5Genesis

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

[H2020 - Grant Agreement No. 815178]

Deliverable D4.11

# The Surrey Platform (Release B)


**Editor** D. Triantafyllopoulou (UNIS)

**Contributors** D. Triantafyllopoulou, S. Vahid, K. Moessner, R. Yogaratnam, F. Carrez (UNIS), A. Brunstrom, M. Rajiullah (KAU), I. Koffmann (REL), P. Matzakos, F. Kaltenberger (ECM), V. Koumaras, C. Sakkas, G. Theodoropoulos (INF), I. Pretel, I. Etxebarria (FON), A.M. Bosneag (LMI/ERICSSON)



**Version** 1.0

**Date** January 31st, 2020

**Distribution** PUBLIC (PU)

# List of Authors


| | |
|---|---|
| UNIS | UNIVERSITY OF SURREY |
| D. Triantafyllopoulou, S. Vahid, Y. Rahulan, F. Carrez | |
| INF | INFOLYSiS |
| C. Sakkas, V. Koumaras, G. Theodoropoulos | |
| KAU | Karlstad University |
| A. Brunstrom, S. Alfredsson, M. Rajiullah | |
| FON | FON TECHNOLOGY SL |
| I. Pretel, I. Etxebarria | |
| LMI | L.M. ERICSSON LIMITED |
| Anne-Marie Cristina Bosneag | |
| REL | RUNEL NGMT LTD |
| I. Koffmann | |
| ECM | EURECOM |
| P. Matzakos | |

# 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 D4.11 | As shown on first page | 31/1/2020 |

# LIST OF ACRONYMS

| Acronym | Meaning |
|---|---|
| 5GC | 5G Core |
| 5GIC | 5G Innovation Centre |
| BH | Backhaul |
| AAA | Authentication, Authorization, and Accounting |
| AMF | Access and Mobility Management Function |
| AP | Access Point |
| APEX | Adaptive Policy Execution |
| API | Application Programming Interface |
| CoAP | Constrained Application Protocol |
| COTS | Commercial of the Shelf |
| CP | Control Plane |
| CPU | Central Processing Unit |
| DC | Data Centre |
| DWDM | Dense Wavelength Division Multiplexing |
| E2E | End-to-End |
| ELCM | Experiment LifeCycle Manager |
| eMBB | Enhanced Mobile Broadband |
| EMS | Element Management System |
| EPC | Evolved Packet Core |
| E-UTRAN | Evolved-UMTS Terrestrial Radio Access Network |
| GUI | Graphical User Interface |
| GW | Gateway |
| HTTP | HyperText Transfer Protocol |
| HW | Hardware |
| IETF | Internet Engineering Task Force |
| IoT | Internet of Things |
| IoT-vGW | IoT virtual Gateway |
| IP | Internet Protocol |
| JANET | Joint Academic NETwork |

| | |
|---|---|
| JSON | JavaScript Object Notation |
| KPI | Key Performance Indicator |
| LTE | Long-Term Evolution |
| MAC | Medium Access Control |
| MANO | Management and Orchestration |
| MEC | Mobile Edge Computing |
| mMTC | massive Machine Type Communication |
| MQTT | Message Queuing Telemetry Transport |
| N3IWF | Non-3GPP Inter-Working Function |
| NB-IoT | Narrowband IoT |
| NF | Network Function |
| NFV | Network Function Virtualisation |
| NFVI | NFV Infrastructure |
| NFVO | NFV Orchestrator |
| NMS | Network Management System |
| NR | New Radio |
| NSA | Non-Standalone |
| OAI | OpenAirInterface |
| 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 Network |
| SIM | Subscriber Identification Module |
| SMF | Session Management Function |
| SW | Software |
| TAP | Test Automation Platform |
| TCP | Transport Control Protocol |
| TTN | The Things Network |
| TTP | "The Things" platform |

| | |
|---|---|
| UDP | User Datagram Protocol |
| UE | User Equipment |
| UEPS | Universal Execution Policy Specification |
| UP | User Plane |
| UPF | User Plane Function |
| URLLC | Ultra-Reliable Low-Latency Communication |
| vGW | Virtual Gateway |
| VIM | Virtualization Infrastructure Manager |
| VLAN | Virtual Local Area Network |
| VM | Virtual Machine |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| WIM | WAN Infrastructure Manager |
| WLAN | Wireless Local Area Network |
| WP | Work Package |
| WSMP | WiFi Service Management Platform |

# 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 Innovation Centre (5GIC) 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 second 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 currently Rel.15 compliant, with the plan to also comply with Rel.16 by the end of the first quarter of 2020. On the user side, the Platform will use the MONROE probes, which are extended with IoT capabilities, and tailored to the needs of the Surrey use case. Moreover, commercial 5G handsets will also be used. On the IoT side, the Surrey Platform uses micro-controller devices that will 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 second report from a series of three, 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 will follow, providing a demonstration of vertical use cases allowing specific KPI evaluation based on relevance to the use case. For the Surrey platform, a specific 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 NFV/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 (EPC and EUTRAN), as well as the mmWave backhaul network and the open-source MANO (OSM)-based Infrastructure.

Following the approach adopted by the 5GENESIS project, the Surrey platform components and technologies (adopted in phase 2) 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 Open Source MANO (OSM) ) components were already implemented and integrated during phase 1, while infrastructure layer comprises multiple components and covering different RATs, together with the 5G Core (5GC) that has been developed in-house, in Surrey, continue to be enhanced and upgraded.

The aim of Phase 2 has been 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. More specifically, at the end of Phase 2, the following milestones have been achieved:

- Integration of INFOLYSiS IoT-vGW and VNF components.
- Configuration and testing of IoT sensors and MONROE nodes.
- Establishment of Virtual Private Network (VPN) for secure access to the Surrey Platform (a.k.a the "5GENESIS Island") and for inter-platform connectivity.
- Test and verification of core/edge VNFs developed in-house.
- 60 GHz mmWave Backhaul (BH) testing and set of indoor and outdoor measurements.
- Testing of context-aware networking[1] capability in the 5G Core.
- IoT-vGW interoperability testing with HTTP, COAP and MQTT protocols.
- Development of N3IWF interfaces (NWu & Y2) by FON & Surrey platform team (In progress).
- Deployment and integration of Release A of Coordination layer components, i.e.
    - The 5GENESIS Portal.
    - The 5GENESIS Experiment Lifecycle Manager (ELCM).
    - The 5GENESIS Slice Manager.
    - TAP tool.
    - Prometheus.
    - Dispatcher and the Experimenter Portal
    - The storage (InfluxDB) and visualization (Grafana) components of the monitoring subsystem.

During the last three months of this phase, extensive tests will be performed to ensure that all the newly integrated services and components work as expected and, in parallel, the platform partners will continue their efforts toward integrating elements and technologies.

The Surrey platform will be used for the validation and testing of second round of experiments until March 2020, and the report on the KPIs measured will be available in deliverable D6.2. This report will be followed up by a final deliverable in these series, based on the Release C of the 5GENESIS SW components.

[1] This feature has been developed and will be deployed on the Surrey Platform. E2E messages have been designed and developed for communication between the prototype 5G UE and the new 5G core network components. This feature will enable context-aware messaging between the UE and the core, which in turn provides input for dynamic network slicing decisions (user application context, and mobility context) and for user plane anchoring decisions (mobility context).

# Table of Contents

# List of Figures

# List of Tables

# 1. INTRODUCTION

## 1.1. Purpose of the document

This is the second deliverable documenting the activities regarding the preparation, upgrading and operating of the Surrey Platform. This work is undertaken in 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 Key Performance Indicators (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 experimentations, an overview of the Platform, covering the infrastructure, Management and Orchestration, as well as the 5GENESIS Coordination Layer is provided (section 2.2). The mobile network technologies employed in all parts of the network (from User Equipment, to Radio Access Network and Mobile Core), resulting in a multi-Radio Access Technology (RAT) environment supporting massive Internet of Things (IoT) and multimedia communications are described in detail. 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. As this Platform focuses on the support of 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 virtual Gateway (IoT-vGW) in order to make them unified and interoperable for further use.

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 Phases 1 and 2 of the 5GENESIS project. Moreover, a detailed plan for Phase 3 is provided.

Currently, the 5GENESIS project has released three 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. |

The Surrey Platform focuses on validating KPIs related to latency, reliability, user density and service creation time. 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 Phases 1 and 2. A detailed plan for Phase 3, is also provided.

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 2 of the 5GENESIS project. Its target audience includes the 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 Site Overview

The 5G Innovation Centre (5GIC) 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 5GIC 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 will be integrated into the Surrey site.

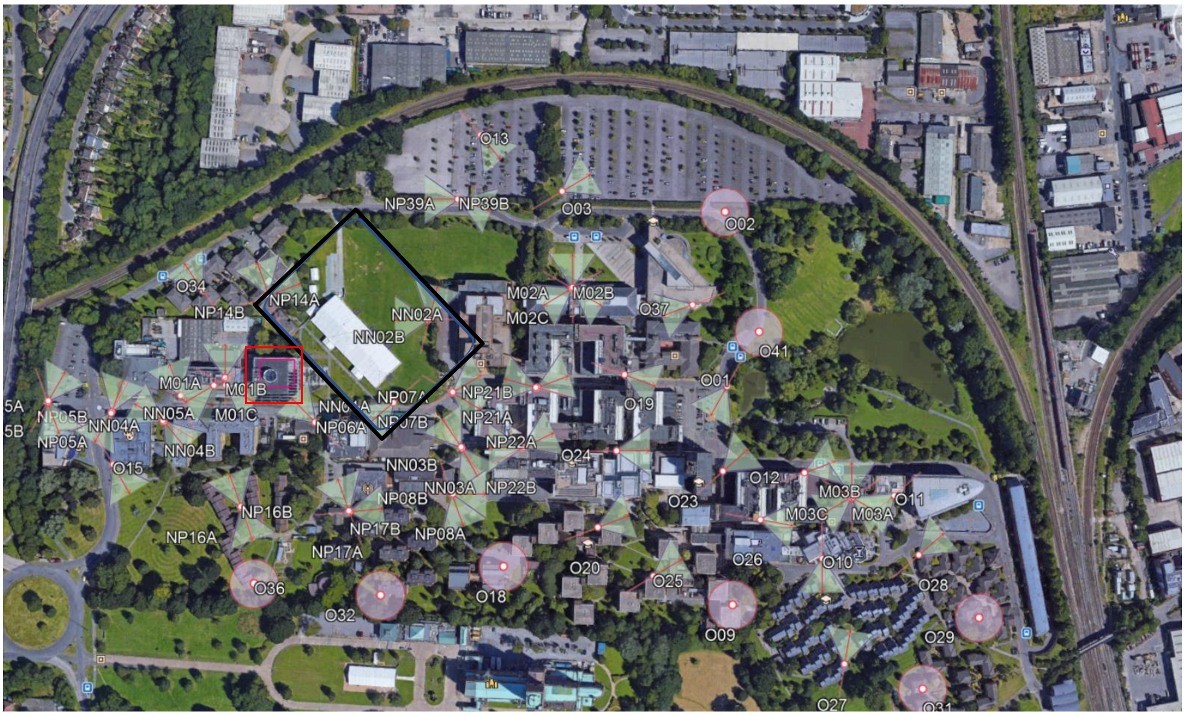


Figure 1: Network RAN deployment at the Surrey site

## 2.2. Target Platform Topology

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

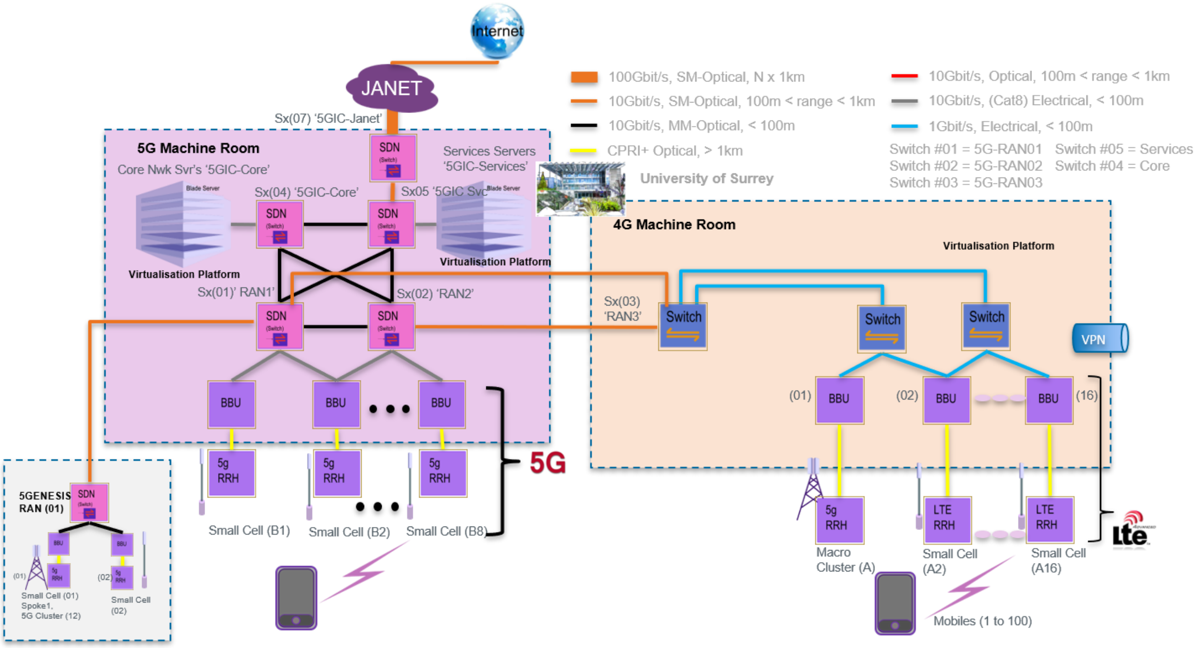



Figure 2: Surrey platform topology/deployment

The Surrey platform is connected to the JANET network[2] with 10 x 10Gbps 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.

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

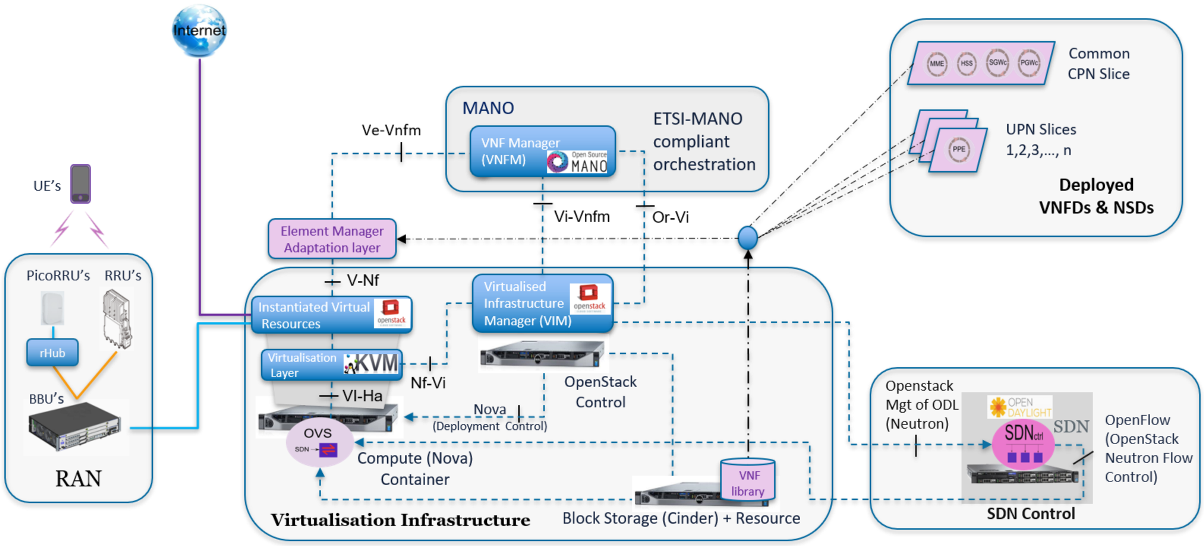



Figure 3: The Surrey platform "5GENESIS Island" – software architecture and components

The 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 described in [1]. The Platform and main components are hosted in and around the premises of the 5G Innovation Centre at the University of Surrey, and within this area, connectivity will be via the 5GENESIS branch of the 5GIC carrier-grade testbed. The Surrey platform currently supports Rel. 15 4G/5G. Figure 4 shows the workflow of the information within the platform. First, sensors deployed or 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. This data is then collected and transmitted using one or more of the available air interfaces on campus (use-case designated arena) and is then passed to the IoT virtual gateway (IoT-vGW) that understands/translates the various incoming IoT protocols into UDP-over-IP packets, and 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 (section 2.2.1.1) and ii) Deployment and operation of INFOLYSiS IoT-vGW (section 2.2.1.2), followed by the setup of a dedicated mMTC slice on the 5GC.

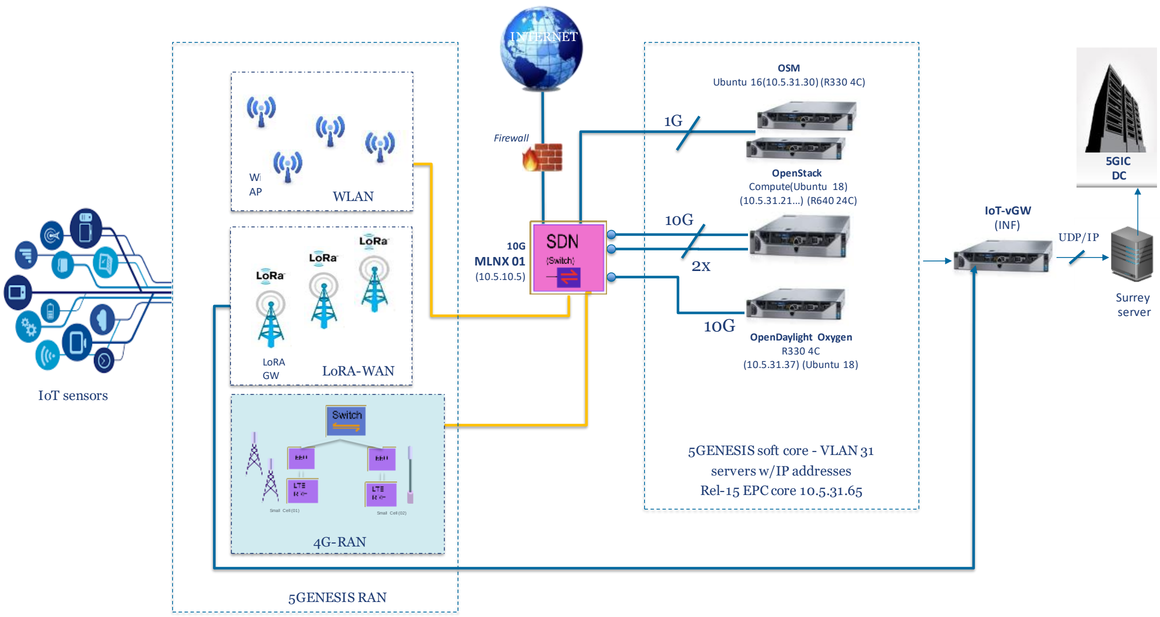



Figure 4: The Surrey Platform IoT network

### 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 following two tables describe i) what sensor data is transmitted by the sensors (Table 2), and ii) the devices/nodes used (Table 3). Both contributions share the same data output (json) structure but depending on device and configuration, some data is ignored.

Table 2: Sensor data and meta-data coverage (per device)

| | Pycom Pysense | Android App |
|---|---|---|
| Pressure | X | X |
| Temperature | X | X |
| Light | X | X |
| Humidity | X | X |
| Proximity | - | X |
| Gravity | - | X |
| Accelerometer | X (roll/pitch) | X |
| Compass | - | X |

| | | |
|---|---|---|
| Timestamp | X | X |
| Latitude | - | X[3] |
| Longitude | - | X[4] |

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

| | Pycom Pysense | SmartPhones (w/ Android app) | IoT (LoRa) GWs |
|---|---|---|---|
| Planned (as in DoW) | 30~50 sensors + load emulation | none[5] | 3 |
| Actual | 35 + load emulation | >= 5 | 5 |

### (ii) Pycom FiPy+Pysense board

The sensors used are manufactured by Pycom[6]. A sensor consists of an ultra-low power programming board <<**FiPy**>> (see Figure 5 below) equipped with an expansion shield <<**PySense**>> that provides a variety of classical sensors (see Figure 6 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)


Figure 5: <<FiPy>> programming board with radio

[3] If the end user decides to share that information
[4] If the end user decides to share that information
[5] Not part of the DoW, no contractual binding.
[6] https://pycom.io/product/fipy/

Figure 6: <<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 setup. The data format used to convey the sensor data to the Surrey 5GENESIS platform follows the 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 seems to be NB-IoT release 13 while the Surrey Platform supports release 15 currently. Backward compatibility will have to be checked as soon as Subscriber Identification Module (SIM) cards are made available for testing;
- 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 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.

### (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).

  As depicted in Figure 7, a preliminary mockup was implemented for the purpose of testing using the IBM *NodeRed* tool. In a nutshell an initial LoRa node [physical GW] (node1 - provided by TTN) receives data from the TTN network, passes the data to a formatting node (node 2) and then finally to an InfluxDB node (node3) for local storage (MacBook Pro). We then used Grafana and an InfluxDB data-source to query the database and achieve some visualisation. At the time of the writing, we are connecting Grafana directly to the data received by the Surrey Platform.

  The collected data records are then pushed to the local Surrey servers using different protocols and using different radio channels. In the case of data-over-LoRa GWs, for example, the data is collected and sent to "The Things Network (TTN)" servers first, located in the Netherlands using LoRa gateways. The Surrey Platform server then connects to TTN using MQTT in order to receive and post-process the data locally (including storage).

  As far as LoRa gateways are concerned, The 5GIC building is equipped with a dozen of small LoRa gateway units (Figure 10), while much larger and roof-top gateways (see Figure 9 below) are also deployed over the campus and university accommodation areas.

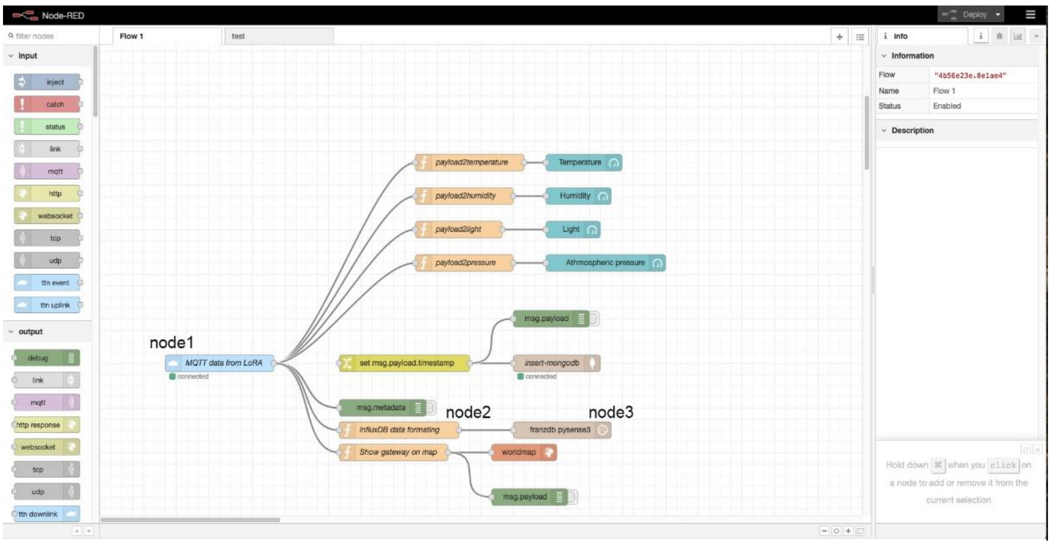



Figure 7: NodeRed nodes involved in the LoRa tests


Figure 8: Visualisation of LoRa data using Grafana and InfluxDB

- HTTP Push over WiFi / LTE: 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.
- CoAP over WiFi / LTE: 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 / LTE: Data is published to the Surrey Platform MQTT queue along with the entityID topic where entityID= ”Pysense”. Only the WiFi part has been implemented at the time of the document writing;


Figure 9: External fixed LoRa gateway


Figure 10: Internal portable LoRa gateway

#### (iv) Android Application

In order to secure the use of LTE for transferring data (as backward compatibility from NB-IoT R15 to R13 is not yet formally established), it was decided to implement a small Android crowdsourcing application, which will transfer data coming from the smartphone embedded sensors, alongside the Global Positioning System (GPS) location to the Surrey Platform, using the various LTE enabled base stations distributed across the Surrey campus.

Like for the Pycom devices, various protocols over LTE (CoAP, MQTT, HTTP POST) will be used depending on how the application is configured by the end-user. It is worth noting that the development and use of this application is not contractually mandated and so it is to be considered as an "additional" extension aiming at increasing even more the IoT data volume. At the time of the writing, the implementation is still work in progress.

### 2.2.2. Platform Infrastructure Layer

#### 2.2.2.1. Mobile Network Technology

The mobile network technologies deployed in the Surrey Platform, along with the respective roadmap, are summarised in Table 4.

Table 4: 5GENESIS Surrey Platform Technology and Roadmap

| | Mobile Core Product | Radio Access Products | UE | 3GPP Rel. (CN compliance) |
|---|---|---|---|---|
| Phase 1 | 5GIC 4G vEPC<br>5GIC 5GC NSA | HUAWEI/AIRSPAN | 4G Handsets<br>MONROE nodes | Rel. 15 |
| | | LoRa WAN | | |
| Phase 2 | 5GIC 5GC SA | HUAWEI gNodeB (Rel. 15) | 5G Handsets &<br>MONROE nodes | Rel. 16 |
| | | COTS NB IoT (Rel.15) | | |
| | | LoRa WAN | | |
| Phase 3 | 5GIC 5GC SA | HUAWEI gNodeB (Re. 16) | 5G Handsets | Rel. 16 |
| | | COTS NB IoT (Rel.15) | | |
| | | WiFi Integration (N3IWF) | | |
| | | LoRa WAN | | |
| | | REL gNodeB | ECM 5G-NR UE | |

## (a) Radio Access

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

Specifically, Commercial of 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 access points (APs) interconnected to the Surrey Platform 5G Core (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 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 5.

Table 5: 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 | 8 | 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 |

3.5GHz eMBB

3.5GHz eMBB & 3.5GHz URLLC    3.5GHz eMBB & LTE    700MHz Omni

Figure 11: 5G NR Coverage at the Surrey Platform

More detailed technical descriptions of the radio access solutions, to be deployed during Phase 3, in the Surrey Platform can be found in Annex 1: 5GENESIS Surrey Platform Radio Access Technologies.

### (b) Mobile Core

The Surrey Platform 5GCore (5GC), developed in-house, fully supports the 3GPP Release 15 for the core network functionality and Rel-16 context-aware network, to intelligently interwork with 5G New Radio, both Stand-alone (SA) and Non-Stand-alone (NSA).

The 5GC network implements the new 5G components as standalone. Through this, it enables fast, reliable, high throughput and multi-access network capability, i.e., non-3GPP access network and Satellite network.

The 5GC includes a large level of newly implemented functions developed on top of an accelerated software platform:

- Integration with 5G New Radio: NSA [S1-MME, S1-U] and SA [N1, N2, N3],
- Implementation of control-user plane split – PFCP [N4],
- Implementation of Service-Based Architecture Features [HTTP/2, REST].

The 5GC integrates with 5G New Radio SA and NSA prototypes and off-the-shelf Long-Term Evolution (LTE) access networks enabling demonstration of different features and applications and supporting the current need to have a genuine 5G Core Network in addition to the evolved EPC one.

The 5GC, as a Network Function (NF), runs on top of common hardware platforms. NFs are deployed with K8s Docker containers platform and virtual machines on top of a large number of virtualization environments. It also supports microservices-based architectures.

#### (c) User Equipment

Two physical dual-node MONROE probes have been deployed as part of the Surrey Platform. In contrast to the original MONROE hardware probes [1], and in support of the Surrey use case, the physical probes in Surrey have been extended with Internet of Things (IoT) capabilities. In line with the Surrey Platform IoT network (see Section 0) the FiPy development board and sensor shield (Pysense) manufactured by Pycom was selected as the IoT device to add to the MONROE probes. The FiPy module supports five different communication interfaces – WiFi, Bluetooth, LoRa, Sigfox and dual LTE-M (Cat-M1 and NB-IoT), with LoRa and NB-IoT the most relevant to the Surrey use case. The Pysense sensor shield is connected to the FiPy module and an array of sensors for temperature, humidity, ambient light, barometric pressure and acceleration sensing are mounted in the Pysense.

The FiPy module is connected to one of the MONROE nodes within the dual-node MONROE probe using USB through a switchable USB hub. Besides the added IoT extensions, the MONROE probes also offer connectivity over 4G and WiFi and can thus be used for multi-RAT experiments over 4G, WiFi, NB-IoT and LoRaWAN. While connectivity over 4G, WiFi, and LoRaWAN have been established, integration with the NB-IoT deployment in Surrey is ongoing, since backward compatibility from NB-IoT Rel.15 (on the network side) with Rel.13 (probes) is not yet formally established. The MONROE hardware probes support running containerized experiments (MONROE measurement probes) in the same way as the MONROE virtual node (VN), which is available as a general performance monitoring probe within the 5GENESIS Monitoring and Analytics Framework (see Deliverable D3.5). Further details on the MONROE IoT extension for Surrey was provided in Deliverable D4.10 [2].

### 2.2.2.2. 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 VPN access.


Figure 12: The 5GIC main data centre

#### 2.2.2.3. 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.

#### 2.2.2.4. 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 Joint Academic NETwork (JANET) backbone network combines high bandwidth and low latency features providing a wide range of services. A transport network capacity of 1 Gbps throughput is available for traffic to and from the Surrey Platform via JANET backbone. Furthermore, OpenVPN based connectivity will be established to secure the links between the Surrey and Berlin sites. This will provide a reliable backhaul link during the experimentation trials (inter-platform connectivity) between Berlin and the Surrey Platforms to validate the Key Performance Indicators (KPIs). Additionally, the JANET/OpenVPN network will provide secure access to services deployed in partner's platform, which is planned for Phase 3.

##### (b) DWDM access to nation-wide fibre network

In addition to the terrestrial backhaul network link provided by JANET, 5GIC 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. The integration of this connectivity with Berlin platform is envisioned as a terrestrial backhaul in the final Phase of the project.

#### (c) mmWave-based backhaul network

The Bluwireless[7] 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 Gbps and ranges over 300m.

### 2.2.3. Management & Orchestration Layer

An overview of the Management and Orchestration layer components and associated technologies deployed in the Surrey Platform is provided in Table 6.

Table 6: Management and Orchestration Layer components – Phase 2

| 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, MEC) and Network (SDN).<br>*NB. API access can be provided from platform SDN controller to the slice manager.* |
| NFV orchestrator | The platform supports OSM v5. |
| NMS | NMS is a composition of different tools/managers:<br>– WAN Infrastructure Manager (WIM)[8]<br>– EMS for the 4G/5G Radio and Core components |
| NFV Infrastructure Management | NFVI/VIM based on OpenStack (at the edge and the core)<br>*NB. NFVI/VIM based on K8s (at the edge): Being evaluated for the next release (if current MANO solution is to be migrated to ONAP from OSM).* |
| WAN/VIM | SDN transport<br>*NB. SDN based WAN controlled by ODL (planned for early 2020)* |

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 Open Stack, the standard de-facto VIM in the ETSI NFV specification. Additionally, it is worth mentioning that the communication between NFVO (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.

[7] https://www.bluwireless.co.uk/
[8] In the reference architecture the VIM in located within the NFV Orchestrator. And the WIM is the one located in the NMS.

- 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 will provide an overview of the physical resources and an interface to manage them. The management of the resources will be provided by the Resource Manager through the network and the inventory repository. The Element Management (EMS), included in the NMS and responsible for the management of a PNF/VNF, is provided by a component in OSM.
- Slice Manager
  The Slice Manager is a common component that is instantiated in all 5GENESIS platforms and its first release was developed within WP3, specifically in Task 3.2.

#### 2.2.3.1. APEX

APEX (Adaptive Policy EXecution)[9] is a lightweight policy engine adaptable on the fly during the operation of the system, based on context information or updated policies. APEX has been developed in LMI and is currently available as open-source. APEX has been installed in the Surrey Platform and it will be integrated for the purpose of enabling closed-loop automation and optimization. APEX is written in Java and runs on any platform that supports a JVM, e.g., Windows, Unix, Cygwin.

An APEX engine has two main interfaces:

- An input interface to receive events: also known as ingress interface or consumer, receiving (consuming) events commonly named **triggers**, and
- An output interface to publish produced events: also known as egress interface or producer, sending (publishing) events commonly named **actions** or action events.

APEX policies are defined in a Universal Execution Policy Specification (UEPS), directly executable in an APEX engine. Higher-level policy specifications (or existing policy languages) can be easily translated in UEPS. An APEX system can use multiple policy engines with different policies deployed on each of them. Context information is automatically shared between all engine instances.

APEX is best located close to the component whose behaviour optimises. Deliverable D3.1 [3] presents a proposal on how to use APEX in conjunction with a Slice Manager for optimisations. In this case, triggers are received from the Analytics module in the Coordination layer, and recommendations are sent to the Slice Manager.

#### 2.2.3.2. NEAT

As introduced in Deliverable D4.10 [2], the Surrey Platform will integrate the NEAT policy framework [4], enabling policy management at the UE. NEAT runs on end hosts and decouples the application from the transport protocol used, allowing the protocol and its configuration to be dynamically selected at run-time based on application preferences, network status, and system policy. This allows performance to be dynamically optimized. NEAT is an

[9] https://ericsson.github.io/apex-docs/

implementation of the transport services concept currently under standardization in the Internet Engineering Task Force (IETF)[10].

Within the Surrey Platform, the NEAT software has been integrated within the physical MONROE nodes as part of the second integration Phase. As such, this allows experiments with dynamic policy-based protocol selection and configuration. Once integrated with the 5GENESIS Slice Manager, this will enable the configuration and optimization of higher layer protocols as part of End-to-End (E2E) slice/policy management. The policy-based protocol selection may involve the following three aspects: (i) transport protocol selection and configuration, (ii) selection between IPv6 and IPv4, (iii) selection of which interface to be used. In the context of the Surrey Platform, with its focus on multi-RAT environments, policy-based interface selection is of particular interest.

To take full benefit from the NEAT system, networked applications must be designed to make explicit use of NEAT. However, it is not always feasible to rebuild non-NEAT applications to use the NEAT User API. To allow such applications to harness NEAT functionality, a proxy solution has been created [5] The NEAT proxy transparently intercepts Transport Control Protocol (TCP) connections and establishes a new connection via the NEAT system between the proxy and the original destination. To allow tests with general applications, the NEAT proxy has been integrated in the MONROE measurement nodes, where experimenters may optionally enable the NEAT proxy to provide MONROE experiments with transparent access to transport services.

### 2.2.4. 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 D4.4 [7] and therefore not included in this document.

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

[10] https://datatracker.ietf.org/wg/taps/

# 3. EVOLUTION OF THE SURREY PLATFORM

## 3.1. Phase 2 Instantiation of the 5GENESIS Architecture

This section reports on the second 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 28 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.

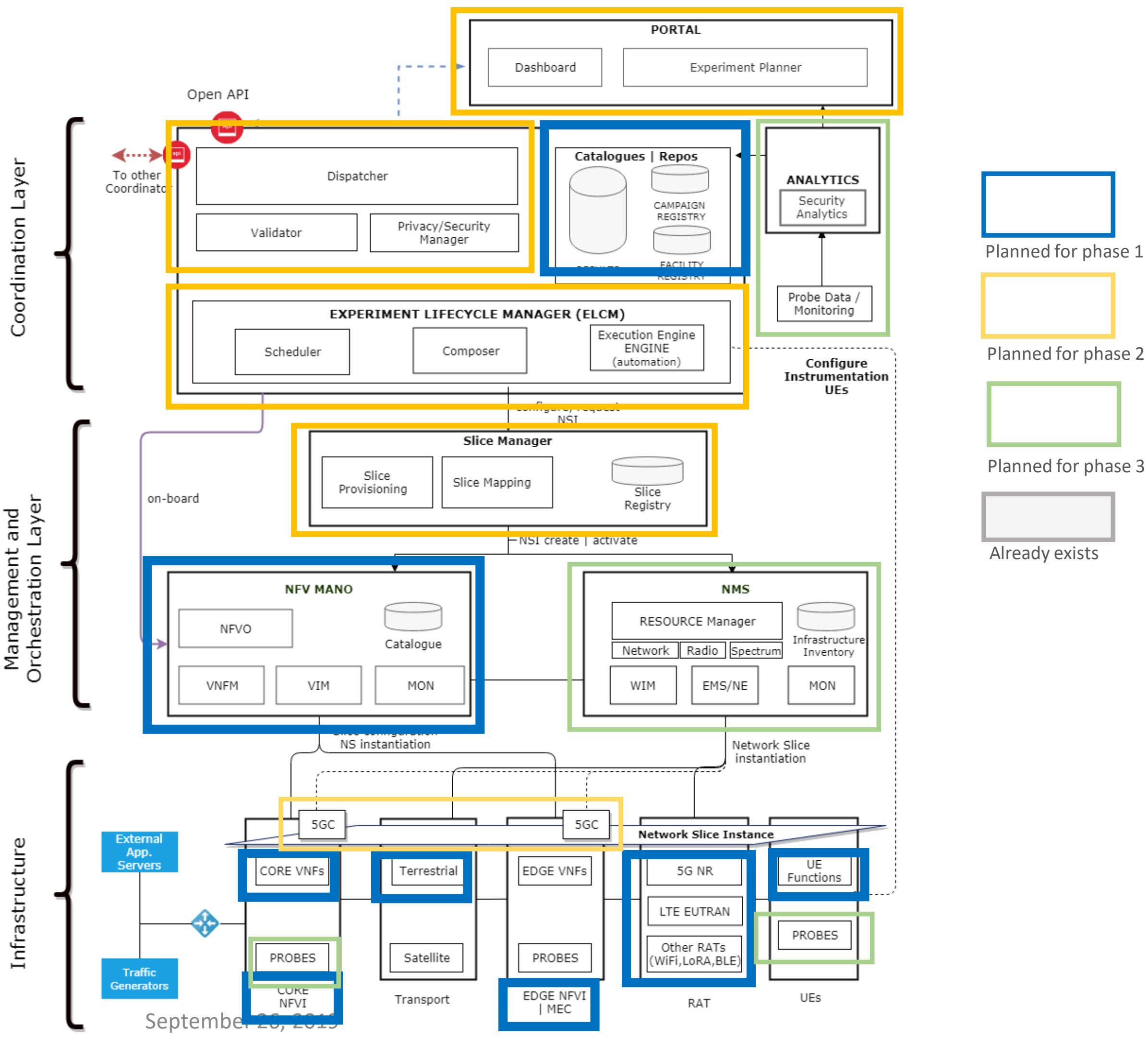



Figure 13: Per-phase instantiation of the 5GENESIS Architecture for the Surrey platform

The aim of Phase 2 is 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 include: 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 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.

During Phase 3, the final releases of the Coordination layer components and the Portal will be integrated and, together with full integration of the Slice Manager with MANO and 5GC components, it will be possible 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 and 2 are summarized in Table 7 below.

Table 7: Phased deployments and configurations in the Surrey platform

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

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

## 3.2. Phase 1 Accomplishments

The Surrey Platform has completed the following achievements during Phase 1:

1. Alignment with the proposed 5GENESIS architecture defined in [8],
2. Purchase and deployment of new servers, switches and communication equipment to support the main data centre (DC),
3. Integration of the Surrey Platform "5GENESIS Island" core network with the indoor HUAWEI FEMOT/small cells and outdoor AirSpan eNBs,
4. In-house development of 3GPP Rel. 15-compliant 4G vEPC and 5G core components. The Evolved Packet Core (EPC) has been installed and tested in conjunction with Evolved-UMTS Terrestrial Radio Access Network (E-UTRAN) and Wireless Local Area Network (WLAN) radio-access technologies,
5. Support of CUPS [9] – providing scalable E2E separation of control and user planes – in the 5G mobile core. The control-plane (CP) endpoint in the core is the new Access and Mobility Management Function (AMF), and the use-plane (UP) endpoint is a User Plane Function (UPF). The 5G UE maintains separate interfaces to a Macro cell for CP and a Micro cell for UP.
6. Deployment of Orchestrator component OSM and SDN transport, deployment and configuration of Core/Edge VNFs.
7. Provision of secure breakout capability at the network edge with specialised User Plane Function (UPF) for MEC services and non-3GPP networks, has been developed. The platform team has implemented and validated a UPF component that provides a local breakout function to the attached local servers, in support of MEC services. This also allows data connectivity to non-3GPP systems.
8. Design and development of various 5G core network components, in accordance with the 3GPP 5G System Architecture [10]. Specifically, the AMF, Session Management Function (SMF), and UPF components have been developed and deployed, and their interworking has been enabled.

9. Accelerated UPF operations. The UPF software has been enhanced to boost data traffic throughput, using multi-threading techniques, as well as user plane enhancements for the use of network equipment interfaces.
10. Development of the user plane control protocol between SMF and UPF, in accordance with 3GPP specifications. Following the Packet Forwarding Control Protocol (PFCP) in [11], and per the descriptions of SMF and UPF functions in [10], UPF is programmable by the SMF with PFCP.
11. Deployment and reachability-testing (via LoRa and LTE connectivity) of physical MONROE nodes. This allows performing small-scale experiments to calculate KPIs such as latency and throughput of a sub-section of the testbed.
12. Regarding the WiFi access, at this stage, FON is able to proceed with a 3GPP Release 15 based integration. Release 15 is the first release of the 3GPP that standardizes 5G requirements. In the case of non-3GPP access WiFi interface, Rel. 15 only defines what to do in the 5G Non-Standalone (NSA) case, i.e. w.r.t a Rel.15 4G EPC as the core of the 5G mobile network. At present, FON is working using the work-in-progress 3GPP documents, to understand how 3GPP intends to standardize the Y2 interface, the one needed to make the integration between non-3GPP access and 5GC possible.
13. Campus-wide coverage deployment of LoRa GWs.
14. Initial deployment of INFOLYSiS IoT vGW and one mapping VNF for HTTP protocol and the initial deployment of IoT-vGW web GUI. Tests were performed in order to verify the correct operation of the components within the system.

Note, however, that Release A of the platform as described in [2] already included some of the progress initially planned for Release B, namely the development and integration of the core/edge NFVIs with OpenStack [12] and OSM [13].

The OSM (v5) datacentre employs an OpenStack Rocky Virtualized Infrastructure Manager (VIM) for orchestration, hosting and storing VNFs as KVM virtual machines. To integrate a network service's VNF, OpenStack Neutron network provider is used. The VNFs can be configured during instantiation using Cloud-Init scripts.

## 3.3. Phase 2 Accomplishments

### 3.3.1. Integration of Coordination layer components

During Phase 2, the deployment and integration of coordination layer components (Release A) were completed. These include:

- The 5GENESIS Portal.
- The 5GENESIS Experiment Lifecycle Manager (ELCM) and its executive back-end, i.e. Open TAP.
- The 5GENESIS Slice Manager (in-progress).
- The storage (InfluxDB) and visualization (Grafana) components of the monitoring subsystem.
- The dispatcher and the experimenter portal that will allow the upload of the 5GENESIS experiment descriptors and VNFs.

During the remainder of phase 2, the aforementioned components will be interconnected using the integration and testing procedures detailed in D5.1 [14]. Therefore, It will be possible to configure and execute an experiment via the Portal, automatize it by the ELCM, collect the measurements in the monitoring backend and visualize them.

Also, the Release A of the Slice Manager is currently being deployed and integrated with the NFVO (OSM). The deployment and lifecycle management of a simple slice containing a set of VNFs has been verified.

All coordination layer components have been deployed as VMs in the OpenStack core infrastructure in the Data Centre, with the exception of ELCM, which currently is able to run only under Windows. Due to inherent difficulties with launching Windows VMs in the OpenStack infrastructure, this component was installed in a separate physical machine.

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

1. **Prometheus:**
   Prometheus has been successfully installed and runs on port 9090, and few simple commands have been run for testing purposes. It generates the data according to given built-in commands. A screenshot of the command output can be observed in Figure 14.

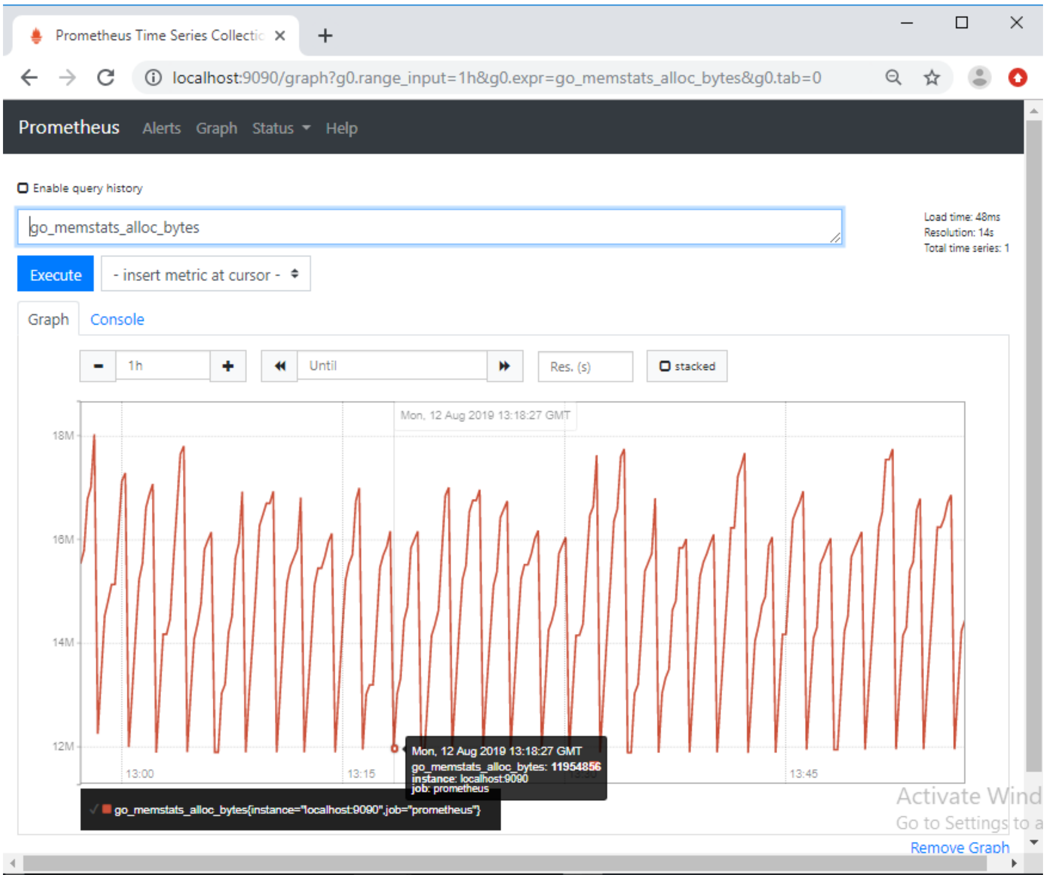



Figure 14: Prometheus generating data from the command: "go_memstats_alloc_bytes"

2. **Grafana:**

Graphana has been successfully downloaded and integrated. It runs on port 3000 and uses the above displayed Prometheus as a data source. Hence, this GUI displays a live graph of Prometheus content. A screenshot of the login page and the dashboard is depicted in Figure 15 and Figure 16 .

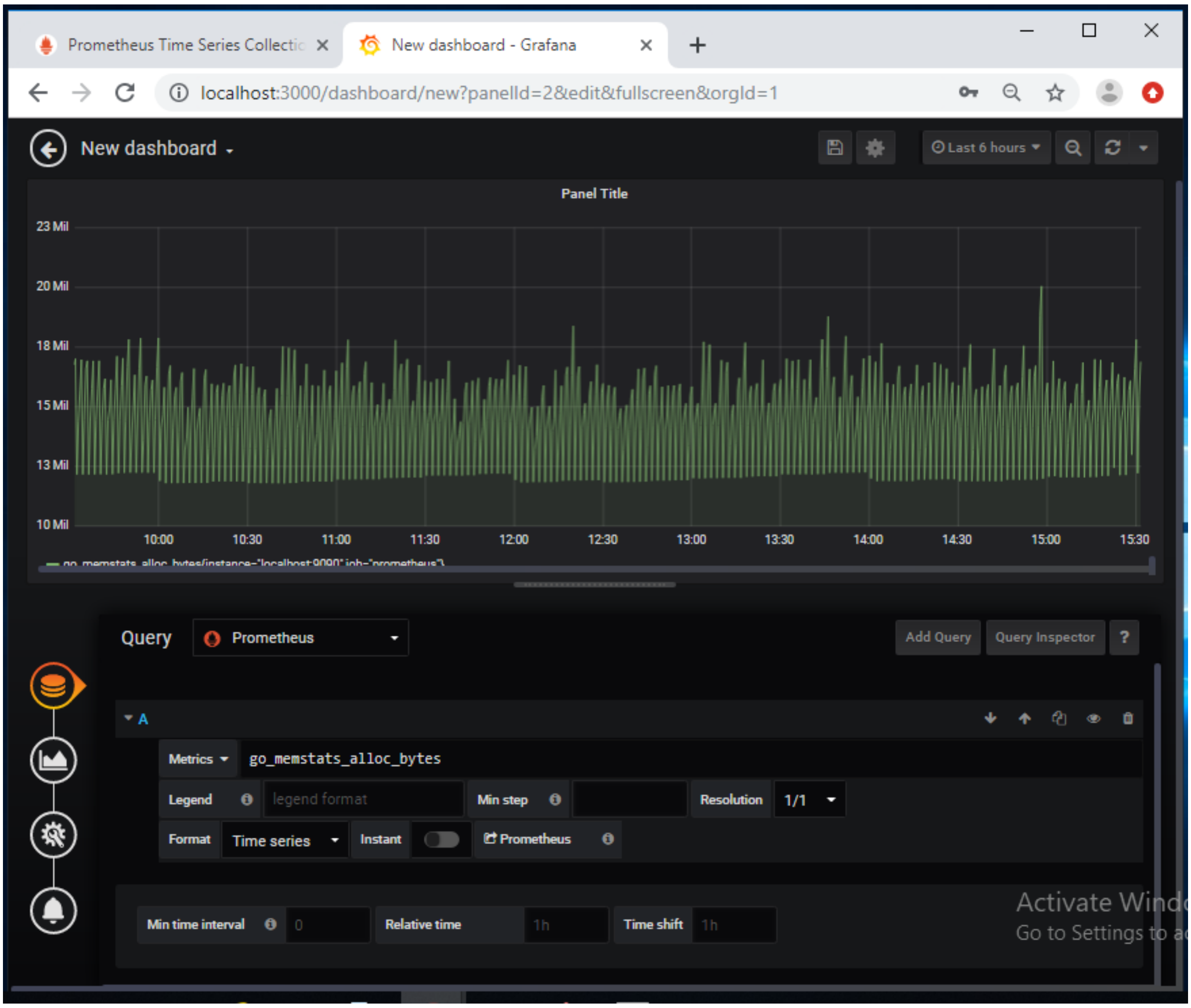


Figure 15: Grafana integrated with Prometheus and retrieving data and displaying.

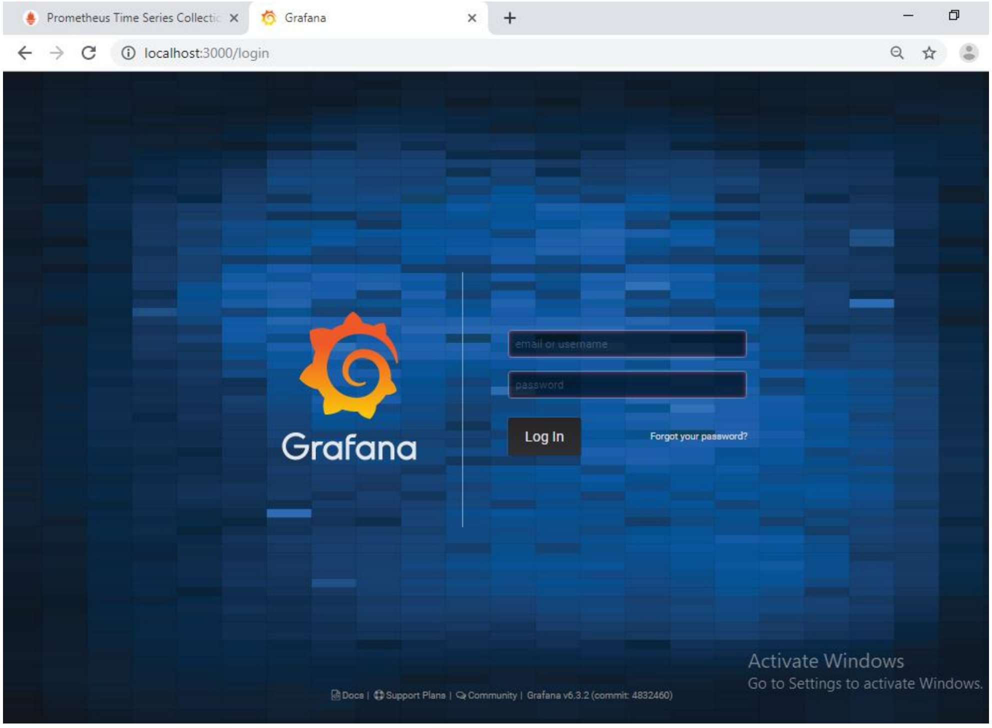



Figure 16: Grafana login page

3. TAP:

TAP has been successfully installed and is ready for integration with other tools for automation. A screenshot of the TAP tool can be observed in **Figure 17**.

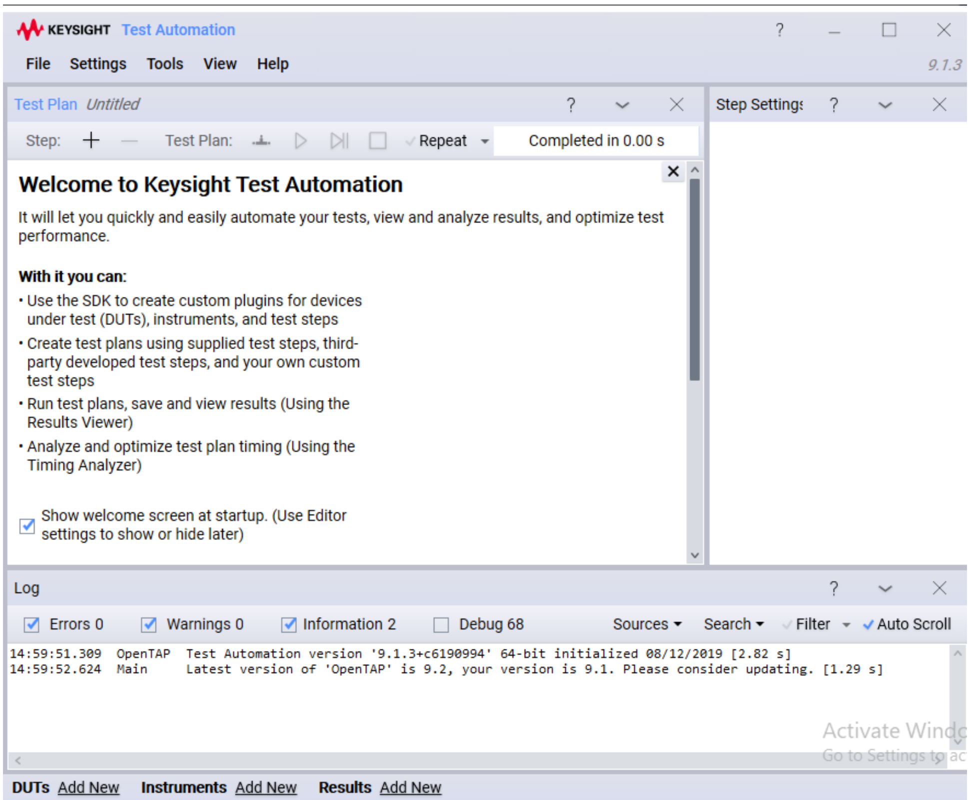



Figure 17: TAP running.

### 4. Portal and ELCM:

Both Portal and ELCM components has been successfully downloaded and integrated. Moreover, different validation test has been performed in order to ensure that they work as expected. Screen shot attached below.

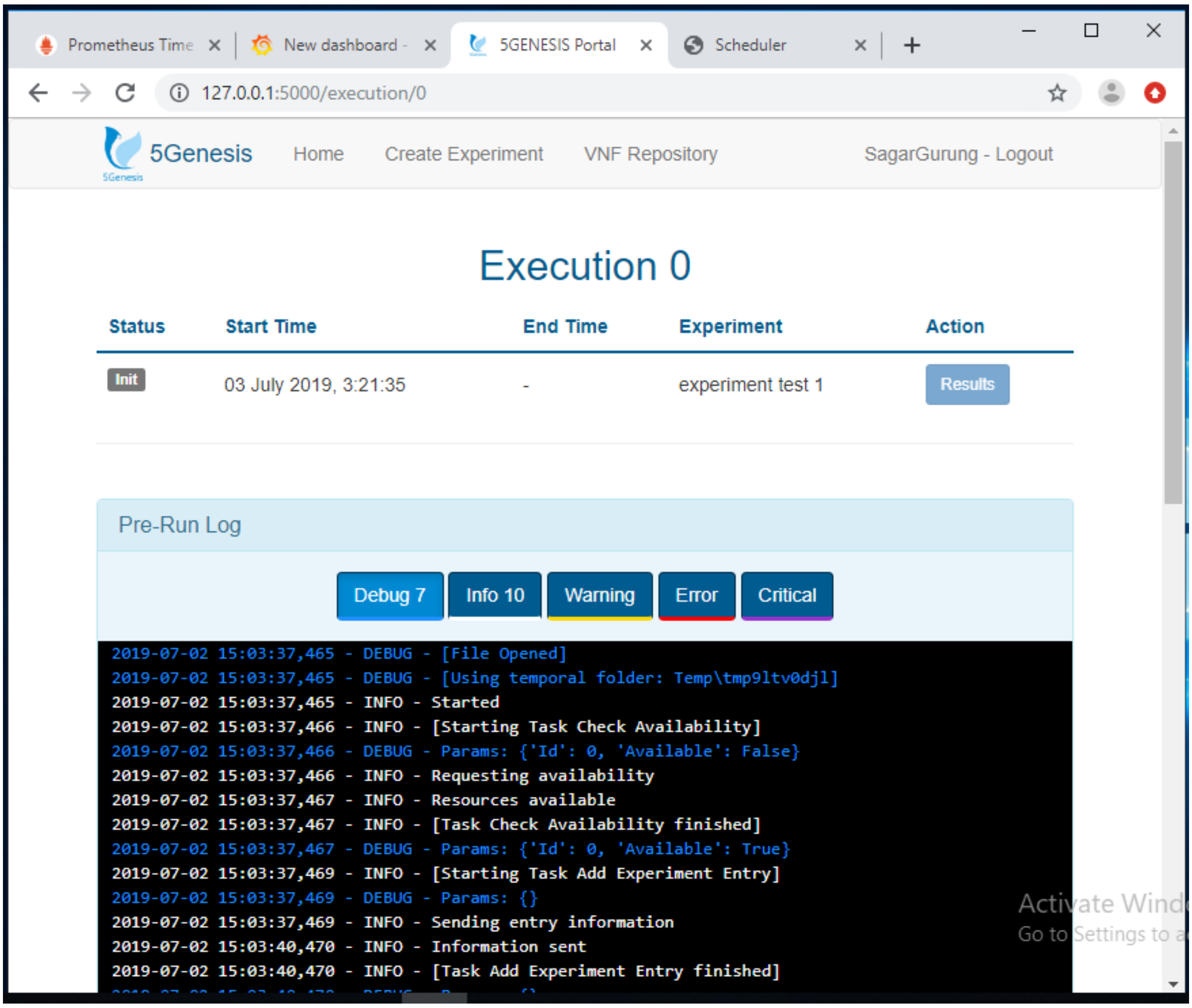



Figure 18: Portal

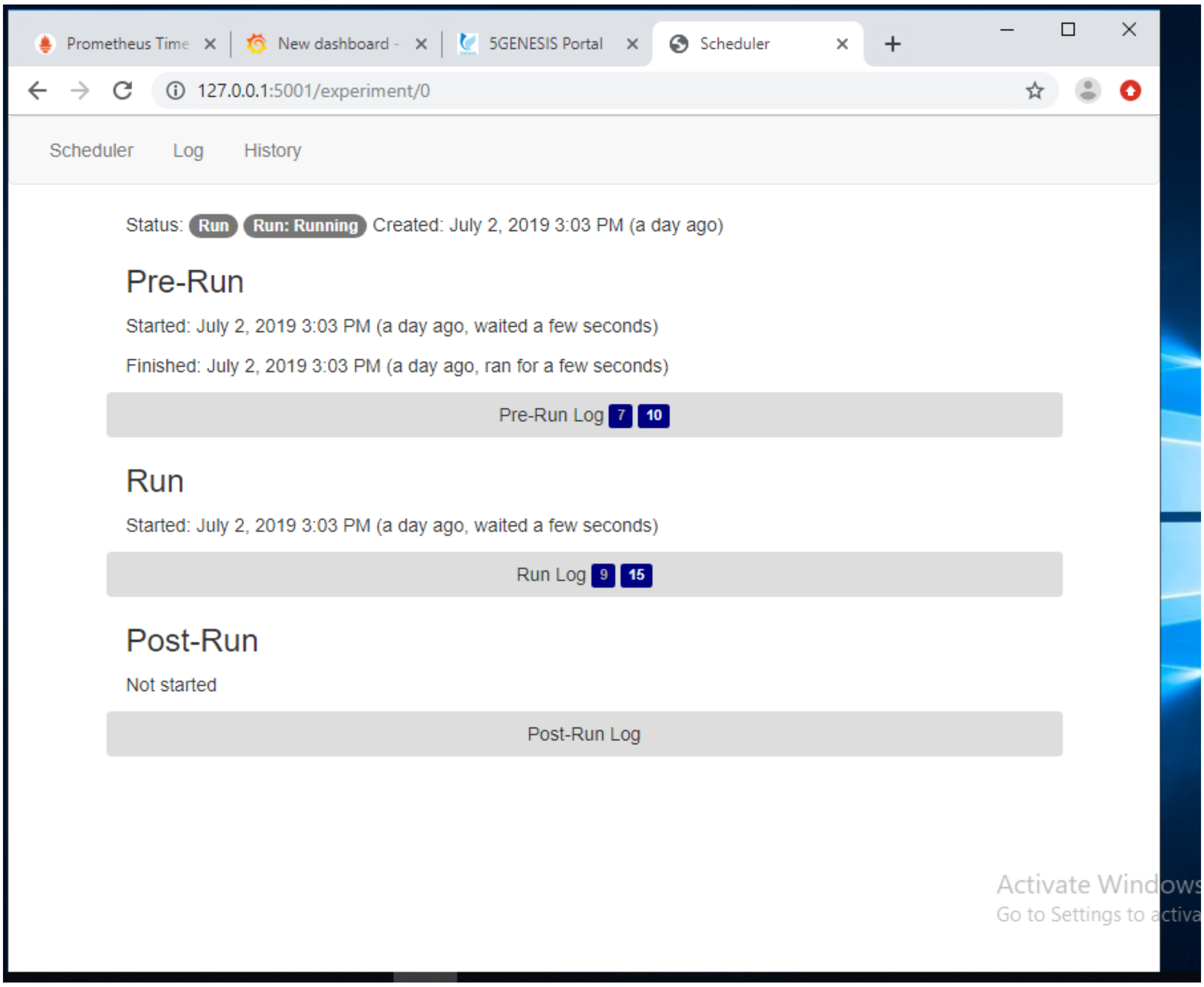



Figure 19: ELCM (Environmental Lifecycle Manager)

5. **Ceilometer[11] and Gnocchi[12]:**
   Ceilometer and Gnocchi have been installed and communicating with each other. Gnocchi is now able to collect and store data into its database and tables. Currently it generates data according to the shown configuration settings under **pipeline.yaml** and **polling.yaml files, depicted in** Figure 20 **and** Figure 21. Integration of **Prometheus** with Ceilometer and Gnocchi is currently in progress.

[11] Ceilometer (a framework for monitoring and metering the OpenStack cloud) provides data collection service with the ability to normalise and transform data. Ceilometer data can be used to provide customer billing, resource tracking, and alarming capabilities across all OpenStack core components.

[12] Gnocchi (https://wiki.openstack.org/wiki/Gnocchi) is a framework to collect, store, manage and query the data collected by Ceilometer.

admin5g@aio-3121: ~

```
admin5g@aio-3121:~$ gnocchi resource list --type instance
+--------------------------------------+----------+----------------------------------+----------------------------------+--------------------------------------
--+----------------------------------+----------+----------------------------------+---------------+---------------------------------------------------------------
---------+--------------+-----------+---------------------------+-------------+-----------+--------------+---------------------------+-------------------------
--------------+--------------------------------------+
| id                                   | type     | project_id                       | user_id                          | original_resource_id
  | started_at                       | ended_at | revision_start                   | revision_end  | creator
         | server_group | deleted_at | launched_at               | flavor_name | host      | display_name | created_at                | image_ref
              | flavor_id                            |
+--------------------------------------+----------+----------------------------------+----------------------------------+--------------------------------------
--+----------------------------------+----------+----------------------------------+---------------+---------------------------------------------------------------
---------+--------------+-----------+---------------------------+-------------+-----------+--------------+---------------------------+-------------------------
--------------+--------------------------------------+
| 86a8758e-a97a-4105-abe2-e9b46f78089a | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | 86a8758e-a97a-4105-abe2-e9b46f78089
a | 2019-08-10T22:55:01.770964+00:00 | None     | 2019-08-10T23:00:18.094877+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-05-20T16:38:39+00:00 | wintest2    | aio-3121  | win1         | 2019-05-20T16:37:44+00:00 | fdf612d7-28e3-4f04-856
1-1db1c663de0d | 660a96a9-43e7-4a7f-b443-a820b9714d3e |
| 58dd0521-1b4f-4712-b3c4-be70d9565343 | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | 58dd0521-1b4f-4712-b3c4-be70d956534
3 | 2019-08-10T22:55:01.509493+00:00 | None     | 2019-08-10T23:00:18.111475+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-04-16T13:03:11+00:00 | 5GEN1       | aio-3121  | INFO3        | 2019-04-16T13:03:04+00:00 | a9328528-32a1-42f0-979
d-a78c241122d6 | 87bdb70d-4124-46c2-ab97-6b77acdb3cf3 |
| 78c825cd-24cc-4ed8-ac5f-980af313c9bf | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | 78c825cd-24cc-4ed8-ac5f-980af313c9b
f | 2019-08-10T22:55:01.292539+00:00 | None     | 2019-08-10T23:00:18.153637+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-04-16T13:05:07+00:00 | 5GEN2       | aio-3121  | INFO5        | 2019-04-16T13:05:01+00:00 | a9328528-32a1-42f0-979
d-a78c241122d6 | 1ad18e50-0f28-4f04-9535-f8814fbaab3a |
| af509733-12c9-4c51-955a-09c6f9abf3af | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | af509733-12c9-4c51-955a-09c6f9abf3a
f | 2019-08-10T22:55:01.679141+00:00 | None     | 2019-08-10T23:00:18.187945+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-03-29T15:00:30+00:00 | demo.small2 | aio-3121  | 5gR15        | 2019-03-29T15:00:22+00:00 | a9328528-32a1-42f0-979
d-a78c241122d6 | 324e46ef-c42f-4485-b403-e07424c8d74d |
| 2fb8f67f-7361-4afc-adcf-fed0b616f007 | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | 2fb8f67f-7361-4afc-adcf-fed0b616f00
7 | 2019-08-10T22:55:01.362216+00:00 | None     | 2019-08-10T23:00:18.226736+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-04-16T13:00:53+00:00 | 5GEN1       | aio-3121  | INFO2        | 2019-04-16T13:00:46+00:00 | a9328528-32a1-42f0-979
d-a78c241122d6 | 87bdb70d-4124-46c2-ab97-6b77acdb3cf3 |
| 03a5d2d0-2525-40e0-9f97-da6d26f6cc8e | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | 03a5d2d0-2525-40e0-9f97-da6d26f6cc8
e | 2019-08-10T22:55:01.887683+00:00 | None     | 2019-08-10T23:00:18.245008+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-04-16T13:04:19+00:00 | 5GEN1       | aio-3121  | INFO4        | 2019-04-16T13:04:11+00:00 | a9328528-32a1-42f0-979
d-a78c241122d6 | 87bdb70d-4124-46c2-ab97-6b77acdb3cf3 |
| 755b12c2-66c0-4c53-bfe8-e24f06bba7a8 | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | 755b12c2-66c0-4c53-bfe8-e24f06bba7a
8 | 2019-08-10T22:55:01.591501+00:00 | None     | 2019-08-10T23:00:18.251417+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-03-29T12:57:45+00:00 | demo.small2 | aio-3121  | 5gR13        | 2019-03-29T12:57:30+00:00 | a9328528-32a1-42f0-979
d-a78c241122d6 | 324e46ef-c42f-4485-b403-e07424c8d74d |
| 90f9b186-09cf-4bbb-9f2e-4982858cd25d | instance | 07f860f8b5a94fabb7846b3d8c9219df | 99382f7e298a4f74807d4acdf0fcb5c1 | 90f9b186-09cf-4bbb-9f2e-4982858cd25
d | 2019-08-10T22:55:01.735161+00:00 | None     | 2019-08-10T23:00:18.266212+00:00 | None          | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd
89ee9d35 | None         | None       | 2019-04-16T12:52:03+00:00 | 5GEN1       | aio-3121  | INFO1        | 2019-04-16T12:51:57+00:00 | a9328528-32a1-42f0-979
```

admin5g@aio-3121: ~

```
+--------------------------------------+----------+----------------------------------+-----------
--+----------------------------------+----------+----------------------------------+-----------
---------+--------------+-----------+---------------------------+-------------+-----------+-----
--------------+--------------------------------------+
admin5g@aio-3121:~$ gnocchi resource show 58dd0521-1b4f-4712-b3c4-be70d9565343
+-----------------------+-------------------------------------------------------------------+
| Field                 | Value                                                             |
+-----------------------+-------------------------------------------------------------------+
| created_by_project_id | 945a9b7a04ba495aae80afdd89ee9d35                                  |
| created_by_user_id    | 2cc627160c3f4e359635c7d10991ffe8                                  |
| creator               | 2cc627160c3f4e359635c7d10991ffe8:945a9b7a04ba495aae80afdd89ee9d35 |
| ended_at              | None                                                              |
| id                    | 58dd0521-1b4f-4712-b3c4-be70d9565343                              |
| metrics               | cpu.delta: 9c1b5e73-569e-423f-a370-1bde5669e7ec                   |
|                       | cpu: 2c553774-df20-4016-84d2-1830aa1cdf11                         |
|                       | cpu_util: 0aa74685-cd1c-4e59-892f-8eef2be7e0cb                    |
|                       | disk.ephemeral.size: beb57791-94c3-47ca-9eda-847af65fa0fc         |
|                       | disk.root.size: fdb8e479-157b-463e-8bb5-642155cc975d              |
|                       | memory.usage: 81bb7f85-5828-4054-8cb6-9a37e186253b                |
|                       | memory: adfb88e2-ec62-4557-86b8-53976a833349                      |
|                       | vcpus: 52d21f26-fad2-4275-adbc-d0fc74aa8a03                       |
| original_resource_id  | 58dd0521-1b4f-4712-b3c4-be70d9565343                              |
| project_id            | 07f860f8b5a94fabb7846b3d8c9219df                                  |
| revision_end          | None                                                              |
| revision_start        | 2019-08-10T23:00:18.111475+00:00                                  |
| started_at            | 2019-08-10T22:55:01.509493+00:00                                  |
| type                  | instance                                                          |
| user_id               | 99382f7e298a4f74807d4acdf0fcb5c1                                  |
+-----------------------+-------------------------------------------------------------------+
```

admin5g@aio-3121: ~

```
admin5g@aio-3121:~$ gnocchi metric list
+--------------------------------------+---------------------+---------------------------------+-----------+--------------------------------------+
| id                                   | archive_policy/name | name                            | unit      | resource_id                          |
+--------------------------------------+---------------------+---------------------------------+-----------+--------------------------------------+
| 00a8181d-4969-46b4-9807-25f1aea30c95 | ceilometer-low-rate | network.incoming.bytes          | B         | 75882683-1ab4-5cfe-9951-b638c485ee96 |
| 016fd466-ccd3-4dc3-bea3-d455b5cc1812 | ceilometer-low      | vcpus                           | vcpu      | 755b12c2-66c0-4c53-bfe8-e24f06bba7a8 |
| 0177022a-699a-49de-9b4d-4dd444cb52ba | ceilometer-low      | network.incoming.packets.rate   | packet/s  | 4563121e-46dd-5762-a745-027ce2dde3c9 |
| 03f25c39-39b5-47e9-84ca-20e12f289e2f | ceilometer-low      | disk.root.size                  | GB        | 90f9b186-09cf-4bbb-9f2e-4982858cd25d |
| 0427a7d7-2c6e-486a-90a6-acb79d0ad23b | ceilometer-low      | disk.root.size                  | GB        | 03a5d2d0-2525-40e0-9f97-da6d26f6cc8e |
| 05213934-b3cb-4657-aba7-34671ca5cf78 | ceilometer-low-rate | disk.device.read.bytes          | B         | eaa51aad-06df-58bd-a986-4d9dc39dca3c |
| 055891c6-29c7-4f21-8514-9f82e76a124d | ceilometer-low-rate | network.incoming.bytes          | B         | b8f434cb-d0dd-58ba-af83-b7a52abd126c |
| 07b4b65c-7739-4719-ab3d-ee964d313fe1 | ceilometer-low      | disk.ephemeral.size             | GB        | 755b12c2-66c0-4c53-bfe8-e24f06bba7a8 |
| 093d6d95-ae19-4335-a3f6-887774012cb8 | ceilometer-low      | cpu_util                        | %         | 78c825cd-24cc-4ed8-ac5f-980af313c9bf |
| 09f26bec-7f97-4e4a-a076-8ba91055558c | ceilometer-low      | network.incoming.packets.rate   | packet/s  | bb13cbe3-046d-510e-a83a-10ebc86477ea |
| 0aa74685-cd1c-4e59-892f-8eef2be7e0cb | ceilometer-low      | cpu_util                        | %         | 58dd0521-1b4f-4712-b3c4-be70d9565343 |
| 0afb3465-953d-400b-8914-a630f98ed4eb | ceilometer-low      | memory.usage                    | MB        | 90f9b186-09cf-4bbb-9f2e-4982858cd25d |
| 0c60255f-ebed-4d97-bbd0-8276f9d9fff3 | ceilometer-low      | cpu.delta                       | ns        | 755b12c2-66c0-4c53-bfe8-e24f06bba7a8 |
| 0d069a07-dfdd-441f-bc8c-4ee37a44f66c | ceilometer-low-rate | network.outgoing.packets        | packet    | bb13cbe3-046d-510e-a83a-10ebc86477ea |
| 10aa8481-05e5-4aad-9a63-c29bd79c8a0c | ceilometer-low-rate | disk.device.write.requests      | request   | f9ff20b4-eba1-5dbe-8d5c-324d0a5b5a75 |
| 1329d79d-9982-45c9-8a0f-4131203cb79b | ceilometer-low      | cpu.delta                       | ns        | 2fb8f67f-7361-4afc-adcf-fed0b616f007 |
| 14e01ab6-571d-415c-b5a6-3b546c71a388 | ceilometer-low-rate | network.incoming.packets        | packet    | b8f434cb-d0dd-58ba-af83-b7a52abd126c |
| 162eee25-ec65-46cb-b81a-32885c6c45e2 | ceilometer-low      | network.outgoing.packets.rate   | packet/s  | c63f153f-ab29-5030-a5c1-a7ff244b4791 |
| 1b13db46-1fb1-4648-9267-b4b947945fc7 | ceilometer-low-rate | cpu                             | ns        | 2fb8f67f-7361-4afc-adcf-fed0b616f007 |
| 1b3588c0-c754-4deb-8318-423514de268e | ceilometer-low      | disk.device.write.bytes.rate    | B/s       | 916255c5-167d-5202-8f0e-0e8378b53bfa |
| 1b7b8181-d87a-4469-be6d-2cfbb6f49eb8 | ceilometer-low-rate | network.outgoing.bytes          | B         | 3a34b37f-9cc5-5be7-b6ac-92611c01a391 |
| 1d80a80f-a94e-4842-b655-98a4fb8892d6 | ceilometer-low      | memory                          | MB        | 2fb8f67f-7361-4afc-adcf-fed0b616f007 |
| 1eea1ec6-28f0-498a-ae47-4df65f941bb1 | ceilometer-low      | memory.usage                    | MB        | 2fb8f67f-7361-4afc-adcf-fed0b616f007 |
| 1f668f1b-10c6-4bc2-b3db-b51bc3d91995 | ceilometer-low      | disk.device.read.bytes.rate     | B/s       | 61a3a898-73db-5279-a46e-5ddbcda5077d |
| 1f7839fc-1a69-460d-a0ef-e0073732f28f | ceilometer-low      | network.outgoing.packets.rate   | packet/s  | 75882683-1ab4-5cfe-9951-b638c485ee96 |
| 1f870a1b-f472-4790-ae97-1ea268ec9f2b | ceilometer-low-rate | disk.device.read.bytes          | B         | f9ff20b4-eba1-5dbe-8d5c-324d0a5b5a75 |
| 209dbde0-bc31-4a6c-bd77-3726e2ec0884 | ceilometer-low      | network.incoming.bytes.rate     | B/s       | 4563121e-46dd-5762-a745-027ce2dde3c9 |
| 23c24f0e-a69b-42ea-a5b8-afe7761f9b43 | ceilometer-low      | memory                          | MB        | 86a8758e-a97a-4105-abe2-e9b46f78089a |
| 25e18c01-ece9-4d9d-bd26-2553a8514031 | ceilometer-low      | network.outgoing.packets.rate   | packet/s  | 4563121e-46dd-5762-a745-027ce2dde3c9 |
| 26586007-3870-4ce9-a33d-93270c94577a | ceilometer-low-rate | disk.device.read.requests       | request   | 6278f6f5-06a2-5996-b096-59ab593bd747 |
| 26a3c85a-3a24-4a93-9324-ffb316059150 | ceilometer-low-rate | disk.device.write.bytes         | B         | 1baed14c-0473-5ce5-a8ab-4d08c8b813d8 |
| 289c9d3c-ac2e-46d8-93ef-56209d1e314a | ceilometer-low      | vcpus                           | vcpu      | 2fb8f67f-7361-4afc-adcf-fed0b616f007 |
| 292f69d7-ccd2-4816-81f2-7f9c375e71d0 | ceilometer-low-rate | network.outgoing.packets        | packet    | b8f434cb-d0dd-58ba-af83-b7a52abd126c |
| 295e2c8a-b398-414b-9a07-fd238a631f87 | ceilometer-low      | disk.device.read.requests.rate  | request/s | 61a3a898-73db-5279-a46e-5ddbcda5077d |
| 29bd0f60-0f56-40c0-895d-3ad723c1cfa3 | ceilometer-low-rate | disk.device.read.requests       | request   | 1baed14c-0473-5ce5-a8ab-4d08c8b813d8 |
| 2aa8d376-4b57-42da-8b0c-b71906303e23 | ceilometer-low      | network.outgoing.bytes.rate     | B/s       | b8f434cb-d0dd-58ba-af83-b7a52abd126c |
| 2b557b1a-abd2-4157-b686-1f283a23658d | ceilometer-low      | disk.device.read.bytes.rate     | B/s       | 6278f6f5-06a2-5996-b096-59ab593bd747 |
| 2c553774-df20-4016-84d2-1830aa1cdf11 | ceilometer-low-rate | cpu                             | ns        | 58dd0521-1b4f-4712-b3c4-be70d9565343 |
| 2c760d37-07e0-4321-9f19-7621bbb28330 | ceilometer-low      | disk.root.size                  | GB        | 86a8758e-a97a-4105-abe2-e9b46f78089a |
| 32410242-861f-4d70-801b-221aaa8c7512 | ceilometer-low-rate | network.incoming.bytes          | B         | 84cce4b3-ad72-53b7-9d73-7849c297c0ec |
```

root@aio-3121: /etc/ceilometer


```
---
sources:
    - name: meter_source
      meters:
          - "*"
      sinks:
          - meter_sink
    - name: cpu_source
      meters:
          - "cpu"
      sinks:
          - cpu_sink
          - cpu_delta_sink
    - name: disk_source
      meters:
          - "disk.read.bytes"
          - "disk.read.requests"
          - "disk.write.bytes"
          - "disk.write.requests"
          - "disk.device.read.bytes"
          - "disk.device.read.requests"
          - "disk.device.write.bytes"
          - "disk.device.write.requests"
      sinks:
          - disk_sink
    - name: network_source
      meters:
          - "network.incoming.bytes"
          - "network.incoming.packets"
          - "network.outgoing.bytes"
          - "network.outgoing.packets"
      sinks:
          - network_sink
sinks:
    - name: meter_sink
      transformers:
      publishers:
          - gnocchi://
    - name: cpu_sink
      transformers:
          - name: "rate_of_change"
            parameters:
                target:
"pipeline.yaml" 89L, 2507C
```


Figure 20: pipeline.yaml

root@aio-3121: /etc/ceilometer

```
---
sources:
    - name: some_pollsters
      interval: 300
      meters:
        - cpu
        - cpu_l3_cache
        - memory.usage
        - network.incoming.bytes
        - network.incoming.packets
        - network.outgoing.bytes
        - network.outgoing.packets
        - disk.device.read.bytes
        - disk.device.read.requests
        - disk.device.write.bytes
        - disk.device.write.requests
        - hardware.cpu.util
        - hardware.memory.used
        - hardware.memory.total
        - hardware.memory.buffer
        - hardware.memory.cached
        - hardware.memory.swap.avail
        - hardware.memory.swap.total
        - hardware.system_stats.io.outgoing.blocks
        - hardware.system_stats.io.incoming.blocks
        - hardware.network.ip.incoming.datagrams
        - hardware.network.ip.outgoing.datagrams
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"polling.yaml" 27L, 841C
```

Figure 21: polling.yaml

### 3.3.2. Integration of INFOLYSiS IoT-vGW and VNF components

Small-scale IoT sensor deployment (indoors, for experimentation) and INFOLYSiS IoT-vGW deployment and mapping VNFs integration with Surrey sensors and infrastructure have been completed. SDN controller (Small scale with real sensors provided by the Surrey Platform) supports all three of the following protocols: HyperText Transfer Protocol (HTTP), Constrained Application Protocol (CoAP) or MQ Telemetry Transport (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 User Datagram Protocol (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 REpresentational State Transfer (REST) Application Programming Interface (API) that is available on the vGW.

More specifically, a detailed explanation with screenshots follows for the functionality and available resources of the INFOLYSiS VNFs. All VNFs are on the same SDN network segment and use Internet Protocol (IP) addresses in the range of 10.5.31.0/24. The 3 mappers have the same Central Processing Unit (CPU), memory and disk, while the vGW VNF has double the resources in order to handle the heavy workload of aggregating all the data and disseminating them to third parties in real-time.

The VNFs setup by INFOLYSiS that is currently on the Surrey Platform is:

- INFOLYSIS HTTP Mapping VNF: Mapping of HTTP IoT data to plain UDP, providing interoperability. The IP of that VNF is 10.5.31.67. The interoperability translation of IoT data is handled from a PHP script running on Apache2 web server. It can handle many simultaneous data packets without any problems, and it consumes very low system resources, as shown in Figure 22. Figure 23 is a screenshot of the Apache2 access log that shows every HTTP IoT data packet that invokes the PHP script in order to be translated to plain UDP.

```
18130 root      20   0  258M 27304 21048 S  0.0  0.5  0:08.28 /usr/sbin/apache2 -k start
```

Figure 22: Resource usage of HTTP mapping process

```
10.5.31.101 - - [03/Jan/2020:06:29:20 +0000] "POST /iotdata.php HTTP/1.0" 200 221 "-" "-"
10.5.31.101 - - [03/Jan/2020:06:29:31 +0000] "POST /iotdata.php HTTP/1.0" 200 221 "-" "-"
10.5.31.101 - - [03/Jan/2020:06:29:42 +0000] "POST /iotdata.php HTTP/1.0" 200 221 "-" "-"
10.5.31.101 - - [03/Jan/2020:06:29:53 +0000] "POST /iotdata.php HTTP/1.0" 200 221 "-" "-"
10.5.31.101 - - [03/Jan/2020:06:30:05 +0000] "POST /iotdata.php HTTP/1.0" 200 221 "-" "-"
```

Figure 23: Translation log of HTTP IoT packets

- INFOLYSIS CoAP Mapping VNF: Mapping of CoAP IoT data to plain UDP, providing interoperability. The IP of that VNF is 10.5.31.68. CoAP handling is done by using the Python programming language and the CoAPthon library. The CoAP IoT data packets are accepted by the python script (Figure 24) and then instantly translated to UDP and sent to the vGW. It also uses minimal system resources as shown in Figure 25.


```
[+] 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": ""}}
```

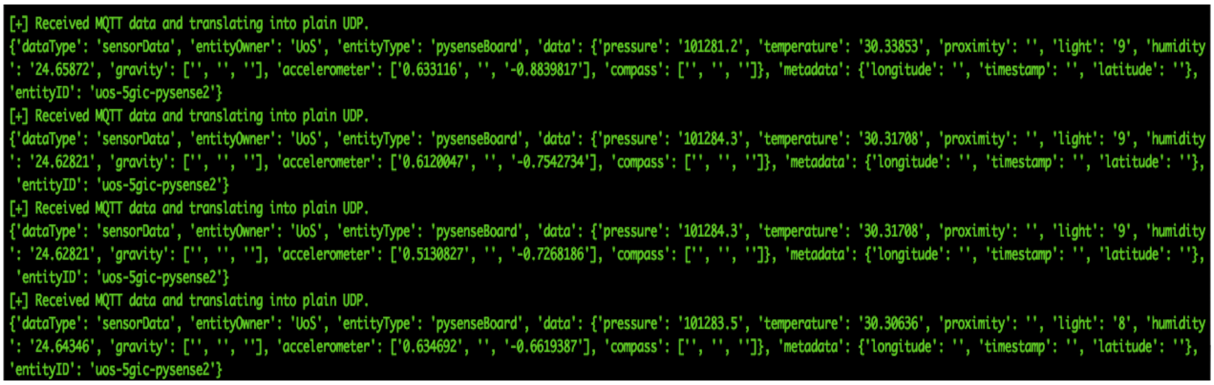

Figure 24: Translation log of CoAP IoT packets

```
25140 root      20   0  257M 13540  6948 S  0.0  0.2  1h21:10 python cserver.py
```

Figure 25: Resource usage of CoAP mapping process

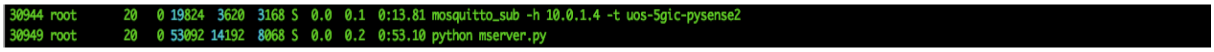
- INFOLYSIS MQTT Mapping VNF: Mapping of MQTT IoT data to plain UDP, providing interoperability. The IP of that VNF is 10.5.31.69. MQTT handling is done by using a

Mosquitto broker and Python script. The MQTT IoT data are published to the Mosquitto broker and the Python script subscribes to those and after it fetches the IoT data it translates them to plain UDP.  system resource usage is also kept very low as depicted in Figure 27.



```
[+] Received MQTT data and translating into plain UDP.
{'dataType': 'sensorData', 'entityOwner': 'UoS', 'entityType': 'pysenseBoard', 'data': {'pressure': '101281.2', 'temperature': '30.33853', 'proximity': '', 'light': '9', 'humidity
': '24.65872', 'gravity': ['', '', ''], 'accelerometer': ['0.633116', '', '-0.8839817'], 'compass': ['', '', '']}, 'metadata': {'longitude': '', 'timestamp': '', 'latitude': ''},
'entityID': 'uos-5gic-pysense2'}
[+] Received MQTT data and translating into plain UDP.
{'dataType': 'sensorData', 'entityOwner': 'UoS', 'entityType': 'pysenseBoard', 'data': {'pressure': '101284.3', 'temperature': '30.31708', 'proximity': '', 'light': '9', 'humidity
': '24.62821', 'gravity': ['', '', ''], 'accelerometer': ['0.6120047', '', '-0.7542734'], 'compass': ['', '', '']}, 'metadata': {'longitude': '', 'timestamp': '', 'latitude': ''},
 'entityID': 'uos-5gic-pysense2'}
[+] Received MQTT data and translating into plain UDP.
{'dataType': 'sensorData', 'entityOwner': 'UoS', 'entityType': 'pysenseBoard', 'data': {'pressure': '101284.3', 'temperature': '30.31708', 'proximity': '', 'light': '9', 'humidity
': '24.62821', 'gravity': ['', '', ''], 'accelerometer': ['0.5130827', '', '-0.7268186'], 'compass': ['', '', '']}, 'metadata': {'longitude': '', 'timestamp': '', 'latitude': ''},
 'entityID': 'uos-5gic-pysense2'}
[+] Received MQTT data and translating into plain UDP.
{'dataType': 'sensorData', 'entityOwner': 'UoS', 'entityType': 'pysenseBoard', 'data': {'pressure': '101283.5', 'temperature': '30.30636', 'proximity': '', 'light': '8', 'humidity
': '24.64346', 'gravity': ['', '', ''], 'accelerometer': ['0.634692', '', '-0.6619387'], 'compass': ['', '', '']}, 'metadata': {'longitude': '', 'timestamp': '', 'latitude': ''},
'entityID': 'uos-5gic-pysense2'}
```

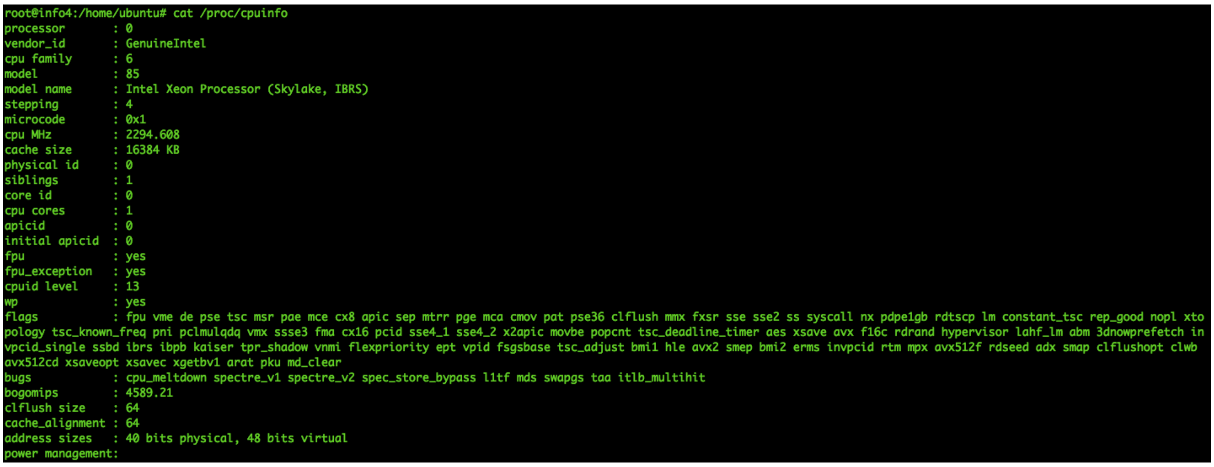
Figure 26: Translation log of MQTT IoT packets



```
30944 root      20   0 19824  3620  3168 S  0.0  0.1   0:13.81 mosquitto_sub -h 10.0.1.4 -t uos-5gic-pysense2
30949 root      20   0 53092 14192  8068 S  0.0  0.2   0:53.10 python mserver.py
```

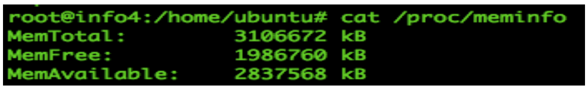
Figure 27: Resource usage of  MQTT mapping processes

All three of the aforementioned mapping VNFs have the same system resources. Those are 2 CPU cores 2.3GHz, 3GB Random Access Memory (RAM) and 30GB of disk space, as shown in Figure 28 (2 cores of the same specifications), Figure 29 and Figure 30, respectively.



```
root@info4:/home/ubuntu# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 85
model name      : Intel Xeon Processor (Skylake, IBRS)
stepping        : 4
microcode       : 0x1
cpu MHz         : 2294.608
cache size      : 16384 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xto
pology tsc_known_freq pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch in
vpcid_single ssbd ibrs ibpb kaiser tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm mpx avx512f rdseed adx smap clflushopt clwb
avx512cd xsaveopt xsavec xgetbv1 arat pku md_clear
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapgs taa itlb_multihit
bogomips        : 4589.21
clflush size    : 64
cache_alignment : 64
address sizes   : 40 bits physical, 48 bits virtual
power management:
```

Figure 28: CPU capabilities of mapping VNFs



```
root@info4:/home/ubuntu# cat /proc/meminfo
MemTotal:        3106672 kB
MemFree:         1986760 kB
MemAvailable:    2837568 kB
```

Figure 29: Memory information of mapping VNFs

```
root@info4:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.5G     0  1.5G   0% /dev
tmpfs           304M   12M  292M   4% /run
/dev/vda1        30G  3.9G   26G  14% /
tmpfs           1.5G     0  1.5G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.5G     0  1.5G   0% /sys/fs/cgroup
tmpfs           304M     0  304M   0% /run/user/1000
```

Figure 30: Disk information of mapping VNFs

- INFOLYSIS IoT vGW: The Final destination of the interoperable IoT data on the INFOLYSiS system. A web Graphical User Interface (GUI), see Figure 31 and Figure 32 is provided in order for the user to be able to monitor the data in real-time and also ensure that all VNFs are in operation. Also, the interoperable IoT data that originate from the mapping VNFs (log of incoming data is depicted in Figure 33 are saved in a MySQL database (Figure 34 shows the data structure of the table, where the interoperable IoT data is saved and Figure 35 shows 20 rows of that table). Those IoT data are available to third parties for future extensions and integrations using UDP streams and a RESTful API.

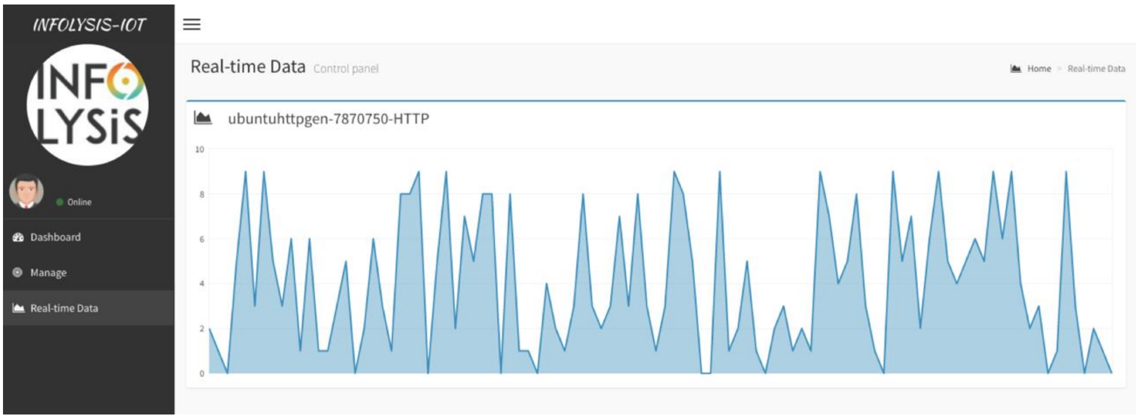



Figure 31: Example of IoT data originating from HTTP protocol

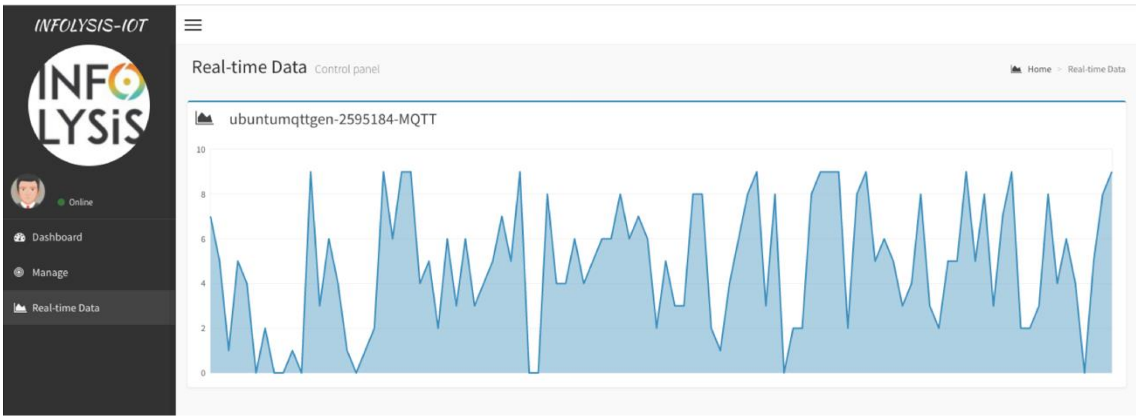



Figure 32: Example of IoT data originating from MQTT protocol

```
[*] Received UDP data: [HTTP:1591] with sequence number 1591
[*] Received UDP data: [COAP:1590] with sequence number 1590
[*] Received UDP data: [MQTT:1592] with sequence number 1592
[*] Received UDP data: [HTTP:1594] with sequence number 1594
[*] Received UDP data: [MQTT:1595] with sequence number 1595
[*] Received UDP data: [COAP:1593] with sequence number 1593
[*] Received UDP data: [HTTP:1597] with sequence number 1597
[*] Received UDP data: [COAP:1596] with sequence number 1596
[*] Received UDP data: [MQTT:1598] with sequence number 1598
```

Figure 33: Log of incoming IoT packets with UDP in the vGW

mysql> describe iotdata;

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int(10) unsigned | NO | PRI | NULL | auto_increment |
| pressure | tinytext | YES | | NULL | |
| temperature | tinytext | YES | | NULL | |
| light | tinytext | YES | | NULL | |
| humidity | tinytext | YES | | NULL | |
| roll | tinytext | YES | | NULL | |
| pitch | tinytext | YES | | NULL | |
| timestamp | tinytext | YES | | NULL | |
| board | tinytext | YES | | NULL | |
| maker | tinytext | YES | | NULL | |
| protocol | tinytext | YES | | NULL | |
| dataSource | tinytext | YES | | NULL | |

12 rows in set (0.00 sec)

Figure 34: Database table structure in vGW

mysql> mysql> select * from iotdata2 order by id desc limit 0,20;

| id | entityID | entityType | entityOwner | dataType | pressure | temperature | light | humidity | proximity | gravity | accelometer | compass | timestamp | latitude | longitude | protocol | dataSource |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1053365 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101421.5 | 29.4269 | 1 | 24.76553 | | | 0.5057054 | | 2020-01-17 19:50:45 | | | MQTT | WIFI |
| 1053364 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101421.5 | 29.4269 | 1 | 24.76553 | | | 0.5057054 | | 2020-01-17 19:50:45 | | | MQTT | WIFI |
| 1053363 | uos-5gic-pysense1 | pysenseBoard | UoS | sensorData | 101426.0 | 33.30938 | 4 | 19.5928 | | | -0.8582994 | | 2020-01-17 19:50:44 | | | HTTP | WIFI |
| 1053362 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101427.0 | 29.4269 | 1 | 24.75791 | | | 0.6407859 | | 2020-01-17 19:50:34 | | | MQTT | WIFI |
| 1053361 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101427.0 | 29.4269 | 1 | 24.75791 | | | 0.6407859 | | 2020-01-17 19:50:34 | | | MQTT | WIFI |
| 1053360 | uos-5gic-pysense1 | pysenseBoard | UoS | sensorData | 101428.3 | 33.32011 | 4 | 19.5928 | | | -0.8504541 | | 2020-01-17 19:50:33 | | | HTTP | WIFI |
| 1053359 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101423.3 | 29.384 | 1 | 24.75028 | | | 0.7207085 | | 2020-01-17 19:50:23 | | | MQTT | WIFI |
| 1053358 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101423.3 | 29.384 | 1 | 24.75028 | | | 0.7207085 | | 2020-01-17 19:50:23 | | | MQTT | WIFI |
| 1053357 | uos-5gic-pysense1 | pysenseBoard | UoS | sensorData | 101426.5 | 33.34156 | 4 | 19.5928 | | | -0.7615909 | | 2020-01-17 19:50:22 | | | HTTP | WIFI |
| 1053356 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101425.0 | 29.40545 | 1 | 24.75791 | | | 0.5551466 | | 2020-01-17 19:50:12 | | | MQTT | WIFI |
| 1053355 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101425.0 | 29.40545 | 1 | 24.75791 | | | 0.5551466 | | 2020-01-17 19:50:12 | | | MQTT | WIFI |
| 1053354 | uos-5gic-pysense1 | pysenseBoard | UoS | sensorData | 101429.2 | 33.33083 | 4 | 19.57755 | | | -0.7186646 | | 2020-01-17 19:50:11 | | | HTTP | WIFI |
| 1053353 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101423.3 | 29.40545 | 1 | 24.75791 | | | 0.6837561 | | 2020-01-17 19:50:01 | | | MQTT | WIFI |
| 1053352 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101423.3 | 29.40545 | 1 | 24.75791 | | | 0.6837561 | | 2020-01-17 19:50:01 | | | MQTT | WIFI |
| 1053351 | uos-5gic-pysense1 | pysenseBoard | UoS | sensorData | 101427.5 | 33.34156 | 4 | 19.60043 | | | -0.8220074 | | 2020-01-17 19:50:00 | | | HTTP | WIFI |
| 1053350 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101423.0 | 29.4269 | 1 | 24.76553 | | | 0.5409802 | | 2020-01-17 19:49:49 | | | MQTT | WIFI |
| 1053349 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101423.0 | 29.4269 | 1 | 24.76553 | | | 0.5409802 | | 2020-01-17 19:49:49 | | | MQTT | WIFI |
| 1053348 | uos-5gic-pysense1 | pysenseBoard | UoS | sensorData | 101426.5 | 33.30938 | 4 | 19.58517 | | | -0.8579885 | | 2020-01-17 19:49:49 | | | HTTP | WIFI |
| 1053347 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101424.5 | 29.37327 | 1 | 24.74265 | | | 0.5631028 | | 2020-01-17 19:49:38 | | | MQTT | WIFI |
| 1053346 | uos-5gic-pysense2 | pysenseBoard | UoS | sensorData | 101424.5 | 29.37327 | 1 | 24.74265 | | | 0.5631028 | | 2020-01-17 19:49:38 | | | MQTT | WIFI |

20 rows in set (0.00 sec)

Figure 35: Interoperable IoT data saved in the database in vGW

The IP address of the vGW VNF is 10.5.31.70 and the system resources that it uses are 4 CPU cores 2.3GHz, 6GB RAM and 40GB of disk space, depicted in Figure 36 (4 cores of the same specs), Figure 36 and Figure 38 respectively.

```
root@info5:/home/ubuntu# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 85
model name      : Intel Xeon Processor (Skylake, IBRS)
stepping        : 4
microcode       : 0x1
cpu MHz         : 2294.608
cache size      : 16384 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtop
ology tsc_known_freq pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch invp
cid_single ssbd ibrs ibpb kaiser tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm mpx avx512f rdseed adx smap clflushopt clwb avx
512cd xsaveopt xsavec xgetbv1 arat pku md_clear
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapgs taa itlb_multihit
bogomips        : 4589.21
clflush size    : 64
cache_alignment : 64
address sizes   : 40 bits physical, 48 bits virtual
power management:
```

Figure 36: Log of incoming IoT packets with UDP in the vGW

```
root@info5:/home/ubuntu# cat /proc/meminfo
MemTotal:        6065248 kB
MemFree:         3913832 kB
MemAvailable:    5275228 kB
```

Figure 37: Memory information of mapping VNFs

```
root@info5:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.9G     0  2.9G   0% /dev
tmpfs           593M   31M  562M   6% /run
/dev/vda1        39G  5.2G   34G  14% /
tmpfs           2.9G     0  2.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           2.9G     0  2.9G   0% /sys/fs/cgroup
tmpfs           593M     0  593M   0% /run/user/1000
```

Figure 38: Disk information of mapping VNFs

### 3.3.3. 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 "The Things Networks" platform (TTN) 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.

### 3.3.4. Miscellaneous

Other achievements accomplished during Phase 2 include:

- Establishment of Virtual Private Network (VPN) for secure access to the Surrey Platform (a.k.a the "5GENESIS Island") and for inter-platform connectivity,

- Test and verification of core/edge VNFs developed in-house,
- 60 GHz mmWave Backhaul (BH) testing, and set of indoor and outdoor measurements,
- Testing of context-aware networking[13] capability in the 5G Core.

## 3.4. Next Milestones

The Phase 3 milestones planned for the Surrey Platform are the following:

1. Integration of the next Release of the 5GENESIS Facility components.
2. Final testing of INFOLYSiS vGW under heavy load of data and performing the appropriate optimizations where needed.
3. Extension of the INFOLYSiS vGW API to support more queries and variables in order to disseminate IoT data more efficiently.
4. Integration of additional IoT sensor data flows into the Surrey IoT network.
5. Integration of NEAT policy system on the UE with the Slice Manager.
6. Integration of additional IoT sensor data flows into the Surrey IoT network.
7. Interconnection and support of the 5GENESIS Slice Manager on the Surrey 5GC (via east-west interface).
8. 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.
9. Testing of a massive Machine Type Communication (mMTC) slice instantiation (in parallel with eMBB slice).
10. Indoor/Outdoor FON WiFi AP deployment and integration testing with Surrey 5GC - FON and Surrey 5GC will support N3IWF[14] related interfaces. – The development of the N3IWF interfaces (on the UE & core-network sides, by both FON and Surrey teams is currently in progress). The deployment of the FON APs in the Surrey premises, and the integration testing of N3IWF interfaces are scheduled for March 2020.
11. Deployment and testing of: i) APEX policy management solution and analytics, ii) NEAT and iii) WSMP from FON, are re-scheduled for phase 3.
12. Installation and integration of REL and ECM 5G-NR solutions at the Surrey premises, and interconnection with the 5GC.
13. Final testing of the Surrey Platform use case in a large-scale event.

[13] This feature has been developed and will be deployed on the Surrey Platform. E2E messages have been designed and developed for communication between the prototype 5G UE and the new 5G core network components. This feature will enable context-aware messaging between the UE and the core, which in turn provides input for dynamic network slicing decisions (user application context, and mobility context) and for user plane anchoring decisions (mobility context).

[14] "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. USE CASE 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, user density and service creation time.

### 4.1.1. Use Case 1 - Massive IoT for Large-Scale Public Events

The core scenario planned for the Surrey use case demonstration will be a public large-scale event taking place on the University of Surrey Stag Hill Campus in Guildford. The campus hosts usually around 16.000 students and 2.000 employees, and the university holds regular large events, including graduation ceremonies and open days, during which up to 2.000-4.000 visitors are on campus at the same time. The procedure is to equip a subset of people on campus with devices that can access and make use of the Surrey Platform and its services. The range of services that will be provided is a mix between high data rate multimedia services (upload and download) and a continuous flow of low bandwidth sensor readings collected through body and environment sensors that will be densely deployed. The objective is to achieve the ITU target density of 1 million devices per square kilometre (which translates, at an even distribution to 1 device per square meter). For this use case, it is planned to be used between 30 and 50 real LoRa capable devices and emulation (using a traffic emulator on the Surrey Platform) of approximately 500.000 devices, covering together around half a square kilometre.

The services provided and delivered over the same infrastructure will be a mix between massive MTC and eMBB. The Surrey Platform will collect, analyse and process the multimedia and sensor content in real-time.

#### Use Case specific Platform Extensions

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 will be 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]** (WiFi Service Management Platform): 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;
- **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.

### Use case 1 - Topology and Architecture

The Surrey Platform follows the 5GENESIS architecture proposed in Deliverable D2.2 [8]. The Platform and main components will be hosted in and around the premises of the 5G Innovation Centre at the University of Surrey, within this area connections will be via the 5GENESIS branch of the 5GIC carrier-grade testbed.

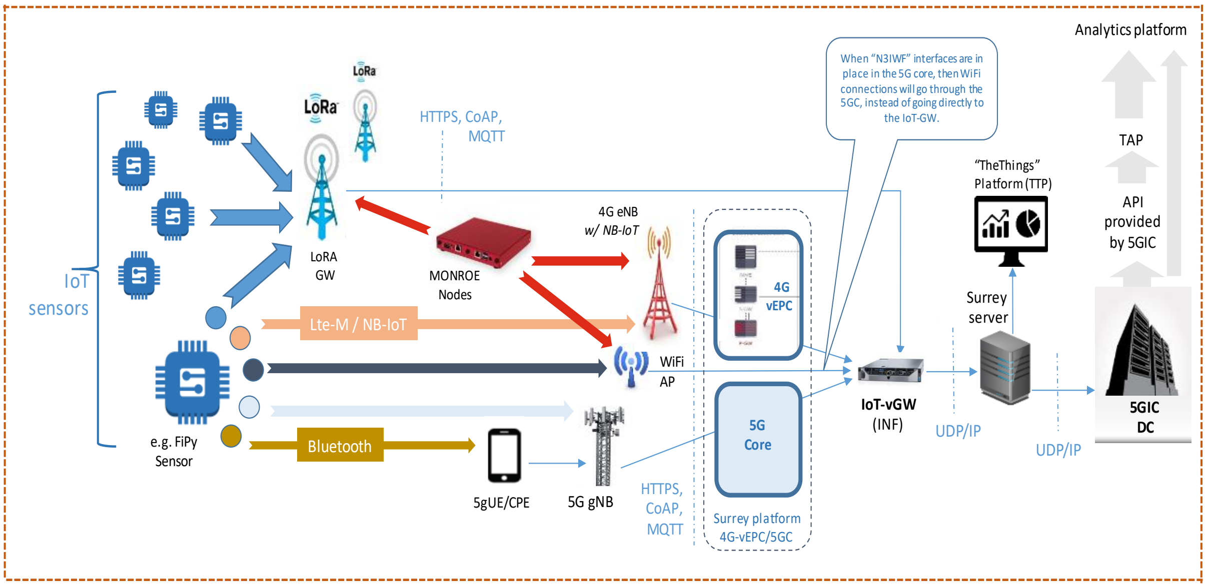



Figure 39. Network topology for the use case scenario at the 5GIC/ICS Site

Figure 39 shows the workflow of the information within the platform, as presented in [2]. 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. This data is collected and transmitted using one or more of the available air interfaces and is 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 includes connection of two physical monitoring (MONROE) nodes, integration of additional air interfaces (NB-IoT, NR, LTE-A, WiFi, Bluetooth, LoRa) and the demonstration of multi-RAT connectivity.

## 4.2. Use Case-related Phase 2 Accomplishments

The use-case related achievements during Phase 2, are as follows:

- Integration of INFOLYSiS IoT-vGW & interoperability testing :
    - INFOLYSIS HTTP Mapping VNF: Mapping of HTTP IoT data to plain UDP
    - INFOLYSIS CoAP Mapping VNF: Mapping of CoAP IoT data to plain UDP
    - INFOLYSIS MQTT Mapping VNF: Mapping of MQTT IoT data to plain UDP
- Deployment and integration of 5GENESIS coordination-layer components:
    - The Experiment LifeCycle Manager (ELCM)
    - dispatcher and the experimenter Portal
    - TAP
    - Prometheus
- Physical MONROE nodes deployment and testing.
- Implementation and deployment of Full Rel.15 compliant 5GC.
- Surrey platform SW upgrade to support NB-IoT Rel .15.
- Development of N3IWF interfaces (NWu & Y2) by FON & Surrey platform team (In progress).

## 4.3. Next Milestones

The use-case related milestones during phase 3, are as follows:

- Full testing of 5GENESIS coordination-layer components:
    - Keysight TAP; implementing the ELCM Integration with Slice Manager.
    - SliceManager interfacing with OSM, VIM, and WIM based on SDN.
    - Monitoring based on Prometheus & Ceilometer
    - Deployment of analytics solutions
- Probe deployment (based on MONROE node).
- First integration of 5GNR (RunEl) & OAI based on 5G UE (Eurecom).
- Deployment of Full Rel.16 compliant 5GC.
- Testing of all three N3IWF interfaces; interworking testing by FON & Surrey platform teams.
- Large-scale testing and demonstration of the Surrey use-case on campus site.
- Testing of INFOLYSiS IoT-vGW under large-scale sensor deployment.
- Reporting and analysis of collected use-case specific KPIs.

# 5. CONCLUSIONS

This document is the second deliverable reporting on the activities regarding the preparation, update and operation of the 5GENESIS Surrey Platform during Phase 2 of the 5GENESIS project. During this Phase, Release A of the 5GENESIS Facility, i.e., Coordination Layer and Slice Manager, has been integrated within the Surrey Platform, while a small-scale 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), as well as non-3GPP networks, as well as a powerful 3GPP Rel.15-compliant 5G core.

During the last three months of Phase 2, the necessary testing will take place, with the aim to guarantee the proper operation of all components and services, while the integration of more mature versions of the various components will be performed.

The second release of the Surrey Platform will be used for the second round of experiments until March 2020, and the report on the measured KPIs will be available in deliverable D6.2. This document will be followed by the final deliverable describing the activities during Phase 3 of the 5GENESIS project i.e. D4.12.

# REFERENCES


[1] "MONROE Project" [Online], https://www.monroe-project.eu/access-monroe-platform/
[2] 5GENESIS Deliverable 4.10, "The Surrey Platform", April 2019.
[3] 5GENESIS Deliverable 3.1, "Management and orchestration", October 2019, https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.1_v1.0.pdf
[4] Z. Bozakov, S. Mangiante, C. Hernandez Benet, A. Brunstrom, R. Santos, A. Kassler, and D. Buckley. A NEAT Framework for Enhanced End-Host Integration in SDN Environments. In IEEE Conference on Network Function Virtualization and Software Defined Networks (IEEE NFV-SDN 2017), Berlin, Germany, November, 2017.
[5] K.-J. Grinnemo, Z. Bozakov, A. Brunstrom, M. Isabel ¨ Sanchez Bueno, T. Dreibholz, K. Rikter Evensen, G. Fairhurst, A. Fosselie Hansen, D. Hayes, P. Hurtig, M. Rajiullah, T. Jones, D. Ros, T. Rozensztrauch, M. Tuxen, and E. Vyncke, "NEAT Deliverable D3.3 - Extended Transport System and Transparent Support of Non-NEAT Applications," Tech. Rep., 2017. [Online]. Available: http: //kau.diva-portal.org/smash/record.jsf?pid=diva2%3A1273851.
[6] 5GENESIS Deliverable 2.3, "Initial planning of tests and experimentation", February 2019, https://5genesis.eu/wp-content/uploads/2019/12/5GENESIS_D2.3_v1.0.pdf
[7] 5GENESIS Deliverable 4.4, "The Malaga Platform", March 2019.
[8] 5GENESIS Deliverable D2.2, "Initial overall facility design and specifications", December 2018, https://5genesis.eu/wp-content/uploads/2018/12/5GENESIS_D2.2_v1.0.pdf
[9] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; "Architecture enhancements for control and user plane separation of EPC nodes; Stage 2 (Release 15)", 3GPP TS 23.214 V15.5.0 (2018-12), https://www.3gpp.org/ftp/Specs/archive/23_series/23.214/23214-f50.zip.
[10] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects; "System Architecture for the 5G System; Stage 2 (Release 16)", 3GPP TS 23.501 V16.0.2 (2019-04), ftp.3gpp.org//Specs/archive/23_series/23.501/23501-g02.zip
[11] 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, " Interface between the Control Plane and the User Plane Nodes - Stage 3 (Release 15)," 3GPP TS 29.244 V15.5.0 (2019-03), ftp.3gpp.org//Specs/archive/29_series/29.244/
[12] OpenStack [Online], https://www.openstack.org
[13] OSM [Online], https://osm.etsi.org/
[14] 5GENESIS Deliverable 5.1, "System-level tests and verification", January 2020.

# ANNEX 1: 5GENESIS SURREY PLATFORM RADIO ACCESS TECHNOLOGIES

The radio access technologies to be deployed and integrated during Phase 3 in the Surrey Platform are the following:

## 3GPP access technologies

### REL gNodeB

REL gNB is comprised of two main units. A Distributed Radio Access Node (DRAN) and several Remote Radio Head (RRH) units. The DRAN and the RRH units are connected by an Ethernet ring. Figure 40 depicts the gNB general structure.

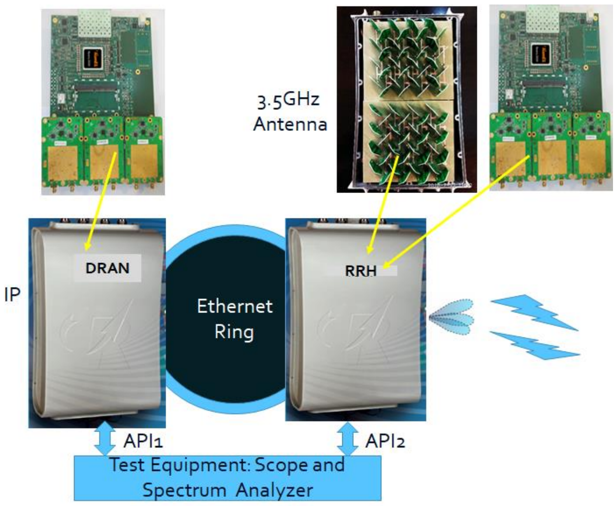



Figure 40: gNB General Structure

#### The DRAN General Description

The DRAN carries out the tasks of the 5G RAN L1 upper layer. These tasks mainly include the FEC encoding and decoding, beam management, distribution of the data to the RRH units over the 10 Gbps rings and the interface with the MAC and higher RAN layers. The DRAN is the physical layer central unit.

The DRAN processor implements algorithms of 5G physical (L1) layer protocols, including forward error correction (FEC) and detection for the data plane (LDPC) and for the control plane (Polar code), synchronization, rate matching (RM), layer mapping and digital beam switching.

The DRAN connects at one hand with the RAN protocol stack L2/L3 unit (including MAC and above (RLC, PDCP, SDAP, RRC) layers) and on its other hand it connects via the 10Gbps rings with the RRH units which are distributed at the coverage area.

In order to connect the DRAN with the Surrey Platform 5GC, there is a need for a third-party protocol stack with L2 and L3 layers as depicted in Figure 41.

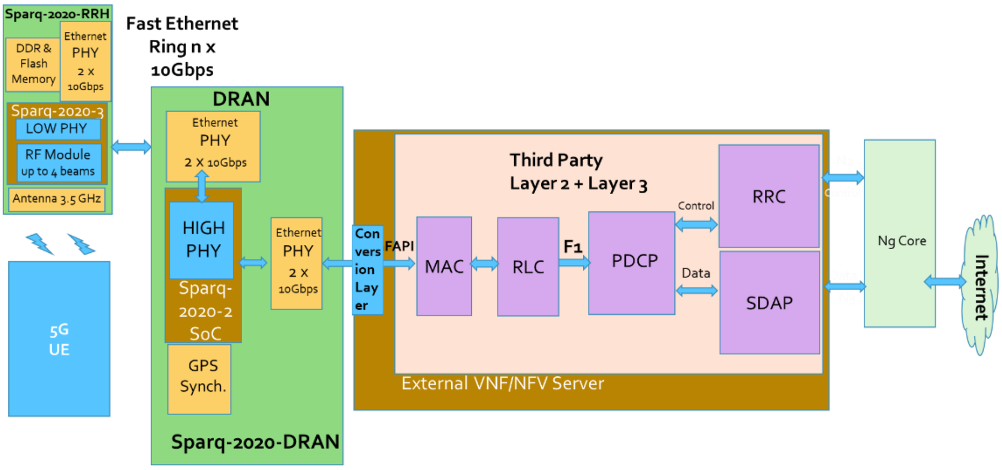



Figure 41: gNB Architecture

## The RRH units

RRH Units, these distributed units are responsible for the major part of the PHY layer processing. Processing includes mainly the data modulation/demodulation, precoding, IFFT, air interface resource mapping, and antenna/LED management. The RRH also includes the digital-to-analogue conversions, the 3.5 GHz RF module and the multi-beam antenna.

The RRH structure is depicted in Figure 42.

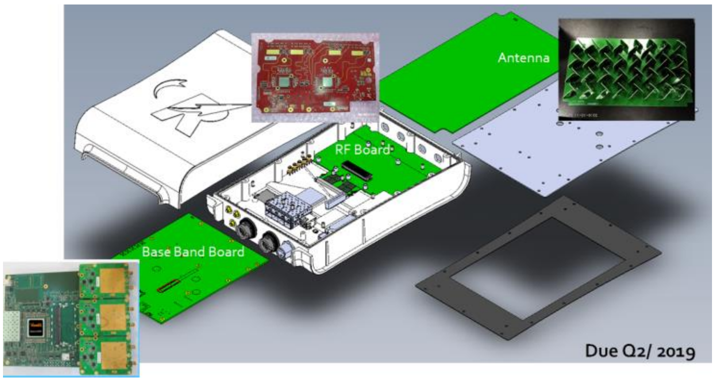



Figure 42: RRH Structure

The DRAN can drive several RRH units deployed at different locations.

The connection is carried out by a ring (Nx10Gbps) using a pair of F/O cables (or twice). One fiber will be used for DL, the other will be used for UL. Each fiber can drive up to 10 Gbps data.

# Non-3GPP access technologies

## WiFi

The WiFi deployment is based on a series of Ruckus access points (APs) interconnected to the Surrey Platform 5G Core (5GC) following the 3GPP Release 16 statements. These APs populate a specific SSID for the 5G non-3GPP network based on the requests of the slice manager. The management of these APs is handled from a WiFi Access Controller (AC) that exposes, via Rest API, all the needed operations to control the WiFi slices. As the recommendation dictates, all the traffic generated in the WLAN connections will be encapsulated and secured in a tunnel to the 5GC Non-3GPP Interworking Function (N3IWF) module through the corresponding interfaces as shown in Figure 43. The authentication will follow the same path toward the HSS in the 5GC. The WiFi network could also have a Radius-based Authentication, Authorization, and Accounting (AAA) server that would act as a proxy to the 5GC Radius services.

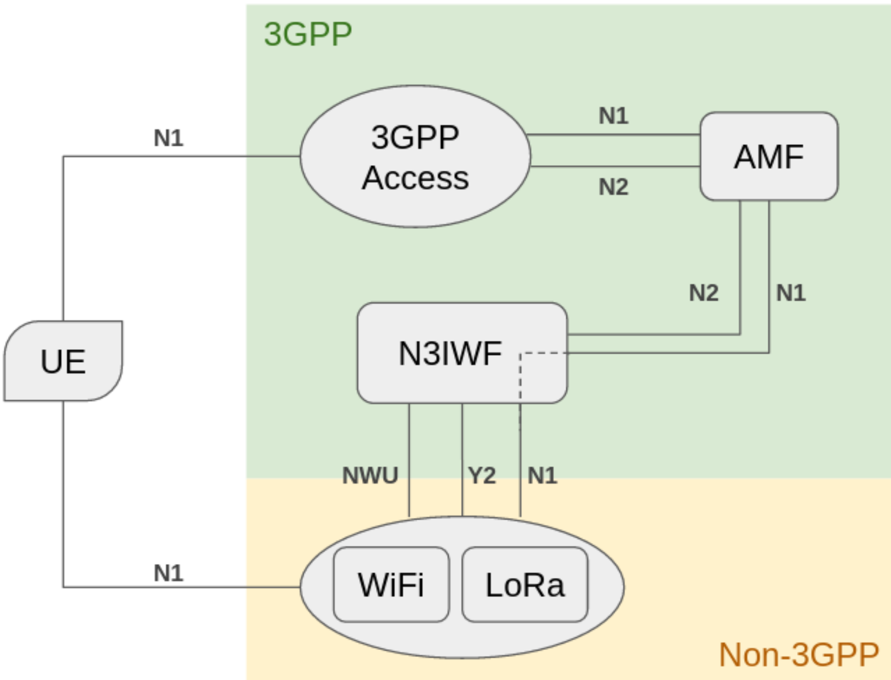



Figure 43: Non-roaming architecture for 5GCN with untrusted non-3GPP access

The WiFi APs deployed allow multiple slices and hundreds of devices connected simultaneously, which covers the necessity of deploying mMTC slices.

The exploitation of the unlicensed spectrum is partly covered with the WiFi deployment and will be complemented with other unlicensed spectrum network access technologies alike LoRa.

# User Equipment

## ECM 5G-NR UE

The ECM 5G-NR UE provided to the Surrey Platform is based on the OpenAirInterface RAN (OAI-RAN) solution. OAI RAN is an open-source software and hardware platform providing a standard-aligned implementation (3gpp Rel. 10/14) for the LTE UE and eNB. Currently, OAI is being extended to support 5G-NR UE and gNB [1], as per Rel.15 standards.

The OAI software is freely distributed by the OpenAirInterface Software Alliance (OSA) and it can be deployed using standard off-the-shelf Linux-based computing equipment (Intel x86 PC architecture) and standard RF equipment (e.g., National Instruments/Ettus USRP). In this context, OAI offers a flexible framework for experimentation with prototype 4G/5G implementations of the UE and base station components.

The OAI 5g-NR UE component will be interoperable with the gNB provided by REL to perform end-to-end experimentation and KPIs measurement collection. In this context, the protocol stack extensions for 5g-NR UE are becoming gradually available throughout the different phases of 5GENESIS, starting from the physical layer (Phase 1) and continuing with the rest of the RAN protocol stack (MAC, RLC, RRC, PDCP). The OAI UE can be launched and configured easily through a Command Line Interface (CLI). Based on this CLI, the UE can also be controlled remotely through external software.

Currently, OAI has introduced a new mode (noS1) to perform tests with IP traffic over 5gNR physical layer. As the interoperability with the REL gNB is in progress, this mode is currently used for native testing only between the OAI components (gNB and 5gUE). In this mode, there is no real connection of the gNB to any core network over the S1 interface and the exchanges that would normally take place between those two entities during the UE attachment to the network are bypassed/emulated. Moreover, the noS1 mode uses some of the functionality of the LTE RAN stack (Figure 44) to cover for the respective 5G-NR blocks which have not been integrated yet. This allows the traffic flow to/from the NR-PHY layer. Finally, by performing static pre-configuration for the data plane (i.e. data radio bearer, IP address configuration, etc.) IP traffic can be transferred between the gNB and UE entities. This mode has been validated so far with Downlink traffic.

The integration of the Uplink procedures (i.e., implementation of NR-PHY uplink channels and procedures, NR-MAC layer extensions and interfacing with NR-PHY) that will allow to enabling noS1 mode for uplink traffic as well are almost complete at the time being. Once they get fully integrated, testing with Uplink IP traffic will also be feasible.

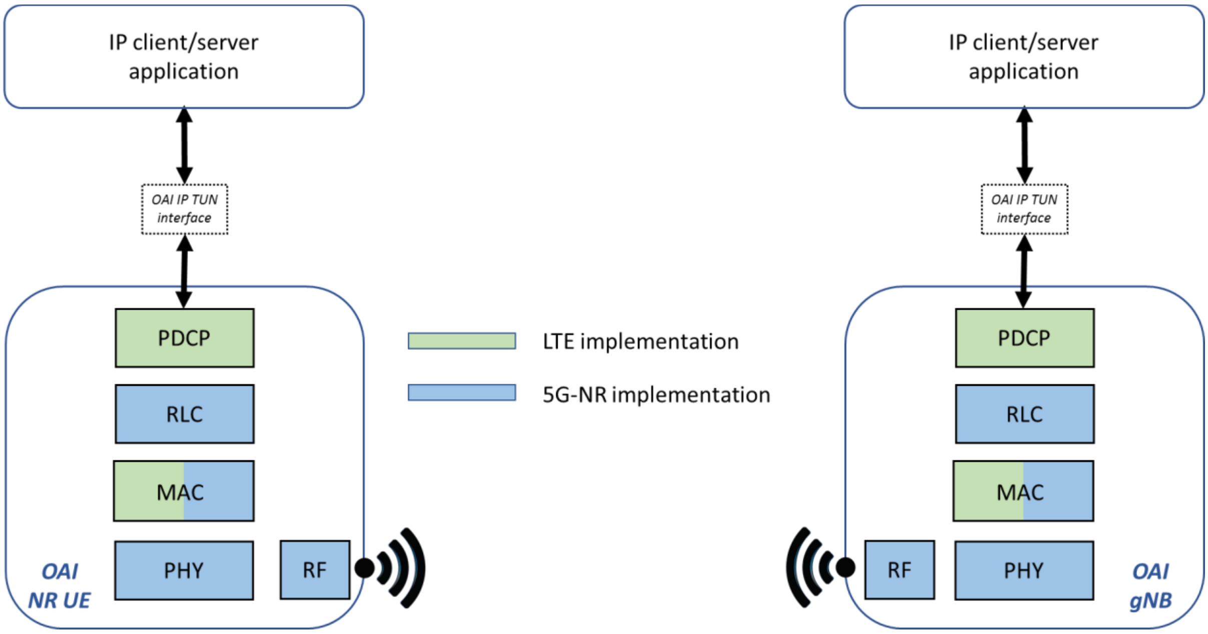



Figure 44: noS1 mode architecture supporting IP traffic flow

The hardware platform, provided by ECM, is going to use the ETTUS N300 boards together with a powerful Laptop with a Core i7-7900 8 core processor. A special adaptor to be able to connect the Thunderbolt 3 interface of the laptop with the 2x10Gbit Ethernet interface of the USRP will be used. An additional RF frontend and antenna will provide enough output power and amplification to operate in an outdoor environment. A picture of the UE is given in Figure 45.

Given the fact that the UE is a simple software program, it can be easily launched and accessed remotely through an SSH interface (provided that the laptop is connected to the Internet through an additional connection). In addition, performance measurements with tools like *iperf* or *ping* can be easily carried out locally or remotely.

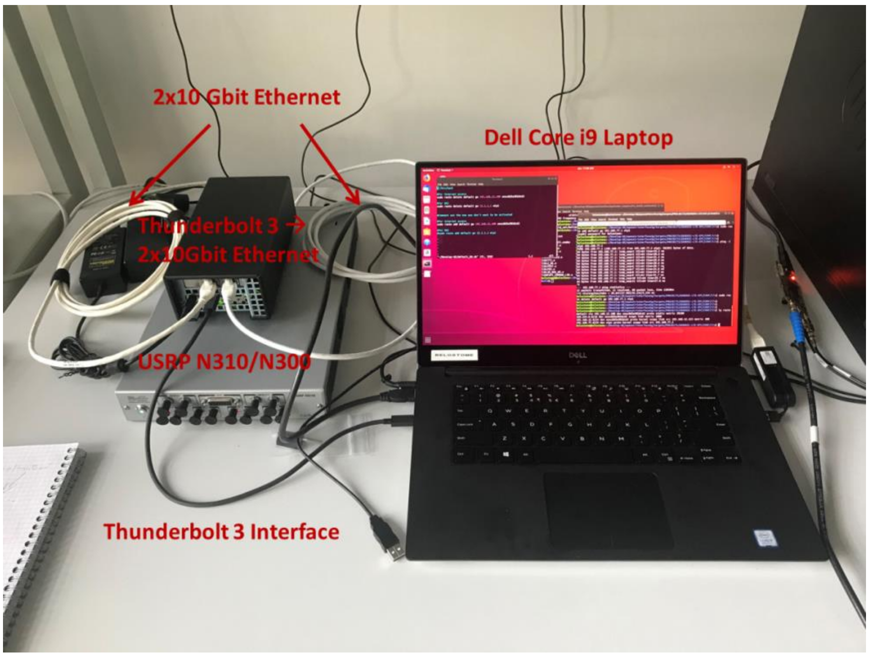



Figure 45:5G-NR UE Platform running OpenAirInterface

# ANNEX 2: DATA FORMAT USED IN SURREY SENSORS

This data is the superset of all pieces of information captured at sensors hosted by the various considered devices (see their descriptions below).

JSONdata =

```
{
 "entityID" : string,
 "entityType" :  string,
 "entityOwner" : string,
 "datatype": string,
"data" : {
        "pressure" : string,
        "temperature" : string,
        "light" : string,
        "humidity" : string,
        "gravity" : [string, string, string],
        "accelerometer" : [string, string, string],
        "compass" : [string, string, string]
        },
"metadata" : {
        "timestamp" : string,
        "latitude" : string,
        "longitude" ; string
        }
}
```

This structure conveys only Strings, which are converted from the Float sensor readings[15].

As a consequence, for a given type of device it might happen that some fields of that JavaScript Object Notation (JSON) structure are not assigned any value if it happens that they are NOT supported by any sensor (e.g., the metadata "longitude" and "latitude" are not assigned any value whenever the structure is produced for a Pycom <<Pysense>> board).

[15] [...] denotes an array.