were integrated and, together with full integration of the Slice Manager with MANO and 5GC components, they will be used to demonstrate end-to-end experiments, orchestration and lifecycle management over multiple network slices. The Surrey mMTC use case scenario will be fully deployed and demonstrated in this Phase.

Overall, the main components deployed during phases 1, 2 and 3 are summarized in Table 5 below.

Table 5: Phased deployments and configurations in the Surrey platform

Components	Technologies Deployed		
	Phase 1	Phase 2	Phase 3
Description	End-to-end testbed	End-to-end testbed	End-to-end testbed
Core Cloud	OpenStack Rocky	As in Phase 1.	As in Phase 1.
Edge Cloud	OpenStack Rocky	As in Phase 1.	As in Phase 1.
# Edge Locations	1	As in Phase 1.	As in Phase 1.
WAN/Network	Dual backhaul (satellite & terrestrial), OpenDaylight SDN controller	As in Phase 1.	As in Phase 1.
Slice Manager	N/A	Katana	Katana
MANO	OSM v5	As in Phase 1.	As in Phase 1.
NMS	Commercial HUAWEI	As in Phase 1.	As in Phase 1.
Monitoring	N/A	Prometheus + Ceilometer	Prometheus + Ceilometer
3GPP Technology	4G LTE + 5G NR (NSA) [both Rel. 15]	Upgrade to NB-IoT (Rel.15)	4G LTE and 5G NR SA Rel. 16
Non-3GPP Technology	LoRA 7 WiFi (802.11ac)	As in Phase 1.	As in Phase 1.
Core Network	5GIC in-house developed	As in Phase 1.	As in Phase 1.
RAN	4G:	As in Phase 1.	As in Phase 1.

	– Huawei (outdoors/indoors) – AirSpan (outdoors) – IP Access (indoors) 5G: – Huawei (outdoors)		
UE	COTS Cat.6	COTS Cat.12	5G CPE Pro version 2
IoT-vGW	INF GW (protocol converter)	As in Phase 1.	As in Phase 1.
IoT sensors	N/A	Various (Pycom, Arduino etc.)	Various (Pycom, Arduino etc.)
WIFI AP/AC	Pre-installed COTS indoor APs in 5GIC	FON indoors/outdoor APs + AC	FON indoors/outdoor APs + AC
Relevant Use Cases	UC1	UC1	UC1

3.2 Phase 3 Accomplishments

3.2.1 Integration of Coordination layer components

During Phase 3, the deployment and integration of the components that comprise Release B of the Open5GENESIS Suite took place. These include:

- The 5GENESIS Portal.
- The 5GENESIS Experiment Lifecycle Manager (ELCM) and its executive back-end, i.e., Open TAP.
- The 5GENESIS Slice Manager (Katana).
- The Dispatcher, that allows the uploading of the 5GENESIS experiment descriptors and VNFs.
- The Monitoring and Analytics component, including the storage (InfluxDB) and visualization (Grafana) components.

The above components were interconnected using the integration and testing procedures detailed in D5.2 [4]. This allows the configuration and execution of an experiment via the Portal, its automatization by the ELCM, collection of the measurements in the monitoring backend and, finally, their visualization.

All coordination layer components have been deployed as VMs in the OpenStack platform on BareMetal KVM Linux infrastructure in the Data Centre, with the exception of ELCM, which is implemented to run on windows platform. Current OpenStack platform supports only Linux-based applications hence the ELMC had to be deployed on a dedicated Windows server.

The following list of tools have been installed and configured at the Surrey 5GENESIS Platform:

Portal and ELCM

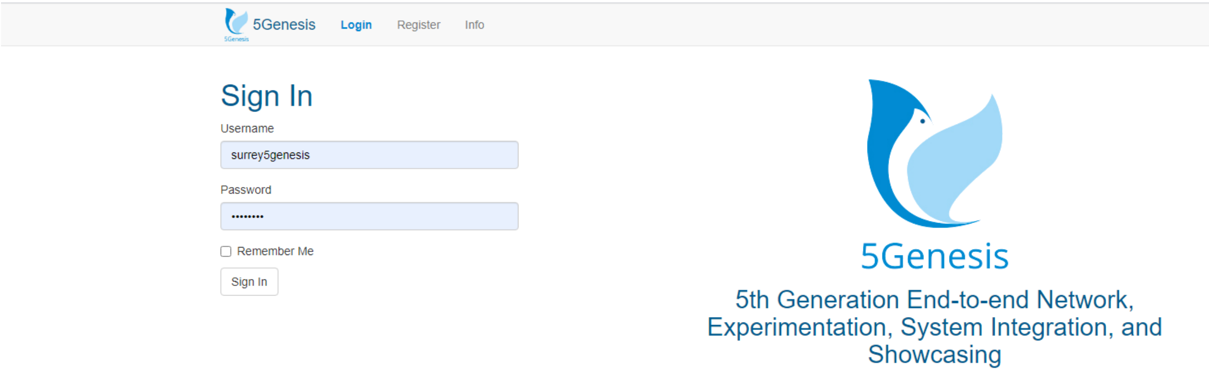


Figure 11: 5GENESIS Portal sign in page

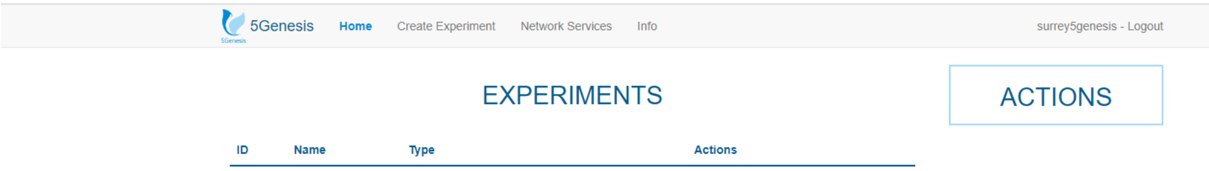


Figure 12: Portal – List of experiments

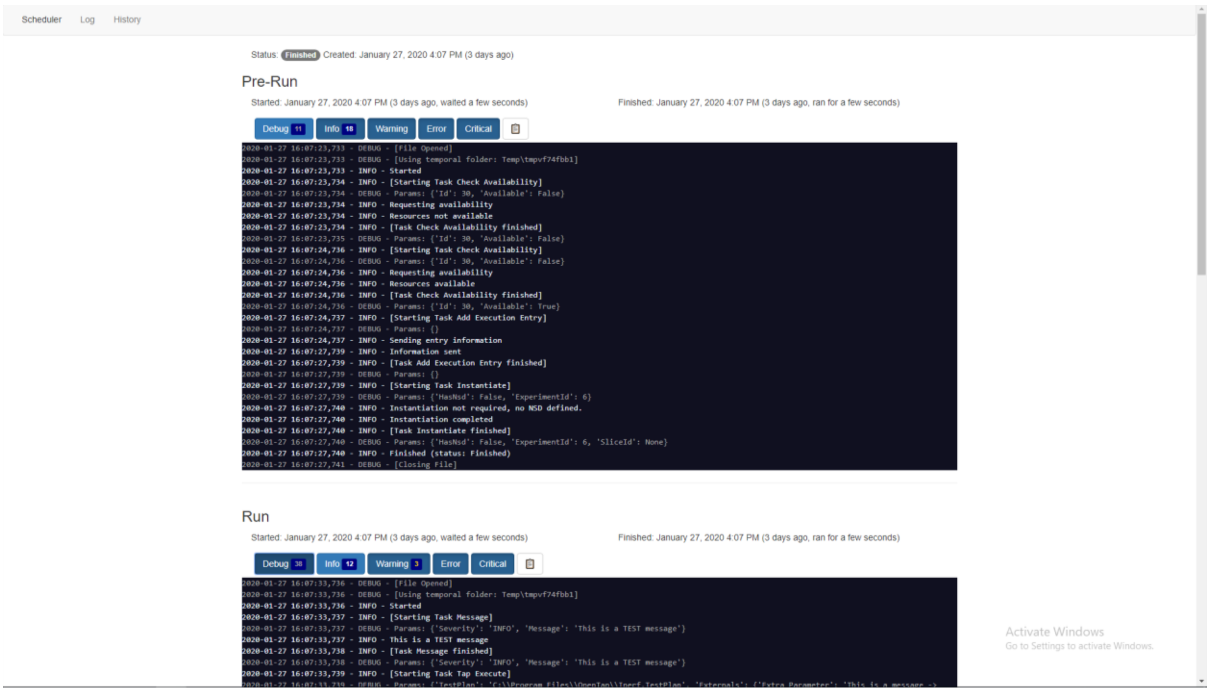


Figure 13: ELCM pre-run and run screens

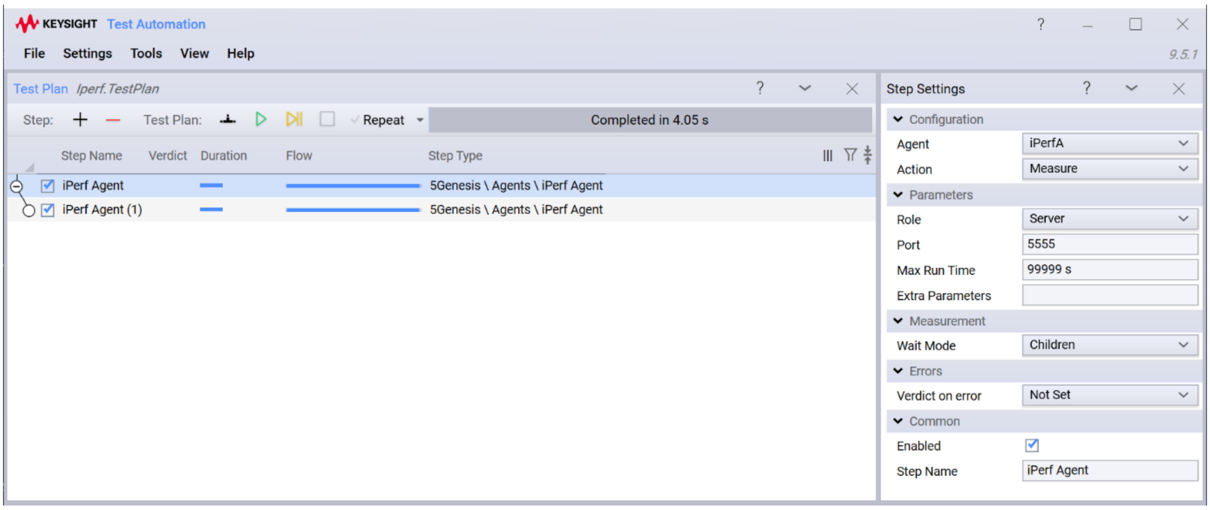


Figure 14: TAP iPerf test plan

Katana Slice Manager

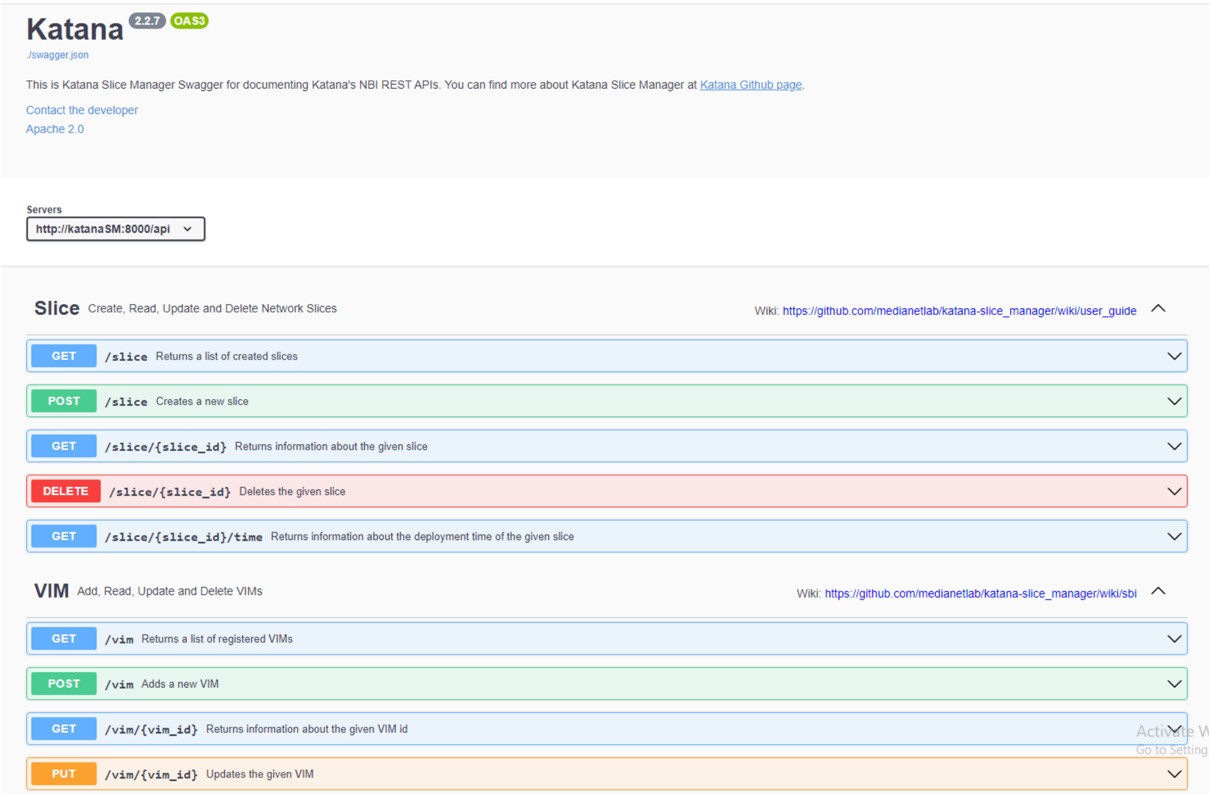


Figure 15: Katana Swagger Interface

Dispatcher

Figure 16: Dispatcher Swagger Interface

Storage (InfluxDB)

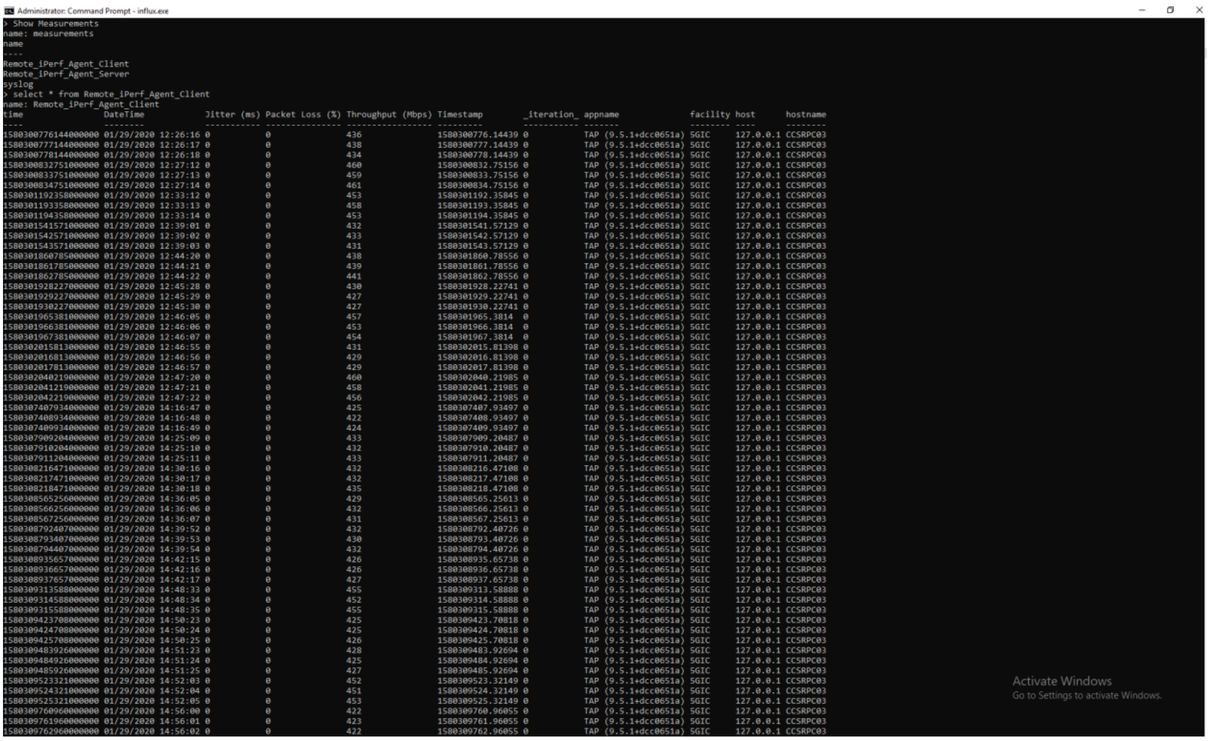


Figure 17: iPerf Client data stored in influxDB

Grafana (Visualisation)

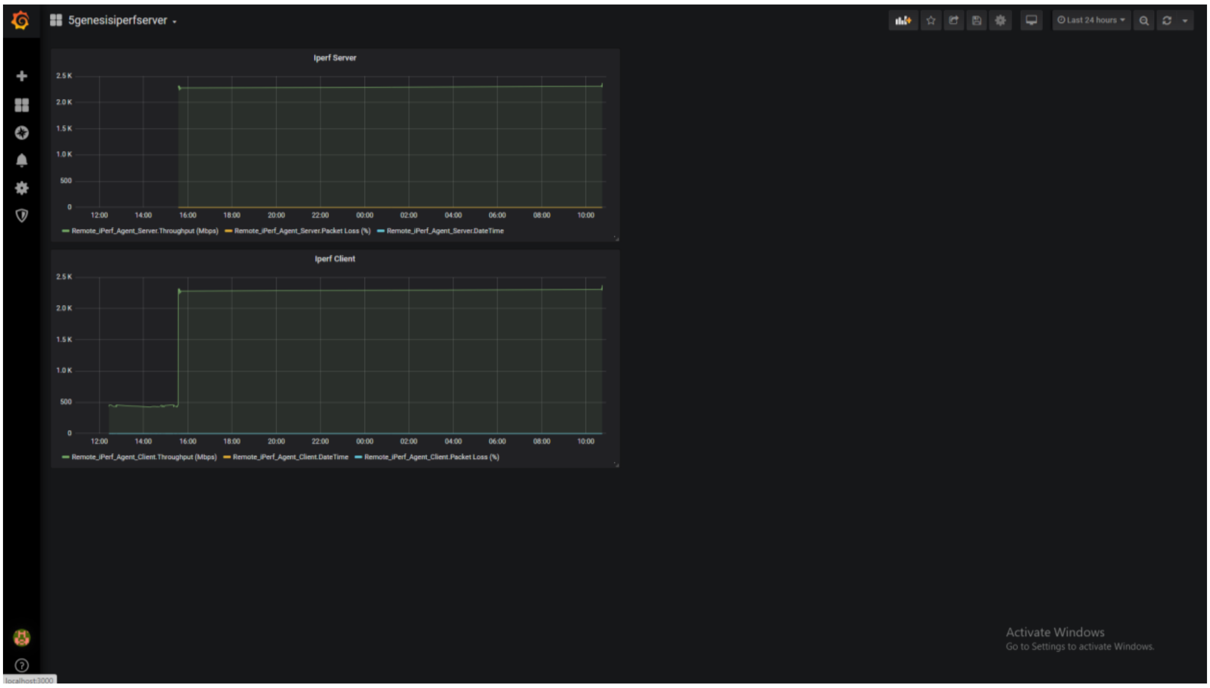


Figure 18: Grafana visualisation of iPerf Client and Server data

3.2.2 Integration of INFOLYSiS IoT-vGW and VNF components

This section describes the evolution of INFOLYSiS IoT-vGW and VNF components that happened during Phase 3 of the project. To avoid repetition from the corresponding section of D4.11 (that described the status for Phase 2), there is only a brief description of the general system architecture and functionality. Then the changes of Phase 3 are described in detail.

The system supports for IoT data interoperability all three of the following protocols: HTTP, CoAP and MQTT.

Implementation and integration of an API in INFOLYSiS vGW in order to enable data access both in real-time and for historical purposes have been performed. The API also supports accessing data with search filters.

The IoT data gathered by the sensing devices in the Surrey Platform are steered with SDN towards the INFOLYSiS system, in order to become interoperable and homogeneous. Depending on the network protocol (HTTP/CoAP/MQTT), the IoT data will be separated and sent to the appropriate mapping VNF that will make the translation from the original protocol to UDP. All UDP packets then will reach INFOLSYSiS IoT-vGW, where they will be stored in a database and become available for further usage. All interoperable IoT data from the INFOLYSiS IoT-vGW can be transmitted in real-time to other destinations using streams of UDP packets. Also, another way of accessing the interoperable IoT data is through the REST API that is available on the vGW.

The specification of each VNF is the same as in Phase 2. New changes and features that were accomplished in Phase 3 follow below in detail:

- INFOLYSIS CoAP Mapping VNF: Mapping of CoAP IoT data to plain UDP, providing interoperability. CoAP handling is done by using the Python programming language, considering the CoAPthon library until Phase B. In Phase 3, after extensive large-scale testing with IoT data flows, the CoAPthon library was changed by the aiocoap library, that handles CoAP packets more efficiently by using Python 3 native asyncio methods to facilitate concurrent operations and it is able to handle all the packets without any problems. The CoAP IoT data packets are accepted by the python script (Figure 19) and then instantly translated to UDP and sent to the vGW.

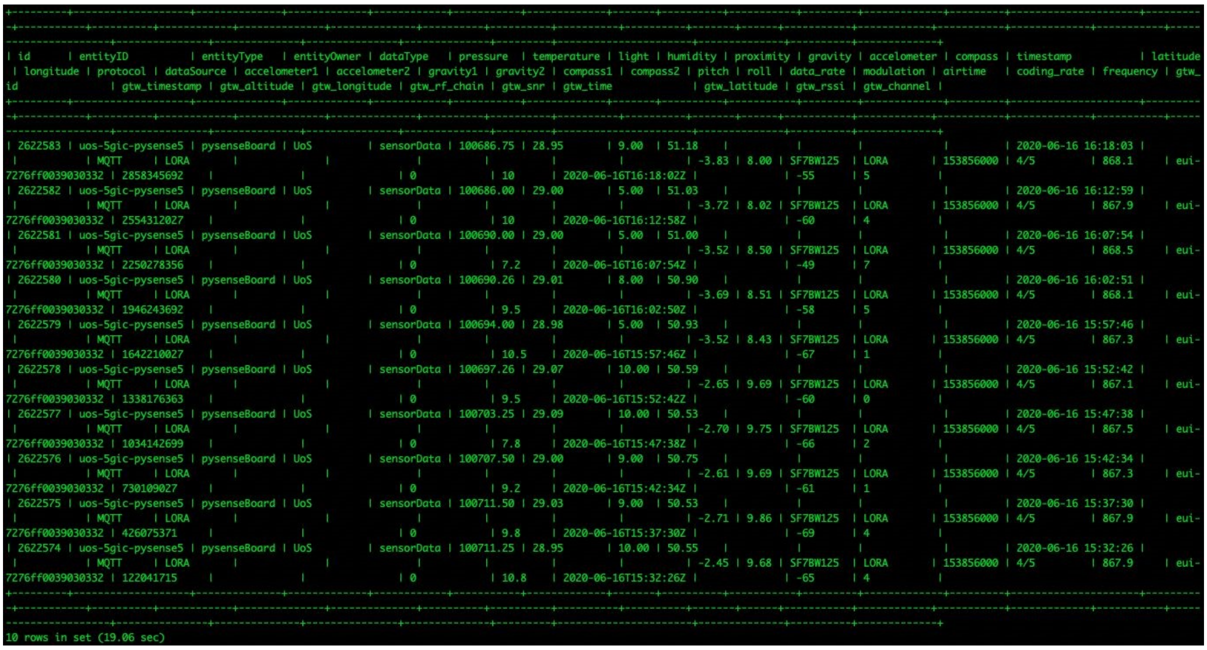

```
[+] Received COAP data and translating into plain UDP.
{"entityOwner": "UoS", "dataType": "sensorData", "entityType": "pysenseBoard", "entityID": "pysense4", "data": {"temperature": "30.242", "accelerometer": ["0.6695957", "", "-0.887
7645"], "light": "5", "proximity": "", "gravity": ["", "", ""], "humidity": "24.64346", "pressure": "101288.5", "compass": ["", "", ""]}, "metadata": {"latitude": "", "timestamp":
 "", "longitude": ""}}[+] Received COAP data and translating into plain UDP.
{"entityOwner": "UoS", "dataType": "sensorData", "entityType": "pysenseBoard", "entityID": "pysense4", "data": {"temperature": "30.25273", "accelerometer": ["0.7191858", "", "-0.7
760832"], "light": "5", "proximity": "", "gravity": ["", "", ""], "humidity": "24.64346", "pressure": "101291.5", "compass": ["", "", ""]}, "metadata": {"latitude": "", "timestamp
": "", "longitude": ""}}[+] Received COAP data and translating into plain UDP.
{"entityOwner": "UoS", "dataType": "sensorData", "entityType": "pysenseBoard", "entityID": "pysense4", "data": {"temperature": "30.242", "accelerometer": ["0.7779813", "", "-0.756
5151"], "light": "5", "proximity": "", "gravity": ["", "", ""], "humidity": "24.64346", "pressure": "101293.3", "compass": ["", "", ""]}, "metadata": {"latitude": "", "timestamp":
 "", "longitude": ""}}[+] Received COAP data and translating into plain UDP.
{"entityOwner": "UoS", "dataType": "sensorData", "entityType": "pysenseBoard", "entityID": "pysense4", "data": {"temperature": "30.23128", "accelerometer": ["0.6844367", "", "-0.7
41414"], "light": "5", "proximity": "", "gravity": ["", "", ""], "humidity": "24.65109", "pressure": "101294.0", "compass": ["", "", ""]}, "metadata": {"latitude": "", "timestamp"
: "", "longitude": ""}}
```


Figure 19: Translation log of CoAP IoT packets

- INFOLYSIS IoT-vGW: For Phase 3, the database model of data was extended in order to include more information about the IoT data. Also, information about radio connectivity, when available, was merged in the database table, so every IoT data point has additional information about the transport layer that was used. Figure 20 shows the new data structure of the table, where the interoperable IoT data is saved and Figure 21 shows 10 rows of data in that table.

Field	Type	Null	Key
id	int(10) unsigned	NO	PRI
entityID	tinytext	YES	
entityType	tinytext	YES	
entityOwner	tinytext	YES	
dataType	tinytext	YES	
pressure	tinytext	YES	
temperature	tinytext	YES	
light	tinytext	YES	
humidity	tinytext	YES	
proximity	tinytext	YES	
gravity	tinytext	YES	
accelometer	tinytext	YES	
compass	tinytext	YES	
timestamp	timestamp	NO	
latitude	tinytext	YES	
longitude	tinytext	YES	
protocol	tinytext	YES	
dataSource	tinytext	YES	
accelometer1	tinytext	YES	
accelometer2	tinytext	YES	
gravity1	tinytext	YES	
gravity2	tinytext	YES	
compass1	tinytext	YES	
compass2	tinytext	YES	
pitch	tinytext	YES	
roll	tinytext	YES	
data_rate	tinytext	YES	
modulation	tinytext	YES	
airtime	tinytext	YES	
coding_rate	tinytext	YES	
frequency	tinytext	YES	
gtw_id	tinytext	YES	
gtw_timestamp	tinytext	YES	
gtw_altitude	tinytext	YES	
gtw_longitude	tinytext	YES	
gtw_rf_chain	tinytext	YES	
gtw_snr	tinytext	YES	
gtw_time	tinytext	YES	
gtw_latitude	tinytext	YES	
gtw_rssi	tinytext	YES	
gtw_channel	tinytext	YES	

Figure 20: Database table structure in vGW



```
| id      | entityID           | entityType    | entityOwner | dataType    | pressure    | temperature | light | humidity | proximity | gravity | accelometer | compass | timestamp            | latitude
 | longitude | protocol | dataSource | accelometer1 | accelometer2 | gravity1 | gravity2 | compass1 | compass2 | pitch | roll | data_rate | modulation | airtime    | coding_rate | frequency | gtw_
id                  | gtw_timestamp | gtw_altitude | gtw_longitude | gtw_rf_chain | gtw_snr | gtw_time             | gtw_latitude | gtw_rssi | gtw_channel |
| 2622583 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100686.75 | 28.95       | 9.00  | 51.18    |           |         |             |         | 2020-06-16 16:18:03 |
 |           | MQTT     | LORA       |              |              |          |          |          | -3.83 | 8.00 | SF7BW125  | LORA       | 153856000 | 4/5         | 868.1     | eui-
7276ff0039030332 | 2858345692 |          |               | 0            | 10      | 2020-06-16T16:18:02Z |              | -55      | 5           |
| 2622582 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100686.00 | 29.00       | 5.00  | 51.03    |           |         |             |         | 2020-06-16 16:12:59 |
 |           | MQTT     | LORA       |              |              |          |          |          | -3.72 | 8.02 | SF7BW125  | LORA       | 153856000 | 4/5         | 867.9     | eui-
7276ff0039030332 | 2554312027 |          |               | 0            | 10      | 2020-06-16T16:12:58Z |              | -60      | 4           |
| 2622581 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100690.00 | 29.00       | 5.00  | 51.00    |           |         |             |         | 2020-06-16 16:07:54 |
 |           | MQTT     | LORA       |              |              |          |          |          | -3.52 | 8.50 | SF7BW125  | LORA       | 153856000 | 4/5         | 868.5     | eui-
7276ff0039030332 | 2250278356 |          |               | 0            | 7.2     | 2020-06-16T16:07:54Z |              | -49      | 7           |
| 2622580 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100690.26 | 29.01       | 8.00  | 50.90    |           |         |             |         | 2020-06-16 16:02:51 |
 |           | MQTT     | LORA       |              |              |          |          |          | -3.69 | 8.51 | SF7BW125  | LORA       | 153856000 | 4/5         | 868.1     | eui-
7276ff0039030332 | 1946243692 |          |               | 0            | 9.5     | 2020-06-16T16:02:50Z |              | -58      | 5           |
| 2622579 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100694.00 | 28.98       | 5.00  | 50.93    |           |         |             |         | 2020-06-16 15:57:46 |
 |           | MQTT     | LORA       |              |              |          |          |          | -3.52 | 8.43 | SF7BW125  | LORA       | 153856000 | 4/5         | 867.3     | eui-
7276ff0039030332 | 1642210027 |          |               | 0            | 10.5    | 2020-06-16T15:57:46Z |              | -67      | 1           |
| 2622578 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100697.26 | 29.07       | 10.00 | 50.59    |           |         |             |         | 2020-06-16 15:52:42 |
 |           | MQTT     | LORA       |              |              |          |          |          | -2.65 | 9.69 | SF7BW125  | LORA       | 153856000 | 4/5         | 867.1     | eui-
7276ff0039030332 | 1338176363 |          |               | 0            | 9.5     | 2020-06-16T15:52:42Z |              | -60      | 0           |
| 2622577 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100703.25 | 29.09       | 10.00 | 50.53    |           |         |             |         | 2020-06-16 15:47:38 |
 |           | MQTT     | LORA       |              |              |          |          |          | -2.70 | 9.75 | SF7BW125  | LORA       | 153856000 | 4/5         | 867.5     | eui-
7276ff0039030332 | 1034142699 |          |               | 0            | 7.8     | 2020-06-16T15:47:38Z |              | -66      | 2           |
| 2622576 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100707.50 | 29.00       | 9.00  | 50.75    |           |         |             |         | 2020-06-16 15:42:34 |
 |           | MQTT     | LORA       |              |              |          |          |          | -2.61 | 9.69 | SF7BW125  | LORA       | 153856000 | 4/5         | 867.3     | eui-
7276ff0039030332 | 730109027  |          |               | 0            | 9.2     | 2020-06-16T15:42:34Z |              | -61      | 1           |
| 2622575 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100711.50 | 29.03       | 9.00  | 50.53    |           |         |             |         | 2020-06-16 15:37:30 |
 |           | MQTT     | LORA       |              |              |          |          |          | -2.71 | 9.86 | SF7BW125  | LORA       | 153856000 | 4/5         | 867.9     | eui-
7276ff0039030332 | 426075371  |          |               | 0            | 9.8     | 2020-06-16T15:37:30Z |              | -69      | 4           |
| 2622574 | uos-5gic-pysense5 | pysenseBoard | UoS         | sensorData | 100711.25 | 28.95       | 10.00 | 50.55    |           |         |             |         | 2020-06-16 15:32:26 |
 |           | MQTT     | LORA       |              |              |          |          |          | -2.45 | 9.68 | SF7BW125  | LORA       | 153856000 | 4/5         | 867.9     | eui-
7276ff0039030332 | 122041715  |          |               | 0            | 10.8    | 2020-06-16T15:32:26Z |              | -65      | 4           |
10 rows in set (19.06 sec)
```

Figure 21: Interoperable IoT data saved in the database in vGW

These IoT data are available to third parties for future extensions and integrations using UDP streams and a RESTful API. The RESTful API that was developed during Phase 2 is described in detail in D3.6 Annex 3 – IoT Management and Monitoring [5].

In Phase 3, extensions were made in the RESTful API in order to accommodate complex queries and fetch data based on multiple criteria. Both the overview and an example execution of the GET request that was implemented is shown in Figure 22 and Figure 23, respectively. The request now can have dynamic number of requirements based on the parameter "complexity". In Figure 24 and Figure 25, the parameters of a request with complexity equal to 3 are depicted. More specifically, for every part of the query (3 parts in this example), three additional parameters are introduced in the API request: "measurement", "operator" and "value". Each and every part by itself works in the same way as the request from Phase 2, but now in Phase 3 those parts are combined based on the parameter "complexityoperator", which can have two different values, i.e., AND or OR, the logical operators.

The operators in each part of the query were also improved by adding support for greater and equal (ge) and less and equal (le).

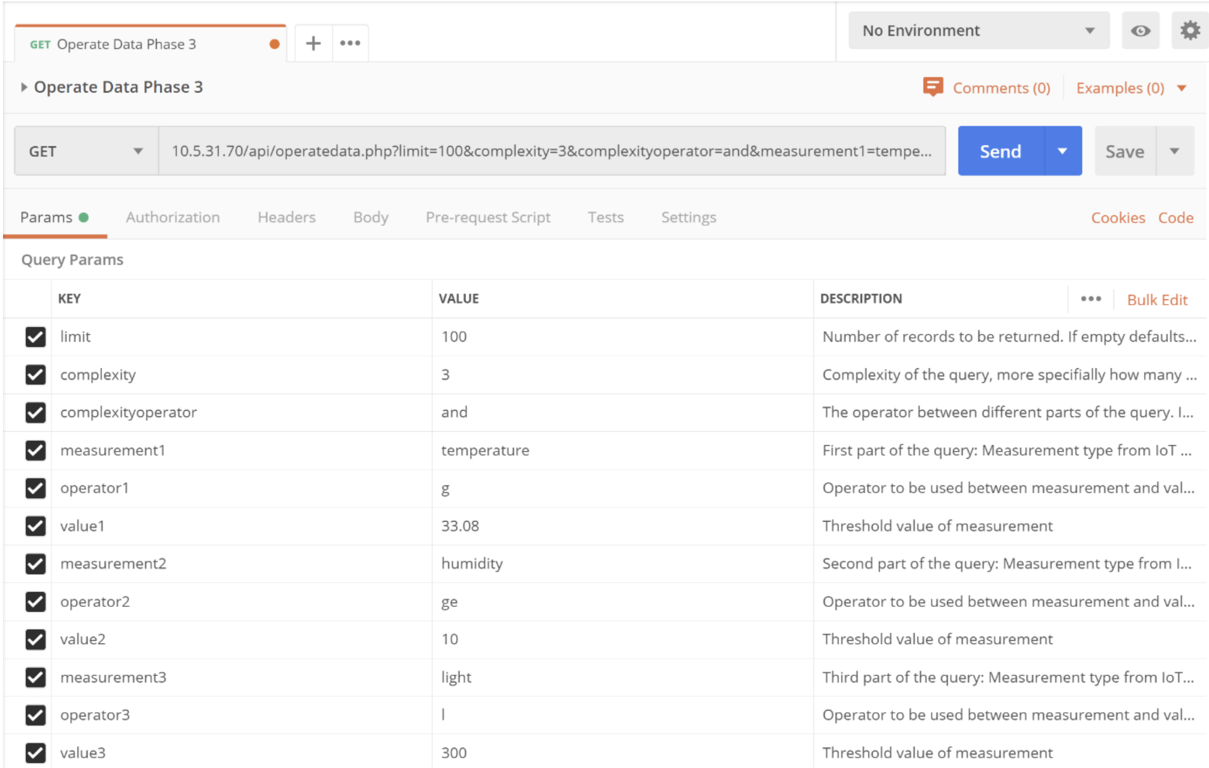


Figure 22: Phase 3 RESTful API extension GET request overview

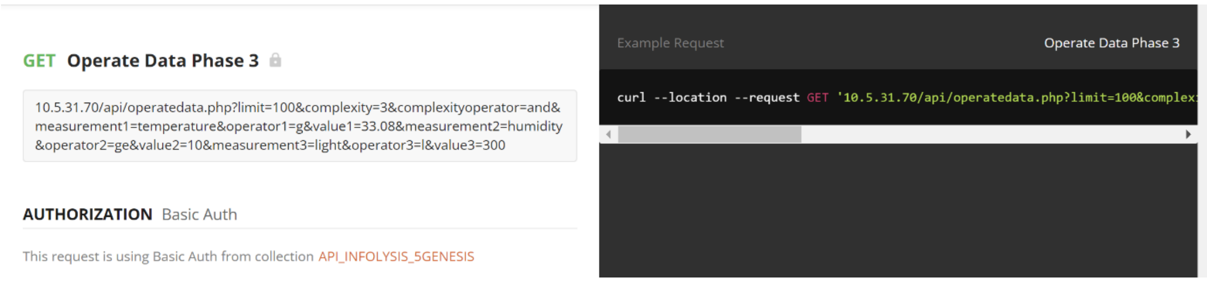


Figure 23: Phase 3 RESTful API extension GET request example

PARAMS

limit	100	Number of records to be returned. If empty defaults to 100
complexity	3	Complexity of the query, more specifially how many different parts the query will have. In this example the query will have 3 different parts.
complexityoperator	and	The operator between different parts of the query. It can be "and" or "or". If the query has only one part it can be ommited.
measurement1	temperature	First part of the query: Measurement type from IoT data. Can be pressure/temperature/light/humidity/accelometer
operator1	g	Operator to be used between measurement and value. Can be g (greater) / e (equals) / l (less) / ge (greater or equal) / le (less or equal)
value1	33.08	Threshold value of measurement

Figure 24: Phase 3 RESTful API extension GET request parameters 1/2

measurement2	humidity Second part of the query: Measurement type from IoT data. Can be pressure/temperature/light/humidity/accelometer
operator2	ge Operator to be used between measurement and value. Can be g (greater) / e (equals) / l (less) / ge (greater or equal) / le (less or equal)
value2	10 Threshold value of measurement
measurement3	light Third part of the query: Measurement type from IoT data. Can be pressure/temperature/light/humidity/accelometer
operator3	l Operator to be used between measurement and value. Can be g (greater) / e (equals) / l (less) / ge (greater or equal) / le (less or equal)
value3	300 Threshold value of measurement

Figure 25: Phase 3 RESTful API extension GET request parameters 2/2

3.2.3 IoT sensors and MONROE nodes

Testing of MONROE nodes with LoRa WAN, WiFi and LTE (Rel. 15 vEPC) have been performed although NB-IoT connectivity testing is on-going (3GPP Rel. compatibility issues currently under investigation). Surrey platform RAN (SW) upgrade to Rel. 15 has been completed.

With regards to the Surrey IoT sensor deployment, a first limited set of experiments (with a small set of IoT sensors) and provision of results to the TTN platform and validation of (sensor) data connectivity via available interfaces, i.e., LoRa, WIFI, and LTE, have been conducted. Additionally, configuration and deployment of Surrey IoT sensors (indoor environment for initial testing purposes) have been conducted.

3.2.4 Overall Phase 3 Achievements

The Phase 3 milestones achieved by the Surrey Platform during Phase 3 are the following:

1. Integration of the Release B of the 5GENESIS Facility components is successfully completed.
2. Final testing of INFOLYSiS vGW under heavy load of data and performing the appropriate optimizations where needed. By testing the system under heavy load, a problem in the

initial implementation of the CoAP mapper was identified and it was fixed by using a different basic library for CoAP packet manipulation, described in detail in section 3.2.2.
3. Extension of the INFOLYSiS vGW API to support more queries and variables in order to disseminate IoT data more efficiently. The INFOLYSiS vGW API was successfully extended to support complex queries using more than one variable, described in detail in section 3.2.2.
4. Deployment and configuration of all planned IoT sensor nodes and emulation of the remaining traffic virtually in order to execute the massive IoT and multimedia communication use case.
5. Testing of a mMTC slice instantiation (in parallel with eMBB slice) is scheduled for October 2021.
6. Indoor/Outdoor FON WiFi AP deployment and integration testing with Surrey 5GC - FON and Surrey 5GC support N3IWF[7] related interfaces. The development of the N3IWF interfaces (on the UE & core-network sides), by both FON and Surrey teams is completed. The deployment of the FON APs in the Surrey premises, and the integration testing of N3IWF interfaces, are also completed.
7. Deployment and testing of the APEX policy management solution and analytics, is re-scheduled for October 2021.
8. Final testing of the Surrey Platform use case is scheduled for October 2021.

[7] "Non-3GPP Inter-Working Function" (N3IWF) is the key element of the untrusted access model in 5G. The UE uses N3IWF to connect to the 5G core over a non-3GPP access layer. N3IWF terminates the security tunnel from the UE side and terminates signaling and data plane from 5G core functional entities. The UE is assumed to support the 5G signaling plane (NAS). N3IWF carries both NAS and user plane data between the UE and 5G core functions.

4 SURREY USE CASES-SPECIFIC EXTENSIONS

4.1 Use Cases Target Deployment

Each 5GENESIS Platform focuses on the validation of a subset of 5G KPIs. Specifically, the Surrey Platform will focus on use case specific requirements for latency, reliability, and coverage.

4.1.1 Massive IoT for Large-Scale Public Events

To be able to implement the planned use-case scenarios a set of technologies and features were added to the Surrey Platform.

- NB-IoT and LoRa [UNIS]: The Surrey Platform has integrated the existing UNIS-wide LoRa deployment. The Surrey Platform infrastructure nodes will incorporate Monitoring Nodes and Probes to support NB-IoT and LoRa;
- WiFi and 5G NR [FON]: The WiFi network is tightly coupled with the 5GC NR deployment and the Surrey test facility (via the N3IWF interface);
- Integration of the WiFi Service Management Platform (WSMP) [FON]: mainly to manage the connection between 3GPP and Wi-Fi (i.e., for LWA based traffic steering), and policy management solution in the core network; FON has created a specific module for its commercial platform WSMP in order to manage the 5G WiFi slices, the project related metrics and the interconnection with 3rd parties modules in order to complete the 5G platform. This module, called WiFi Slice Controller (WSC) is ready to be fully integrated with WSMP once the experiments on the project finish, it will add full 5G capabilities to WSMP;
- Integration of the IoT GW [INF]: The solution supports gateway functionality and monitoring functions with GUI for management. The requirements from the platform side relating to virtualisation environment, IoT devices' protocols, and the preferable protocol for gateway output (called as interoperability protocol), have been addressed;
- Policy Management
  - [KAU/SRL]: Integration of policy management in the Monitoring Nodes and Probes, enabling the incorporation of dynamic control of higher layer protocols for E2E slice management.
  - [LMI]: Integration of policy management in the Monitoring Nodes and Probes, enabling the incorporation of dynamic control of higher layer protocols for E2E slice management.

Finally, NFV and SDN enablers are already in place and operational and provide services such as virtualisation, service function chaining, and network slicing.

4.1.1.1 Use case 1 - Topology and Architecture

The Surrey Platform follows the 5GENESIS architecture proposed in Deliverable D2.2 [6]. The Platform and main components will be hosted in and around the premises of the 5G and 6G

Innovation Centre at the University of Surrey, within this area connections will be via the 5GENESIS branch of the 5G & 6GIC carrier-grade testbed.

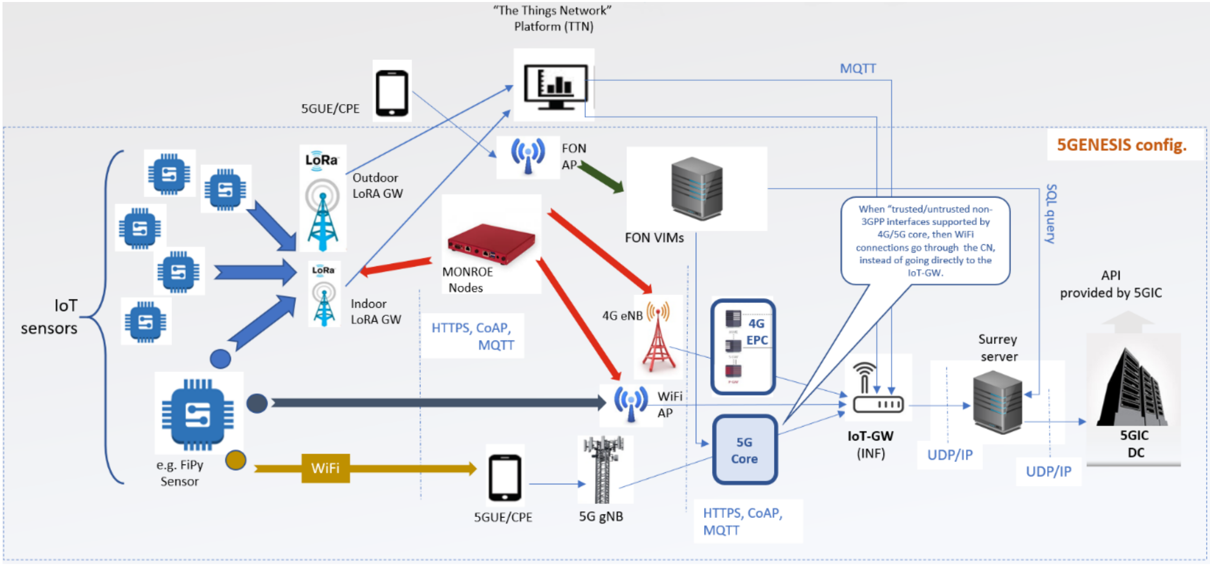


Figure 26: Network topology for the use case scenario at the 5G & 6GIC/ICS Site

Figure 26 shows the workflow of the information within the platform. First, sensors carried by users when visiting a large-scale event on campus will collect sensing data including information about temperature, air quality, presence, movement, acceleration, and other parameters. These data are collected and transmitted using one or more of the available air interfaces and are then passed to the IoT virtual gateway that understands/translates, with the help of the mapping VNFs, the various incoming IoT protocols into UDP-over-IP packets. Then the interoperable IoT data reach the Surrey server with the help of the vGW API.

Phase 2 of the implementation included the connection of two physical monitoring (MONROE) nodes, integration of additional air interfaces (NR, LTE-A, WiFi, Bluetooth, LoRa) and the demonstration of multi-RAT connectivity.

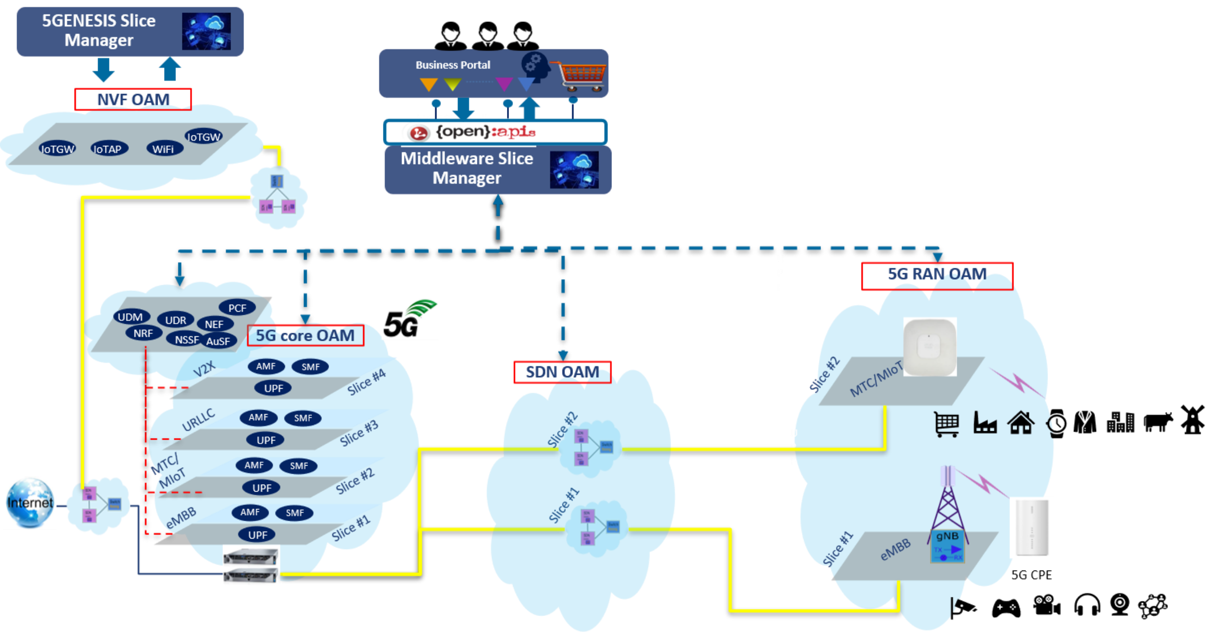


Figure 27: Slicing in the Surrey Platform IoT use case

In Phase 3, changes were made in the INFOLYSiS IoT vGW in order to be compatible with the slicing system for the Massive IoT for Large-Scale Public Events use case. In each slice of the system, an IoT vGW will be automatically instantiated, completely isolated from the others in order to handle all the IoT data of the specific slice. The functionality of each IoT vGW will be exactly the same as it was in Phase 2. The tenants of each slice will have access only to their IoT data.

4.1.2 WiFi Slice Controller

The WiFi Slice Controller (WSC) is the component that receives the slice controller instructions through an API interface to proceed with the creation and management of WiFi slices. It is responsible for creating, deleting, modifying, reactivating and monitoring the status of each created slice.

The WiFi Slice Controller, as exposed in Figure 28, needs to be connected to it through its API in order to execute the requests of the UNIS 5G Core. This API enables 5G Core to automatically manage the RAT slices based on WiFi. According to the standards, WiFi RAT is connected using the N3IWF interface. All these connections are managed by the NW Tunnelling Manager which handles the requests made by the AP Manager component. The AP Manager component not only manages the deployment, provisioning and creation of WiFi networks, but also generates the sent request to create the corresponding tunnels.

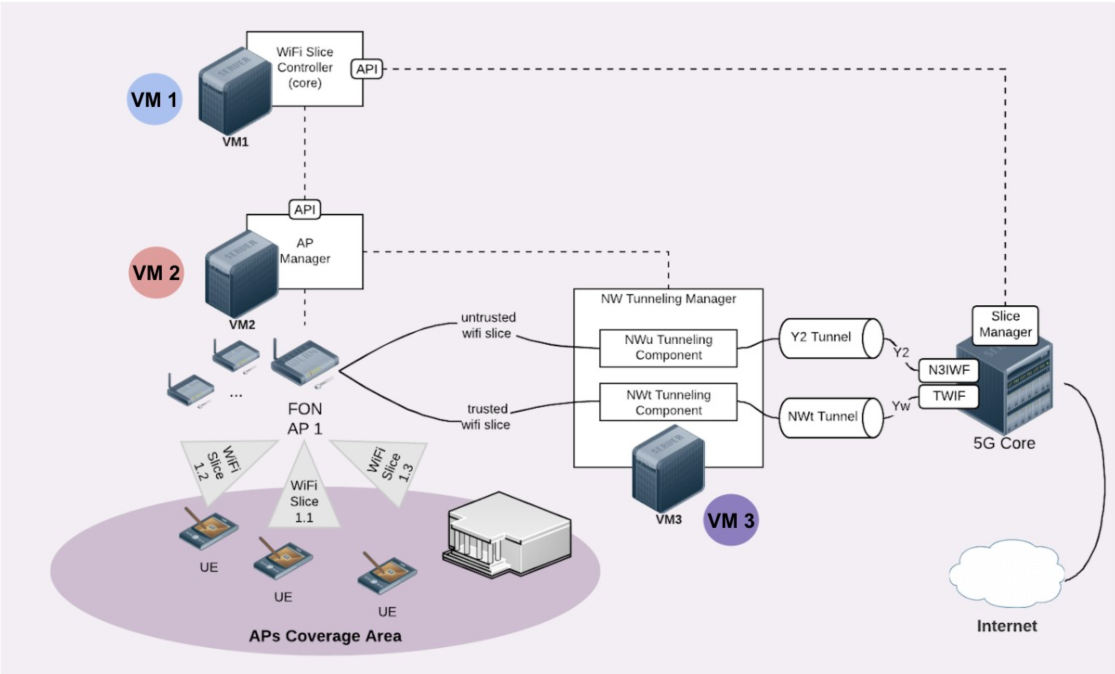


Figure 28: WiFi Slice Architecture: Logic Diagram

As previously introduced, WSC is formed by three main components. The core of WSC is running in one VM, which communicates its decisions to the AP Manager component in order to delegate the operational functions (deploy APs, networks ...). Finally, the tunnelling manager is the third VM which communicates with the 5GC in terms of control and data layers.

The main requirements of the mentioned components are depicted by Figure 29:

WIFI SLICE MANAGER (VM 1)	AP MANAGER (VM 2)	TUNNELING MANAGER (VM 3)
Orchestrator & Manager of required modules to deploy WiFi Slicing: **WS Controller (WSC)** Orchestrator & Manager of the rest of components of the WiFi Slice Platform. **WS Analytics Monitor (WSAM)** Monitors creation of WiFi Slices, APs, WLANs and network status, and stores observed KPIs in InfluxDB	Includes deployment of Ruckus' Virtual SmartZone (vSZ) Controls and configures APs & WLANs (WiFi Slices), as orchestrated by WSC.	Includes deployment of Ruckus' Virtual SmartZone-Dataplane (vSZ-D) Deploys dataplane as needed in NFV architecture, offering secured tunneling of user data traffic.
	Sample Requirements vSZ version: 5.2.0.699 4 Cores, 16 GB RAM, 1TB HD 3 Network Interfaces	**Sample Requirements** vSZ-D version: 5.2.0.699 4 Cores, 16 GB RAM, 1TB HD 2 Network Interfaces

Figure 29: Main WiFi Slice Management Modules

There exist several networks in terms of internal deployment. Figure 30 depicts the needed distribution.

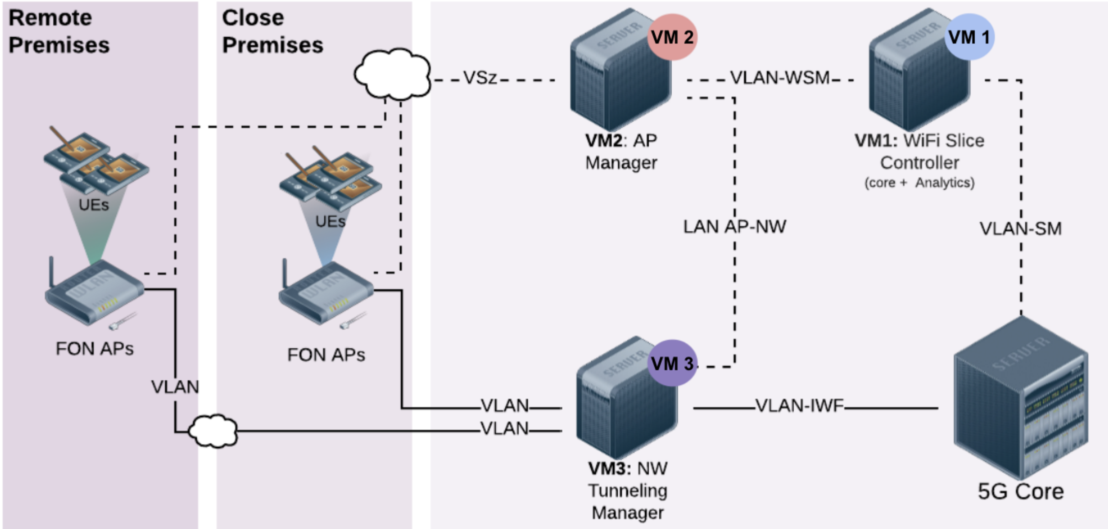


Figure 30: WiFi Slice Architecture: VM Interfaces

In order to perform the tests, 4 APs have been deployed in UNIS premises (Remote premises) as well as in FON ones (close premises), all of them connected to the 5GC using the NW Tunneling Manager. Furthermore, AP Manager and WSC (including the WSAM component) have been deployed inside UNIS premises.

Figure 31: Ruckus AP deployed in the UNIS premises

One of the key components is the WiFi Slice Analytics Monitor (WSAM), see Figure 32. It is the analytics tool which monitors the WiFi Slice Controller (WSC). The WSAM component interacts with both the WSC and the AP Manager. When the WSC receives a request to create or destroy a WiFi Slice, the WSC forwards the request to the AP Manager and then accesses the WSAM API (port 9001) to perform the corresponding notification, including the request type (creation or destroy of a WiFi Slice), the WiFi Slice information (SSID, and internal UUID), and the timestamps of request and task execution. This information is collected in the WSAM component and forwarded to the time series storage service (InfluxDB). It is important to mention that the WSAM component uses SNMP (Simple Network Management Protocol) to communicate with the AP Manager, in both active polling (sending SNMP queries) and passive monitoring (receiving SNMP traps).

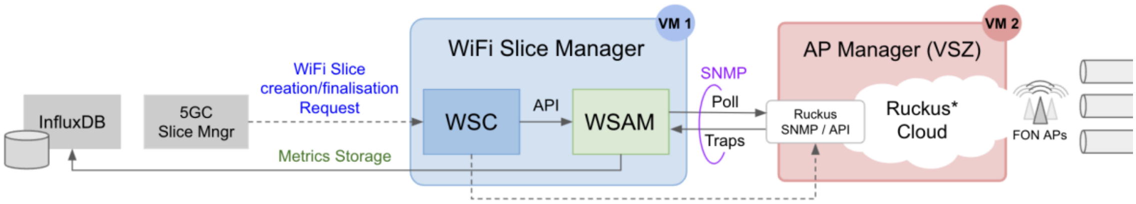


Figure 32: WiFi Slice Architecture: Operation and Monitoring

The main monitoring techniques can be grouped into 2 categories, see Figure 33. On one hand, the active monitoring is a periodical polling of the available metrics obtained from the AP Manager. It periodically performs several SNMP queries to monitor the APs' measurements (i.e., availability, status and performance metrics), number of stations attached, and WLANs' measurements (i.e., performance metrics) and are dynamically stored. On the other hand, the passive monitoring registers changes in the APs' availability, WLAN creation, deployment and finalisation for a defined zone. It receives data not only from the AP Manager, but also from WSC via the WSC-WSAM channel.

Active Monitor (SNMP Query)	Passive Monitor (SNMP Traps / API)	Validation & Storage (InfluxDB)
Periodical polling of available metrics obtained from **AP Manager** (VM2, VSZ) Monitors: · AP availability status & metrics · Number of stations attached · WLAN metrics	Registers changes in APs availability, WLAN creation, deployment and finalisation for a defined zone. Either... - from **AP Manager** (VM2, VSZ) - from **WSC** (VM1) via internal API As a result of this, the list of metrics to poll is updated accordingly.	Every 6 minutes, all collected measurements are sent to InfluxDB. Previously, all snapshots [per AP, per WLAN & per WLAN-&-AP] are validated regarding APs availability and each WLAN's (WiFi Slice) life cycle.

Figure 33: WiFi Slice Analytics Monitor characteristics

5 CONCLUSIONS

This document is the third and final deliverable reporting on the activities regarding the preparation, update and operation of the 5GENESIS Surrey Platform during Phase 3 of the 5GENESIS project. During this Phase, Release B of the Open5GENESIS Suite, i.e., Coordination Layer and Slice Manager, has been integrated within the Surrey Platform, while an IoT sensor deployment has taken place in preparation for the Surrey use case. The Surrey Platform is a multi-RAT environment, comprising both 3GPP (COTS 4G and 5G NR), and non-3GPP networks, as well as a powerful 3GPP Rel.16 SA-compliant 5G core.

The final release of the Surrey Platform will be used for the final round of experiments, which is scheduled for October 2021, and the report on the measured KPIs will be available in deliverable D6.3.

REFERENCES

[1] 5GENESIS Deliverable D4.11, "The Surrey Platform (Release B)", January 2020.
[2] "MONROE Project" [Online], https://www.monroe-project.eu/access-monroe-platform/
[3] 5GENESIS Deliverable D2.4, "Final report on facility design and experimentation planning", July 2020.
[4] 5GENESIS Deliverable D5.2, "System-Level Tests and Verification", September 2021.
[5] 5GENESIS Deliverable D3.6, "Monitoring and Analytics", April 2021.
[6] 5GENESIS Deliverable D2.2, "Initial overall facility design and specifications", December 2018.
[7] 5GENESIS Deliverable D3.5, "Monitoring and analytics", October 2019.

APPENDIX

User Equipment

MONROE Nodes

To support the Surrey use case, a number of physical MONROE nodes were first extended to support Internet of Things (IoT) capabilities and were then deployed in the Surrey testbed. The attached IoT devices to the MONROE nodes are FiPy development board and Pysense sensor shield from Pycom[8]. The FiPy module (shown in Figure 34) supports five different communication interfaces – WiFi, Bluetooth, LoRa, Sigfox and dual LTE-M (Cat-M1 and NB-IoT). It runs a stripped-down implementation of Python3 (called micropython) that is optimized to run on IoT devices. FiPy was chosen because of the ease of writing, uploading and managing code on the device. The FiPy module is connected to the Pysense sensor shield that hosts an array of sensors for temperature, humidity, ambient light, barometric pressure and acceleration sensing.

Figure 34: Product image of the Pycom FiPy

The Pysense board, interfaced with FiPy, is connected to the MONROE node using USB through a switchable USB hub. Sometimes the USB connected devices require power cycling which can be difficult to do manually in case the MONROE nodes are placed remotely (i.e., where there is no physical reach). In order to enable remote power cycling, Yukush hub from Yepkit[9] was used to enable software-based ON/OFF powering of the connected USB devices. Figure 35 shows how a FiPy module is connected to a MONROE node. The MONROE nodes contain the necessary functionalities so that the IoT devices are discoverable and can be used to run measurements.

[8] Pycom.io
[9] Yepkit.com

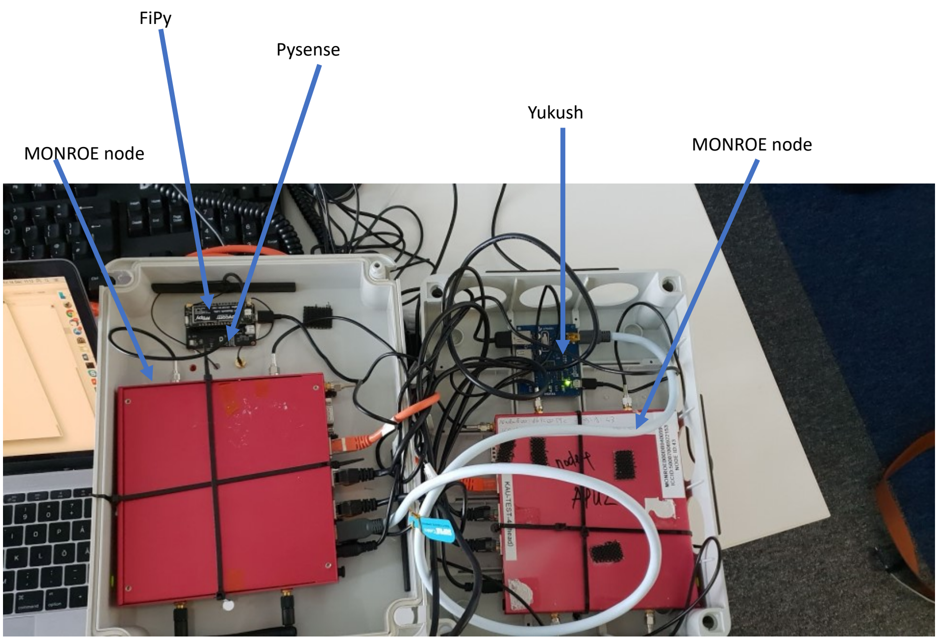


Figure 35: MONROE node connected to FiPy module.

Two MONROE probes (each with two nodes - head and tail) were deployed in the Surrey platform. The tail node in each probe is connected to a FiPy module. The probes offer connectivity through both NB-IoT and LoRaWAN. Since backward compatibility from NB-IoT Rel.15 (on the network side) with Rel.13 (probes) is not yet formally established, only LoRaWAN has been connected so far.

The LoRaWAN specification requires at least three components to form a LoRaWAN[10] network: (i) A LoRa capable end device, in our case the FiPy module, (ii) a gateway, acting as a base station that communicates with LoRa devices using the LoRa RF protocol and forwards data from the devices over IP backhaul to any LoRa server, and (iii) the LoRa server that receives, decodes and stores data sent by IoT devices. The Things Network (TTN)[11] is a globally distributed, crowdsourced LoRa server that handles communication with IoT devices through LoRa gateways. There are several TTN gateways placed in Surrey that the MONROE nodes can connect to. Some initial tests were performed, which showed that MONROE devices in Surrey could successfully send sensor values to the TTN server through these gateways.

Besides the added IoT extensions, the MONROE probes also offer connectivity over 4G (Rel. 15 vEPC) and WiFi. WiFi connection was tested against both ethernet connected WiFI and 5G CPE. The MONROE probes at Surrey support running containerized experiments as a general performance monitoring probe within the 5GENESIS Monitoring and Analytics Framework (see Deliverable D3.5 [7]).

[10] Lora-alliance.org
[11] Thethingsnetwork.org