

Received 7 March 2023, accepted 3 April 2023, date of publication 10 April 2023, date of current version 17 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3265894

## APPLIED RESEARCH

# NLP Powered Intent Based Network Management for Private 5G Networks

JOSEPH MCNAMARA<sup>1</sup>, DANIEL CAMPS-MUR<sup>2</sup>, MEYSAM GOODARZI<sup>3,6</sup>,  
HILARY FRANK<sup>4</sup>, (Graduate Student Member, IEEE), LORENA CHINCHILLA-ROMERO<sup>5</sup>,  
FERRÁN CAÑELLAS<sup>2</sup>, ADRIANA FERNÁNDEZ-FERNÁNDEZ<sup>2</sup>,  
AND SHUANGYI YAN<sup>4</sup>, (Member, IEEE)

<sup>1</sup>Ericsson LMI, Athlone, Westmeath, N37 PV44 Ireland

<sup>2</sup>i2CAT Foundation, 08034 Barcelona, Spain

<sup>3</sup>IHP—Leibniz-Institut für Innovative Mikroelektronik, 15236 Frankfurt (Oder), Germany

<sup>4</sup>Smart Internet Laboratory, University of Bristol, BS8 1QU Bristol, U.K.

<sup>5</sup>Department of Signal Theory, Telematics and Communications (DTSTC), University of Granada, 18014 Granada, Spain

<sup>6</sup>Department of Computer Science, Humboldt University of Berlin, 10117 Berlin, Germany

Corresponding author: Joseph Mcnamara (joseph.mcnamara@ericsson.com)

This work is supported by the European Commission's Horizon 2020 research and innovation program under grant agreement No 871428, 5G-CLARITY project.

**ABSTRACT** Intent driven networking holds the promise of simplifying network operations by allowing operators to use declarative, instead of imperative, interfaces. Adoption of this technology for 5G and beyond networks is however still in its infancy, where the required architectures, platforms, interfaces and algorithms are still being discussed. In this work, we present the design and implementation of a novel intent based platform for private 5G networks powered by a Natural Language Processing (NLP) interface. We demonstrate how our platform simplifies network operations in three relevant private network use cases, including: i) an intent based slice provisioning use case, ii) an intent based positioning use case, and iii) an intent based service deployment use case. Finally, all use cases are benchmarked in terms of intent provisioning time.

**INDEX TERMS** Artificial intelligence, intent based networking, machine learning, natural language processing, private 5G networks.

## I. INTRODUCTION

After almost ten years of research and standardization, the design of 5G networks as a distributed software and cloud/edge based infrastructure managed by per-domain orchestrators is well established. Softwarization has enabled virtualization of network functions that brought along flexibility for network services and deployments. For example, through the adoption of network slicing, services are created and tailored to vertical users. Thanks to all these key technical enablers, more heterogeneous, dynamic, expanded and distributed services have emerged in 5G network scenarios, increasing the complexity of network operations and hindering the management of such networks.

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Pu<sup>1</sup>.

The virtualization of network functions also allows for the deployment of private networks where the Virtual Network Functions (VNFs) can be exclusively used by a private entity such as an enterprise. Private networks, named as Non-Public Networks (NPNs) by 3rd Generation Partnership Project (3GPP), are categorized based on their deployment type as (i) Stand-alone NPNs (SNPNs), which do not rely on network functions provided by a public land mobile network (PLMN); and (ii) public network integrated NPNs (PNI-NPNs), where the private network deployment is supported by a PLMN [1]. In both deployment types, NPNs can be managed and controlled by a private network operator or entity owner. Private 5G networks are being increasingly demanded by several vertical domains, but their adoption is hindered by the lack of a labor force skilled in 5G within the private entities deploying these networks.

In this vein, simplifying operational complexity is the main goal of intent based networking, which originated around 2014 with the advent of the first Software-Defined Networking (SDN) open platforms [2]. Intent based networks are emerging as a key enabler to facilitate the management of private 5G networks where, through the use of intents, end-users with no technical network insights are able to issue network configurations and to operationally manage the private network. In this way, intent based networks offer a significant added value to the network owners and/or customers that have no expert knowledge on communication networks.

The main concept behind intent based networking is adopting declarative interfaces where network operators specify *what* they want to achieve, instead of specifying *how* to achieve it. In this regard, Machine Learning (ML) is seen as a key enabler to autonomously fill the missing context required to translate high level intents into low level network configurations [3]. A relevant example of the configuration complexity incurred in private 5G networks is provided in [4], where the authors introduce 5G-CLARITY, which is a novel architecture for 5G private networks integrating 5G NR, Wi-Fi and Li-Fi access networks. The 5G-CLARITY architecture is composed of three strata, namely an infrastructure stratum, a virtualised network and application function stratum and a management and orchestration stratum, and provides support for multi-connectivity, positioning and slice provisioning. Thus, deploying end-to-end network services in 5G-CLARITY requires the provisioning and configuration of both virtual and physical network functions (V/PNFs), imposing a steep learning curve for private network operators that are not familiar with 5G technologies.

The main contribution of this paper is twofold. First, we extend the original 5G-CLARITY architecture proposed in [4] by adding a new *intelligence stratum* that enables the intent driven operation of the private 5G network by means of ML models and a novel natural language processing (NLP) intent driven interface. Second, we describe a prototype implementation of our intent driven platform and demonstrate its generality applying it to three relevant private 5G network use cases, including: i) an intent based slice provisioning use case, ii) an intent based positioning use case, and iii) an intent based service deployment use case.

The rest of the paper is structured as follows. Section II provides an overview of the existing standards and related works, highlighting the novelties of this work compared to the revised ones. Section III describes our proposed intelligence stratum to provide NLP based intents, including an Intent Engine and Artificial Intelligence (AI) Engine. Section IV describes the application of our designed intent based interface to three relevant private 5G network use cases. Finally, Section V summarizes and concludes the paper.

## II. STATE OF THE ART

Over the last several years, intent-based networks (IBNs) have attracted the attention of academia researchers, industry, and standardization bodies due to their capabilities of

mitigating the complexity of the management tasks in emerging networks. This section overviews previous works related to IBNs mainly focusing on the existing standards and the academic research works.

### A. STANDARDIZATION

Several Standards Development Organizations (SDOs) (e.g., 3GPP, European Telecommunications Standards Institute (ETSI), TM Forum) are concentrating efforts on the standardization processes related to IBNs, although they are still under development. The standardization efforts of 3GPP are led by the SA5 group. In [5] the concepts of intent and intent driven management are introduced, where an intent driven management service allows their consumers to express intents to manage the 5G network and its services. The document also provides several intent driven management scenarios. In [6], 3GPP describes potential use cases (e.g., intent containing expectations for deploying a radio network in a specified area), their requirements, and the procedures to support intent driven network and service management. Another SDO contributing to this field is ETSI. In [7], the Zero-touch network and Service Management (ZSM) group starts describing the automation of network management. They concentrate on policy-driven automation, intent-based automation, and intent-based service orchestration. The 5G-CLARITY management stratum is an example implementation of an ETSI ZSM network [8], [9]. In [10], the Experiential Networked Intelligence (ENI) working group differentiates between declarative and imperative policies, and intent policies. Imperative policies specify how the network has to achieve a given task. In [10], these policies are stated as a triplet of event-condition-action, in which the condition determines the action to be taken to accomplish a desired state depending on the current state. On the other side, declarative policies just specify the desired state and let the system compute the action. ETSI defines an intent-based policy as a simplified declarative policy which declares what is wanted, but not how to do it. This intent definition approach is the one aligned with our proposal in this work.

TM Forum is another SDO working on facilitating business, services and network operations in 5G networks. The TM Forum Autonomous Networks Project defines a fully automated zero wait, zero touch, zero trouble innovative network for users and consumers of vertical industries. In several documents [11], [12], [13], [14], [15], [16] intent proposal, recommendations, interfaces and models specifications required to implement intents in autonomous networks can be found. Reference [11] describes a reference architecture for autonomous networks, including the three layers of business operations, service operations and resource operations, in order to provide guidance on the development of solution architectures for intelligent automation and autonomy within communications service provider (CSP) operations. IG1253 [12] is the main overview document of a set of documents describing the proposal for implementing

intents in autonomous networks and intent-driven operation. It defines an intent as a formal specification of all expectations including requirements, goals, and constraints given to a technical system. Our proposal of intent concept in this work also aligns with this definition since the intents allow the autonomous system to know the goals together with the requirements and constraints that issue an action to accomplish the mentioned goal. Additionally, it introduces the intent management function as an architectural building block for the intent-based operation, which will receive an intent fulfillment, control the execution of these actions, and report the progress. The document also discusses the role of natural and domain specific language in intent-driven autonomous networks, proposing an interpreter, which acts as a mediator between the natural or domain specific and the formal intent framework. Our intent approach aligns with two components detailed in the document. The intent engine, which we introduce in section III, with minor changes to the functionality template and context information may perform the role of both an intent management function or interpreter. The rest of documents [13], [14], [15], [16] are complementary documents deepening on other key aspects. For example, [13] introduces the models that determine how an intent must be expressed by defining the modelling concepts to be used when formulating and interpreting an intent. Reference [14] proposes extensions to the common model for expressing intents that might be relevant in specific domains or contexts. Reference [15] defines the details of the intent management function and the interface through which the intent objects are exchanged, negotiated, and managed as part of the life cycle management process. Reference [16] defines mechanisms of registration and discovery of capabilities and the scope of the responsibility associated with an intent manager.

## B. RELATED WORK

Application of intent based networking for 5G networks has been an intensive area of academic research in the last years. In [17], the authors propose an architecture for an intent driven orchestrator that can manage an end-to-end 5G system. The intent driven platform includes an ML function orchestrator (MLFO) component to manage the life cycle of ML functions required to support intents, and a centralized Data Lake component to expose network telemetry. In this work, we leverage the ideas of the MLFO and Data Lake components to design a novel intent based platform for private 5G networks.

Other authors have focused on the demonstration of intent based platforms operating over 5G research infrastructures. For example, [18] and [19] introduce intent based platform prototypes aiming at simplifying the lifecycle management of 5G network slices. The authors in [18] use a 5G stack based on Open Source MANO (OSM) [20] and FlexRAN [21], while in [19] the authors use a stack based on M-CORD [22]. In both cases, intents are provided through a graphical user interface (GUI), where the user indicates a slice type and

some complementary parameters that are translated by the intent engine into detailed configuration requests towards the Network Function Virtualization (NFV) and Radio Access Network (RAN) orchestrators. Unlike these previous works, our proposed intent based platform offers a natural language interface instead of a GUI.

The work in [23] surveys current intent based standardization and open-source related initiatives claiming that the field has not progressed much since 2015 in terms of new platforms or product adoption, mainly due to the lack of intent based interfaces that can effectively simplify network management operations. In this regard, the authors identify advances in NLP, thanks to AI/ML, as a potential catalyzer to adopt intent based interfaces. Regarding NLP interfaces for network management, the work in [24] presents a voice assisted network management platform, where the Alexa appliance is extended [25] to trigger establishment of end-to-end paths in a simulated SDN enabled carrier network. The authors demonstrate that voice-activated high level intents such as *set up a path from CityA to CityB* can be automatically translated into the required low level configurations. Similarly, in [26], the authors propose an intent-based network design based on natural language specifications to trigger network tasks in a network of several Open vSwitch switches emulated in Mininet. The authors check the functionality of intents like *create flow from 10.0.0.1 to 10.0.0.2 and block web traffic*. In future work we expect to see more investigations into the use and chaining of large language models for intent fulfillment. The authors of [27] have identified that complex operations may require the chaining of several models, each specialised for specific domains and as a result have investigated the user experience when authoring these chains. This work in the academic space coupled with the increased interest of large language models such as CHATGPT in popular culture may lead to similar approaches being investigated in the network and telecommunications domain.

To the best of our knowledge, no previous works are available demonstrating application of NLP based intent interfaces to management of 5G networks. Our goal in this work is to cover this gap. To that end, we put our focus on private 5G networks and we design a novel *intelligence stratum* that enables the intent driven operation of private 5G network by means of a novel NLP intent driven interface. Our designed *intelligence stratum* builds on the 5G-CLARITY architecture presented in [4].

## III. 5G-CLARITY INTELLIGENCE STRATUM DESIGN

Fig. 1 depicts the 5G-CLARITY private network architecture introduced in [4] augmented with a detailed *management and orchestration stratum* and with a novel *intelligence stratum* that enables the intent driven operation of the private network infrastructure.

As introduced in [4], the infrastructure stratum includes the physical network functions that compose the private 5G network infrastructure, i.e. wireless access points, radio heads,

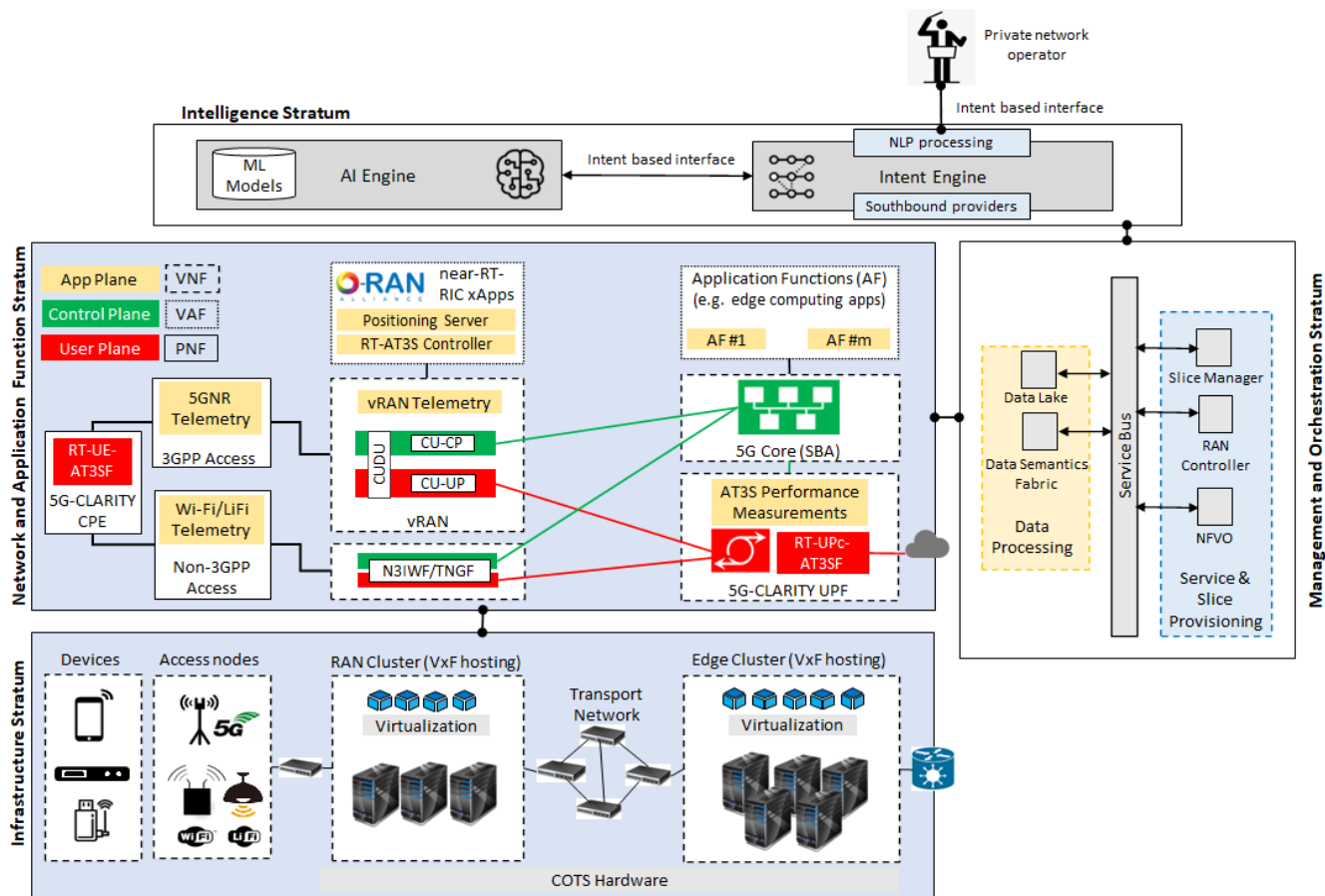


FIGURE 1. Proposed 5G-CLARITY intelligence stratum. Original strata proposed in [4] shown in blue. Novel strata proposed in this paper shown in white.

Ethernet switches, as well as compute clusters dedicated to host virtualized RAN and edge network functions. These virtualized RAN and edge network functions compose the network and application function stratum including, among others, core network, multi-connectivity, positioning, and O-RAN related network functions. Connectivity services over this infrastructure consist of radiating a private PLMN identifier (PLMNID)/Single – Network Slice Selection Assistance Information (S-NSSAI) pair with a dedicated user plane function (UPF) and an associated quality of service (QoS). Thus, provisioning connectivity services requires complex low level configurations on both the physical and virtual network functions.

The purpose of the Management and Orchestration stratum depicted in Fig. 1 is to enable an application programming interface (API) driven operation of the private network infrastructure. The management and orchestration stratum is composed of two main subsystems, the *service and slice provisioning subsystem* and the *data processing subsystem*.

The goal of the service and slice provisioning subsystem is to configure and provision the physical and virtual network functions required for a specific service. To this end, the service and slice provisioning subsystem includes three

management functions, namely: i) an *NFV Orchestrator (NFVO)*<sup>1</sup> used to manage the lifecycle of network services required for a slice, ii) a *RAN controller* in charge of configuring the wireless physical and virtual network functions, and iii) a *Slice Manager* function, which encodes the end-to-end business logic of a network slice, and offers a simplified north-bound interface to manage network slices. The interested reader is referred to [4] for a detailed description of the infrastructure slices supported by the 5G-CLARITY system.

The goal of the data processing subsystem is to collect and maintain network telemetry associated to the private 5G infrastructure, where telemetry is generated both by the physical and virtual functions in all segments of the network, i.e., RAN, transport, and core. Public cloud Data Lake services and data processing pipelines, such as the Semantic Data Aggregator available in [28], can be used to implement this stratum. The stored network telemetry will be queried by the intelligence stratum to generate low level configurations.

While enabling API driven control, the operational complexity exposed by the management and orchestration stratum is still significant. As a matter of example, let us take a closer look at the provisioning process of a network slice over the

<sup>1</sup>In our prototype implementation we use OpenSource MANO.

5G-CLARITY infrastructure, which involves two main steps. First, the infrastructure stratum resources allocated to the slice (i.e., access points and compute resources) need to be selected. Listing 1 depicts the API endpoint used to select cells and configure low level radio parameters, and Listing 2 the endpoint used to select compute resources, where the required compute, memory and storage resources are specified. The second step is to instantiate the network services composing the slice. Listing 3 illustrates the API endpoint used to configure a radio service, which includes QoS, service and user related low level configurations. We can see how the provided endpoints are far from the simplicity envisioned by our intent based interface, where a user should be able to provision a network slice using natural language through a simple intent such as *I want to provision a slice*. Delivering this abstraction is the goal of the intelligence stratum depicted in Fig. 1.

The intelligence stratum is composed of two functional elements, the *Intent Engine* and the *AI Engine* (c.f. Fig. 1). The goal of the Intent Engine is to offer an NLP based interface to the end user, which is then translated into low-level configurations towards the functions of the appropriate stratum, e.g. the API calls of the Slice Manager, illustrated in Listings 1, 2 and 3. The goal of the AI Engine is to host and manage the lifecycle of ML models, which will support the Intent Engine into generating the required low level context. The Intent Engine and AI Engine work together to make informed complex changes in the network, which are dynamically communicated to the responsible components in both the Network and Application Function Stratum and the Management and Orchestration Stratum. Next, we describe the detailed design of the Intent Engine and AI Engine components, and the intent related workflows that describe the low level actions performed in our platform when high level intents are issued by an user.

### A. INTENT ENGINE

The Intent Engine is built upon Adaptive Policy EXecution (APEX), a state machine approach to policy execution that incorporates a context functionality into the decision making mechanism. APEX was initially published in [29] and was later adopted as a Policy Decision Point (PDP) in the Open Network Automation Platform.<sup>2</sup> Our previous work in [30] provides the core NLP-based interpretation mechanism for the Intent Engine.

Fig. 2 shows the interpretation and matching process utilised in this work. The core components of the mechanism are the use of Context, Interpreter and Intent Matching. Context contains information of available interfaces and functionality, Interpreter manages the translation and building of API calls to fulfil the intent request, and Intent Matching provides an evaluation of the intent request when compared to available functions and strategies.

<sup>2</sup><https://docs.onap.org/projects/onap-policy-parent/en/latest/apex/APEX-Introduction.html>

```
{
  "name": "slicing-bosch-rc",
  "user_id": "6318bb4c0d798400090f95d5",
  "chunk_topology": {
    "selectedPhys": [
      {
        "id": "80e73ad2-e0ff-4c3b-b4e5-3a19e7491e1c",
        "name": "Cell-1",
        "mac": null,
        "friendlyName": null,
        "type": "AMARISOFT_CELL",
        "config": {
          "selectedRFPport": 0,
          "cellID": 1,
          "physicalCellId": 1,
          "type": "5G-TDD",
          "earfcnDl": 660000,
          "tac": 67,
          "rootSeqIndex": 1,
          "bandwidth": 40,
          "tddConfig": 2,
          "cellGain": 0,
          "cellNSARelationship": []
        }
      },
      {
        "id": "5e2352d3-9fd3-4953-a9ed-d89217233a63",
        "name": "en01",
        "mac": null,
        "friendlyName": null,
        "type": "WIRED_ROOT",
        "config": null
      }
    ]
  }
}
```

LISTING 1. Radio chunk.

```
{
  "name": "slice1-bosch-cc",
  "user_id": "6318bb4c0d798400090f95d5",
  "username": "slice1-bosch-cc",
  "password": "slice1-bosch-cc",
  "description": "slice1-bosch-cc",
  "compute_id": "6318bb980d7984000c48d58f",
  "requirements": {
    "ram": {
      "required": 12,
      "units": "GB"
    },
    "cpus": {
      "required": 6
    },
    "storage": {
      "required": 120,
      "units": "GB"
    }
  }
}
```

LISTING 2. Compute chunk.

These components of the mechanism will be described in more detail in the following paragraphs. The mechanism was then adapted to expand the action and interface generation features to incorporate the AI Engine, ML models and orchestration type components. This allowed the Intent Engine to communicate with slice provisioning systems, active ML

```

{
  "name": "slicel-bosch-rs",
  "user_id": "6318bb4c0d798400090f95d5",
  "slic3_id": "635a41290d79840009478235",
  "wifi_config": {
    "ssid": "MYSSID",
    "encryption": "WPA",
    "password": "minimumlengthpassword",
    "airtimeWeight": 100,
    "contentionWindowIncrease": 5
  },
  "cellular_config": {
    "plmn_id": "00103",
    "core_vendor": "open5gs",
    "enable_day2": true,
    "apn": "apn_test",
    "apn_qci": 9,
    "apn_dl_ambr": 10000000,
    "apn_ul_ambr": 10000000,
    "ue_data": [
      {
        "imsi": "001030056734816",
        "dl_ambr": 10000000,
        "ul_ambr": 10000000
      }
    ]
  },
  "user_profile": {
    "opc": "000102030405060708090A0B0C0D0E0F",
    "k": "00112233445566778899AABBCCDDEEFF",
    "amf": "8000"
  }
}

```

LISTING 3. Example radio service activation API endpoint.

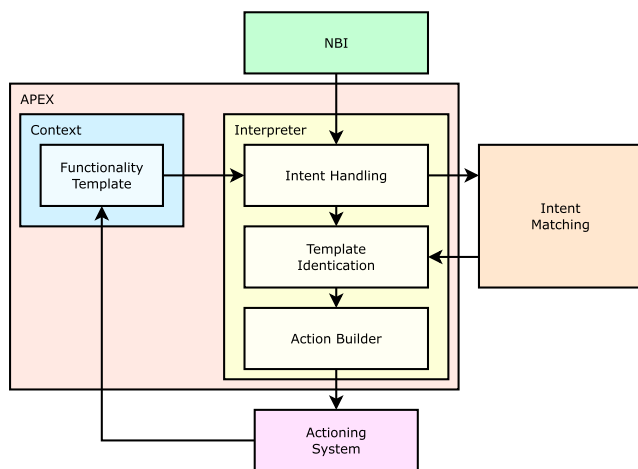


FIGURE 2. Intent interpretation and matching mechanism.

models running in the AI Engine and production-quality Network Service Orchestrators (NSO). This communication will be shown in Section IV.

Next, we describe the key design principles of our Intent Engine, namely: i) the *Intent Design*, describing how intents are built, ii) the *Intent Matching Engine*, describing how intents are matched to specific API endpoints, and iii) the *South-bound provider integration*, describing how different management functions can be integrated with the Intent Engine.

### 1) INTENT DESIGN

An Intent message has two fields: *request* and *parameters*. The request field contains an intent in the form of an English sentence. This is used in the matching process to compare the users request with operations available to the Intent Engine. The parameters field contains information the user would like to inject into the action building process. This is required information that is not provided through component defaults or through a supplementary ML model in the AI Engine.

The request and parameters fields can be seen in Listing 4 in the form of an example, namely a Slice Provisioning Intent. In the example, the request details the creation of a slice through an ML model in the AI Engine. As this is a complex process containing several sequenced executions the ML model populates many of the sequenced operations, however some parameters cannot be queried or generated. These parameters are detailed in the example such as *user-list*, *location* and *technology*. A detailed explanation of the intent based slice provisioning use case is provided in Section IV-A.

```

{
  "intent": {
    "request": "Create a slice",
    "parameters": {
      "name": "nova",
      "user-list": [
        {
          "imsi": "001035432100005"
        }
      ],
      "location": {
        "latitude": 0.0,
        "longitude": 0.0
      },
      "technology": [
        "AMARISOFT_CELL",
        "SUB6_ACCESS"
      ]
    }
  }
}

```

Listing 4. Slice provisioning intent example.

### 2) INTENT MATCHING ENGINE

The Intent Engine adopts an NLP-based matching process to correlate intents with executable operations available in the moment. This is achieved through a text distancing algorithm trained on Wikipedia data, which scores the intent request against a range of operation descriptions. The operation descriptions are stored in a model referred to as the Functionality Template (see Fig. 2). The user provides the Functionality Template to the Intent Engine through a registration process, informing the Intent Engine how and where to communicate with the component. This template follows OpenAPI Specification allowing it to be processed in the intent matching and action generation stage of the execution. Information from this model also informs the dynamic building of the interface to allow communication between

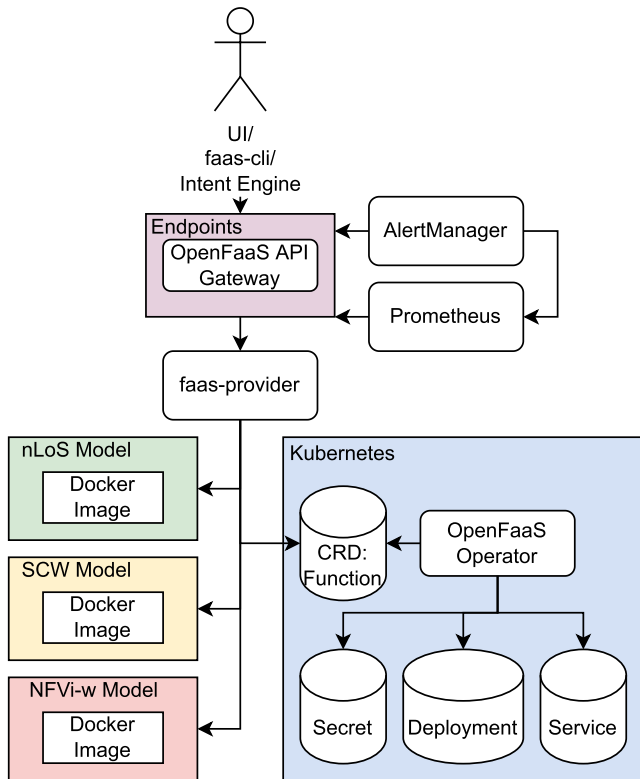


FIGURE 3. High level overview of OpenFaaS concepts.

the Intent Engine and the target component as a result of the processed intent. A detailed description and evaluation of our matching engine can be found in [30].

3) SOUTH-BOUND PROVIDER INTEGRATION

Informed through the Functionality Template, the Intent Engine dynamically builds the interfaces for the south-bound communications. Through the use cases described in Section IV we show the Intent Engine communicating several requests to three components operating in two independent strata. These interfaces are built at execution time to ensure the most up to date information is used during the action building and interface building stage of execution. As a result, changes or re-configurations of components do not require a hard restart of the Intent Engine. Instead, the reissuing of an updated Functionality Template would restore the communication.

B. AI ENGINE

The AI Engine is built upon the open-source Function-as-a-Service (FaaS) platform OpenFaaS. The OpenFaaS platform provides a flexible and lightweight toolkit which can be deployed through Docker, built using a range of languages and scales appropriate to usage.

The conceptual diagram shown in Fig. 3 gives a high-level overview of OpenFaaS concepts, with endpoints for the user, monitoring, Kubernetes as the FaaS provider and two

example functions. OpenFaaS exposes REST endpoints to the user accessible through CLI client, GUI or REST request. The central FaaS provider running on Kubernetes provides the infrastructure for deployed functions. In the AI Engine these functions represent the ML models accessible through the Intent Engine. Examples of these models will be provided in Section IV.

OpenFaaS provides a command line interface tool referred to as *faas-cli*. This tool allows straightforward control of key OpenFaaS operations for the creation and deployment of ML models onto the AI Engine. OpenFaaS was customised for the 5G-CLARITY project through an embedding of the Prometheus monitoring and Grafana visualisation tools. These tool sets are incorporated into the model creation process through the creation of templates for 5G-CLARITY AI Engine models. Models created with this template embed the previous tool functionalities. This template uses a python3 language and is adapted with a Prometheus monitoring library to expose internal model metrics to the telemetry subsystem of the 5G-CLARITY system (c.f. Fig. 1). This tool also initiates the building of models, pushing of models to the Docker registry and deployment of models onto the running AI Engine. Execution of these models is triggered through the Intent Engine, as will be seen through the use cases described in Section IV. The AI Engine supports models for the coordination and sequencing of complex network management and configuration tasks. In the event of a flag or identified conflict, log information is provided to the user detailing the events leading up to the issue.

C. INTENT PROCESSING WORKFLOWS

Fig. 4 depicts four types of intent related workflows, describing how user issued high level intents translate into low level network actions in our platform.

Starting with case (a), we depict a situation where the Intent Engine is able to directly match a user issued intent to a concrete API endpoint exposed by the service and slice provisioning subsystem. We note that while possible this will not be common in practice, because user intents will often translate into complex interactions with the service and slice provisioning subsystem that cannot be directly inferred by the intent matching engine. The alternative to handle complex user interactions is depicted in case (b), where the Intent Engine will match a high level intent to a specific ML model that will be instantiated in the AI engine to satisfy the intent. The ML model will encode the low level interactions required to implement the requested intent.

Cases (c) and (d) depict how instantiated ML models interact with the network, which is also by means of issuing intents. The intents issued by the ML models will however contain more information than user issued intents, in the form of additional parameters, such that the Intent Engine can properly match them to the corresponding southbound providers. ML models are expected to generate two types of intents. First, as depicted in case (c), data related intents

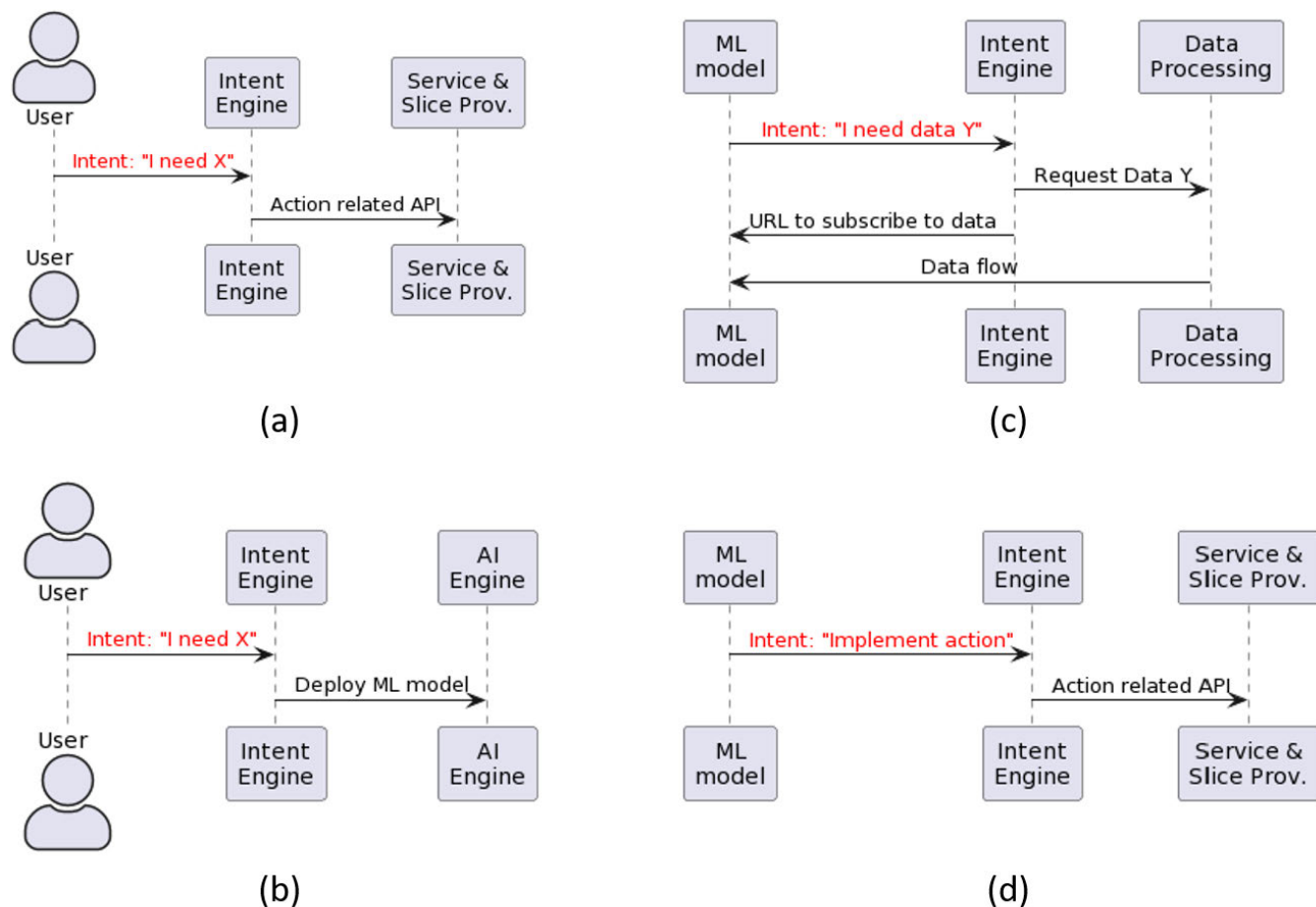


FIGURE 4. Intent related workflows.

are issued towards the data processing subsystem to subscribe to relevant telemetry including access network and user telemetry. Second, as depicted in case (d), an action related intent is issued towards the service and slice provisioning subsystem to enforce an action on the private 5G network. In Section IV, the generic workflows of Fig. 4 are applied to three different use cases.

Finally, it is worth highlighting that the entity triggering the intent in the cases depicted in Fig. 4 could be located within the private venue, or could be an external entity offering managed services, e.g. a mobile network operator (MNO) or system integrator.

#### IV. DEMONSTRATION ON SELECTED USE CASES

We present in this section three use cases enabled by our intent driven platform that are relevant for private 5G network operators. The first use case, described in Section IV-A, demonstrates the provisioning of network slices using intents. The second use case, described in Section IV-B, illustrates the deployment of an ML model that supports high precision positioning services in indoor environments. The third use case, described in Section IV-C, illustrates the intent based

deployment of an application function connected to the private 5G infrastructure.

All use cases are evaluated in terms of their required execution time, which we define as the time required from the moment the user inserts the intent into the intelligence stratum until the corresponding actions are enforced in the network.

#### A. INTENT BASED SLICE PROVISIONING

##### 1) MOTIVATION

As discussed in [4], network slicing over private 5G networks can be used to support multiple services over a common infrastructure. For example, in a factory environment, different production lines can be connected to dedicated network slices to ensure isolation and provide a dedicated QoS. The 5G-CLARITY system supports infrastructure based network slices that are composed of the following elements:

- i. A *compute chunk*, defining the compute infrastructure that will host the VNFs allocated to that slice. The compute chunk specifies the compute, memory and storage resources allocated to that slice.



- ii. A *radio chunk*, specifying the 5G NR cells and WiFi and LiFi access points from which devices can access the provisioned slice.
- iii. A *core network service*, deploying the 5G Core VNFs, and configuring them with the appropriate PLMNID/S-NSSAI identifiers associated to the slice, with the initial set of User Equipments (UEs) that are allowed to connect to the slice, as well as the QoS parameters allocated to that slice.
- iv. A *radio service* adding the PLMNID/S-NSSAI identifier and service set identifier (SSID) to the cells and APs selected in radio chunk. The radio service also specifies the radio resources (physical resource blocks (PRBs) or airtime) that are allocated to the slice.

A detailed description of the 5G-CLARITY approach to infrastructure slicing can be found in [4].

## 2) INTELLIGENCE STRATUM DESIGN

Despite the complexity involved in the slice provisioning process described in the previous subsection, our intent based platform enables private network operators to provision slices through a natural language interface. The key to our design is a set of ML models deployed in the AI Engine of the intelligence stratum (c.f. Section III), illustrated in Fig. 5. These models allow to translate a high level intent to the required low level configuration, including:

- i. The *Slice Creation Workflow* (SCW) model that encodes the multiple low level interactions with the Slice Manager function required to provision a network slice (c.f. Listings 1, 2 and 3). The SCW model offers a single endpoint that can be identified by the intent matching component of the Intent Engine (see Fig. 5), and is in charge of orchestrating the inputs provided by the other models.
- ii. The *Radio Node Selection* (RNS) model that is in charge of selecting the 5G NR cells or WiFi and LiFi access points from which the slice needs to be served. An ML-based radio node selection may consider current and predicted load demands around the area where the slice needs to be served.
- iii. The *Compute Requirements* (CR) model that is in charge of defining the computational requirements (CPU, memory and storage) allocated to the slice. An ML-based compute node selection may consider predictions on CPU utilization or energy saving constraints.
- iv. The *Slice QoS* (SQS) model that is in charge of defining the QoS Class identifier (QCI) and uplink (UL)/downlink (DL) Aggregate Maximum Bit Rate (AMBR) parameters to be applied to the devices connecting to the slice. An ML-based QoS selection may consider a Service Level Agreement (SLA) definition at slice level, e.g. as the authors proposed in [31].

Note that we focus on the architectural definition of the ML models needed to support an intent based slice provisioning,

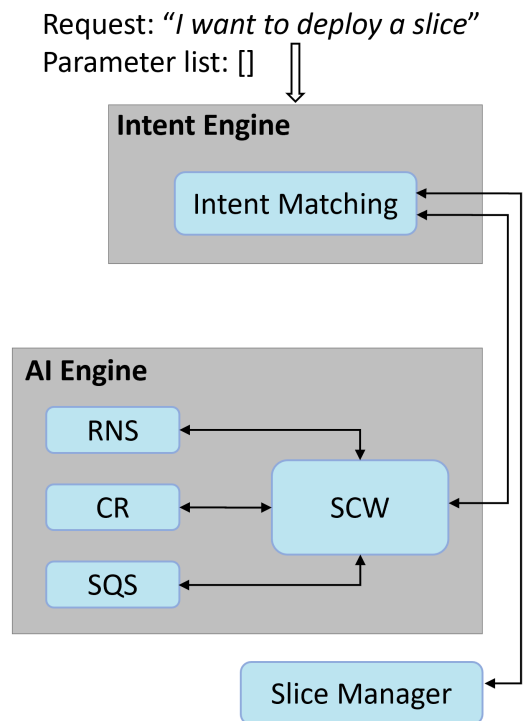


FIGURE 5. ML models supporting intent based slicing.

and leave a concrete implementation of these models for future work. Fig. 6 describes the detailed interactions between the Intent Engine, the SCW, RNS, CR and SQS modules in the AI Engine, and the Slice Manager function to enable intent based slice provisioning. The generated intents are shown in red in Fig. 6, and we note that intents are generated both by the human operator and by the ML models (case (c) in Fig. 4).

## 3) INTENT DEFINITION

Table 1 describes four types of slice related intents supported by our system. The different intent types differ in the type of provided parameters, which result in the involvement of different ML models in the slice provisioning process.

Starting from the *default* intent type at the top of Table 1, we can see that no parameters other than the desired user list are provided. In this case, the user is not providing any useful context information that can be used by the RNS, CR and SQS models to optimize the slice provisioning process. Thus, these models provide default slice parameters. In the second intent type radio information is provided as context in the form of a desired geographical area where the slice needs to be provisioned, or the desired access technologies that should advertise the slice (e.g. 5G NR, Wi-Fi or Li-Fi). In this case, the RNS model uses the provided parameters to select the target cells and APs. Fig. 6 depicts the interactions generated by the RNS model. In the third type of intent in Table 1 compute related information is provided as a list of target network service identifiers (NSIDs) that will make use of the

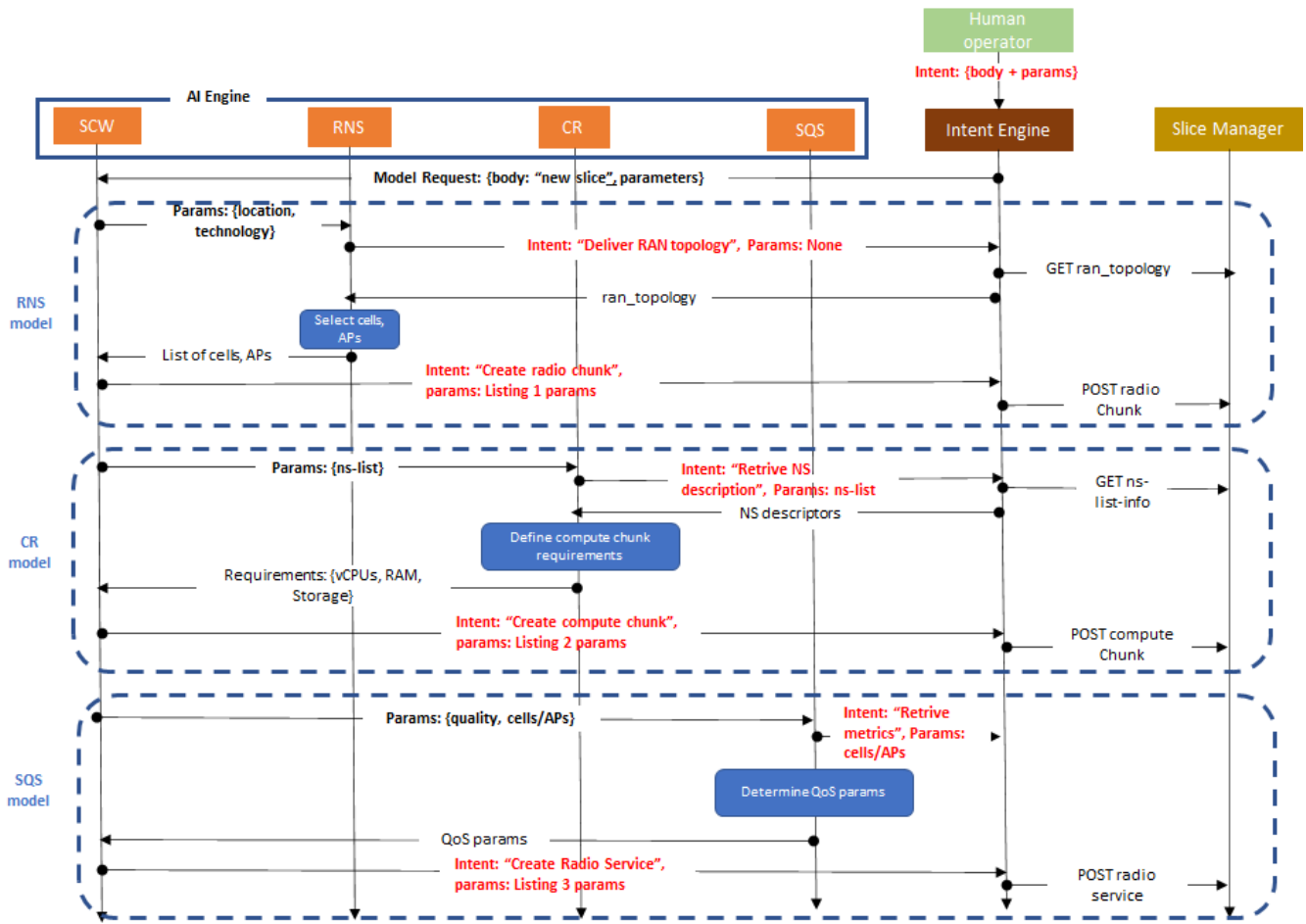


FIGURE 6. Intent based slicing workflow. The interactions triggered by each model in the AI Engine are shown separately.

deployed slice.<sup>3</sup> The CR module uses the list of NSIDs to dimension the CPUs, memory and storage requirements of the slice, through the interactions depicted in Fig. 6. Finally, the fourth type of intent in Table 1 includes a parameter providing a high level quality indicator. The SQS model makes use of this parameter to derive the 5G Core QoS parameters to be used in the slice. The interactions triggered by the SQS model are depicted in Fig. 6.

**B. INTENT BASED NLoS IDENTIFICATION**

1) MOTIVATION

Accurate positioning is a key service to be delivered by private 5G networks, especially in indoor environments such as factories where there is no Global Navigation Satellite System (GNSS) coverage. To this end, 3GPP has specified (in Release 16) positioning extensions that include new Positioning Reference Signals (PRS) in the physical layer, as well as a Location Management Function (LMF) in the 5G Core.

<sup>3</sup>NSIDs refer to the identifiers of the network services onboarded on the NFV orchestrator of the service and slice provisioning subsystem. In use case 3 we discuss how these NSIDs could be derived from high level user information.

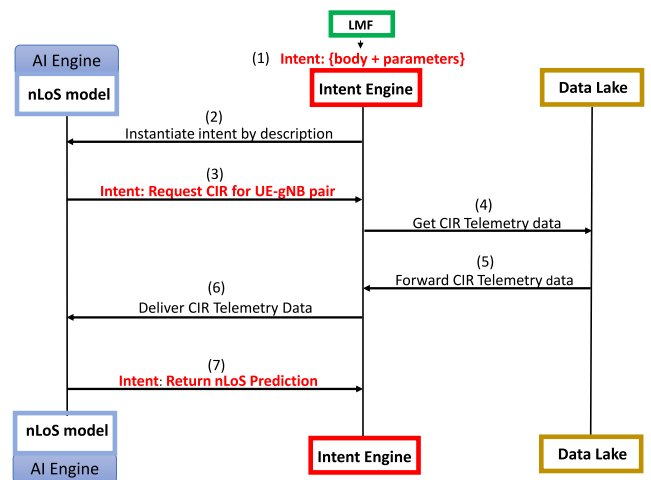


FIGURE 7. Sequence diagram for NLoS use case.

A UE measures PRS signals from different base stations resulting in power, angular momentum, or time measurements. These measurements are then delivered to the LMF

**TABLE 1.** Slicing related intents and role of RNS, CR and SQS models in AI engine.

<b>Intent type:</b>	Default
<b>Intent body:</b>	<i>I want to create a slice</i>
<b>Parameters:</b>	<i>Name: my-slice, user-list: {IMSI-list}</i>
<b>Model Description:</b>	<i>RNS model:</i> Applies slice to all available cells and APs. <i>CR model:</i> Applies default CPU, memory and storage requirements. <i>SQS model:</i> Applies default CQI and UL/DL AMBR.
<b>Intent type:</b>	Adding radio information
<b>Intent body:</b>	<i>I want to create a slice</i>
<b>Parameters:</b>	<i>Name: my-slice, user-list: {IMSI-list}, location: Coordinates, access technology: {wifi,5g,lifi,5g+wifi,5g+lifi,5g+wifi+lifi}</i>
<b>Model Description:</b>	<i>RNS model:</i> Filters cells and APs according to location coordinates. Filters according to access technology. Infers selection of exact cells and APs according to predicted load demands. <i>CR model:</i> Applies default CPU, memory and storage requirements. <i>SQS model:</i> Applies default CQI and UL/DL AMBR.
<b>Intent type:</b>	Adding compute information
<b>Intent body:</b>	<i>I want to create a slice</i>
<b>Parameters:</b>	<i>Name: my-slice, user-list: {IMSI-list}, access technology: {wifi,5g,lifi,5g+wifi,5g+lifi,5g+wifi+lifi}, services: {ns-id1, ns-id2}</i>
<b>Model Description:</b>	<i>RNS model:</i> Filters cells and APs according to access technology. <i>CR model:</i> Infers required compute resources from service descriptors stored in NFVO. <i>SQS model:</i> Applies default CQI and UL/DL AMBR.
<b>Intent type:</b>	Adding QoS information
<b>Intent body:</b>	<i>I want to create a slice</i>
<b>Parameters:</b>	<i>Name: my-slice, user-list: {IMSI-list}, access technology: {wifi,5g,lifi,5g+wifi,5g+lifi,5g+wifi+lifi}, quality: {gold, silver, bronze}</i>
<b>Model Description:</b>	<i>RNS model:</i> Filters cells and APs according to access technology. <i>CR model:</i> Applies default CPU, memory and storage requirements. <i>SQS model:</i> Infers exact QCI and UL/DL AMBR parameters from quality parameter, depending on demand and requirements from other instantiated slices.

that runs a localization algorithm to estimate the position of the UE [32]. In the case of time measurements, it is crucial for the LMF algorithms to discern if the measurement between the UE and a gNB has been performed under Line of Sight (LoS) conditions or not, as this heavily impacts the accuracy of the positioning measurement. Thus, in this section we describe a use case where the LMF can generate an intent towards the 5G-CLARITY intelligence stratum to verify if the communication link between a given UE and gNB pair is LoS or Non-LoS (NLoS).

## 2) INTELLIGENCE STRATUM DESIGN

A single ML model referred to as *NLoS model* instantiated in the AI Engine is sufficient to implement this use case.

**TABLE 2.** NLoS identification intent.

<b>Intent type:</b>	Default
<b>Intent body:</b>	<i>Provide the link condition</i>
<b>Parameters:</b>	List of (gNB, UE) pairs
<b>Return value:</b>	"NLoS/LoS" and confidence level
<b>Model Description:</b>	<i>NLoS model:</i> Retrieves CIR telemetry about the provided UE and gNB pair. Feeds CIR telemetry to a neural network that determines if the link condition is LoS or NLoS.

Fig. 7 depicts the implementation scheme of our intent-based NLoS identification. The process starts when the LMF generates an intent towards the Intent Engine to validate if a certain gNB is in the LoS of a UE (step 1). The Intent Engine matches the received intent to an endpoint offered by the NLoS model (step 2), which has been previously registered as a south-bound provider of the Intent Engine. Subsequently, the NLoS model generates a new intent towards the Intent Engine asking for the Channel Impulse Response (CIR) data corresponding to the requested UE and gNB pair (step 3). The Intent Engine matches the received intent to an end-point offered by the data lake, which has also been registered as a south-bound provider of the Intent Engine, resulting in the delivery of the CIR data to the NLoS model (steps 4 and 5). Finally, the link condition is computed by the NLoS model, returned to the Intent Engine, which then forwards it to the LMF (steps 6 and 7).

## 3) INTENT DEFINITION

Table 2 describes the intent related to the NLoS identification. The *intent body* is any high level sentence containing any of the phrases *link condition*, *NLoS*, and *LoS*. The *parameters* are the UE and gNB identifiers for which the link condition needs to be returned. Finally, the *description* explains the steps of intent processing. In particular, the CIRs are firstly retrieved from the data lake. Next, they are fed into the DNN pretrained on the data from the environment. The DNN consists of 2 hidden layers with  $N$  neurons in each layer where  $N$  denotes the length of the discrete CIR. Such a DNN has been previously proposed in [33] and successfully employed in [34]. Lastly, the results are returned to the operator by denoting the link condition, i.e., LoS or NLoS, and their corresponding probabilities. For example, a response such as {Link condition: "NLoS," Probability: 0.76} means that the link condition is NLoS with the probability of 0.76. The LMF will then use this information when computing the position estimate for the UE, e.g., by dropping the measurement corresponding to that specific link.

## C. INTENT BASED NETWORK SERVICE PROVISIONING

### 1) MOTIVATION

Private 5G operators need to connect their enterprise services to the deployed 5G network slices automatically without a complex configuration workflow. To this end, enterprise services, which are virtualized and represented through a

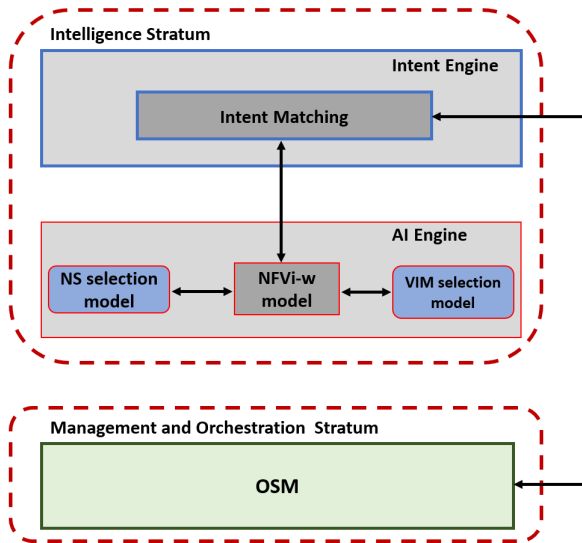


FIGURE 8. Intent based ML model for NFVaaS architecture.

network service descriptor (NSD) that is onboarded to the NFVO, can be instantiated over the appropriate compute resources under requests. However, direct operation of the NFVO is still too cumbersome for a private network operator that is not skilled in 5G. Therefore, there is a need to enable intent based deployment of enterprise services over the compute resources available in the private 5G network, e.g. the edge cluster in Fig. 1.

In this section we describe an intent based NFV service deployment use case based on the 5G-CLARITY architecture, where OSM is used as NFVO [35].

2) INTELLIGENCE STRATUM DESIGN

Like in the design of the intent based slice provisioning use case, several ML models need to be instantiated in the AI engine to support this use case, namely:

- The *NFVi workflow (NFVi-w)* model, which offers a single endpoint that can be matched by the Intent Engine upon receiving an intent including the words *network service*. In addition, this model receives in the intent parameters the context information required to resolve the exact NSD identifier that needs to be provisioned.
- The *NS selection* model is in charge of resolving the required NSD identifier from the context information provided in the intent. ML models may assist in the operation of the NS selection model by autonomously choosing application parameters, e.g. resolution in the case of video, as a function of the current state of the network.
- The *VIM selection* model that is in charge of selecting the exact compute resources, i.e. Virtualization Infrastructure Manager (VIM) registered to the NFVO where the network service needs to be instantiated.

TABLE 3. Intent based service provisioning.

<b>Intent type:</b>	Default
<b>Intent body:</b>	Deploy a new Network Service
<b>Parameters:</b>	VNF related parameters, e.g. for video, Resolution: {Low, Medium, High}, Audio: {true, false}
<b>Model Description:</b>	NS selection: Based on the provided parameters identifies the concrete network service that needs to be deployed. VIM selection: Deploys the network service on a default compute resource.
<b>Intent type:</b>	Adding placement context
<b>Intent body:</b>	Deploy a new Network Service
<b>Parameters:</b>	Placement context, e.g. preferred edge location, VNF related parameters, e.g. for video, Resolution: {Low, Medium, High}, Audio: {true, false}
<b>Model Description:</b>	NS selection: Based on the provided parameters identifies the concrete network service that needs to be deployed. VIM selection: Uses placement context to influence compute resources by selecting one of the VIM accounts registered in the NFVO.

Fig. 8 depicts the architecture of the intelligence stratum required to enable this use case, with the NFVi-w, the NS selection and the VIM selection models in the AI engine. Fig. 9 illustrates the sequence of interactions between the Intent Engine, the models in the AI Engine and OSM described above. These interactions are further detailed in next paragraph.

3) INTENT DEFINITION

In Table 3 we describe two types of Intents that can be supported by our platform, which differ in the context information provided in the parameters of the intent.

In the default type of intent, the provided parameters contain context information that is used by the NS selection model to select the appropriate NS identifier from all the NSDs that have been pre-onboarded in the NFVO. Fig. 9 depicts the detailed interactions triggered by the NS selection model. The second type of intent incorporates placement context, also passed as a parameter to the intent body. The VIM selection model processes the provided context, along with telemetry information available in the data processing subsystem, to select the VIM account where the selected network service should be provisioned. Fig. 9 also depicts the detailed interactions triggered by the VIM selection model.

D. DEMONSTRATION

We describe in this section a demonstration of our implemented use cases, and a benchmark that measures for each use case the intent execution time, defined as the time required from the moment the user inserts the intent into the intelligence stratum until the corresponding actions are enforced in the network. Fig. 10 contains three snapshots of the public demonstrations of each use case available in Youtube.

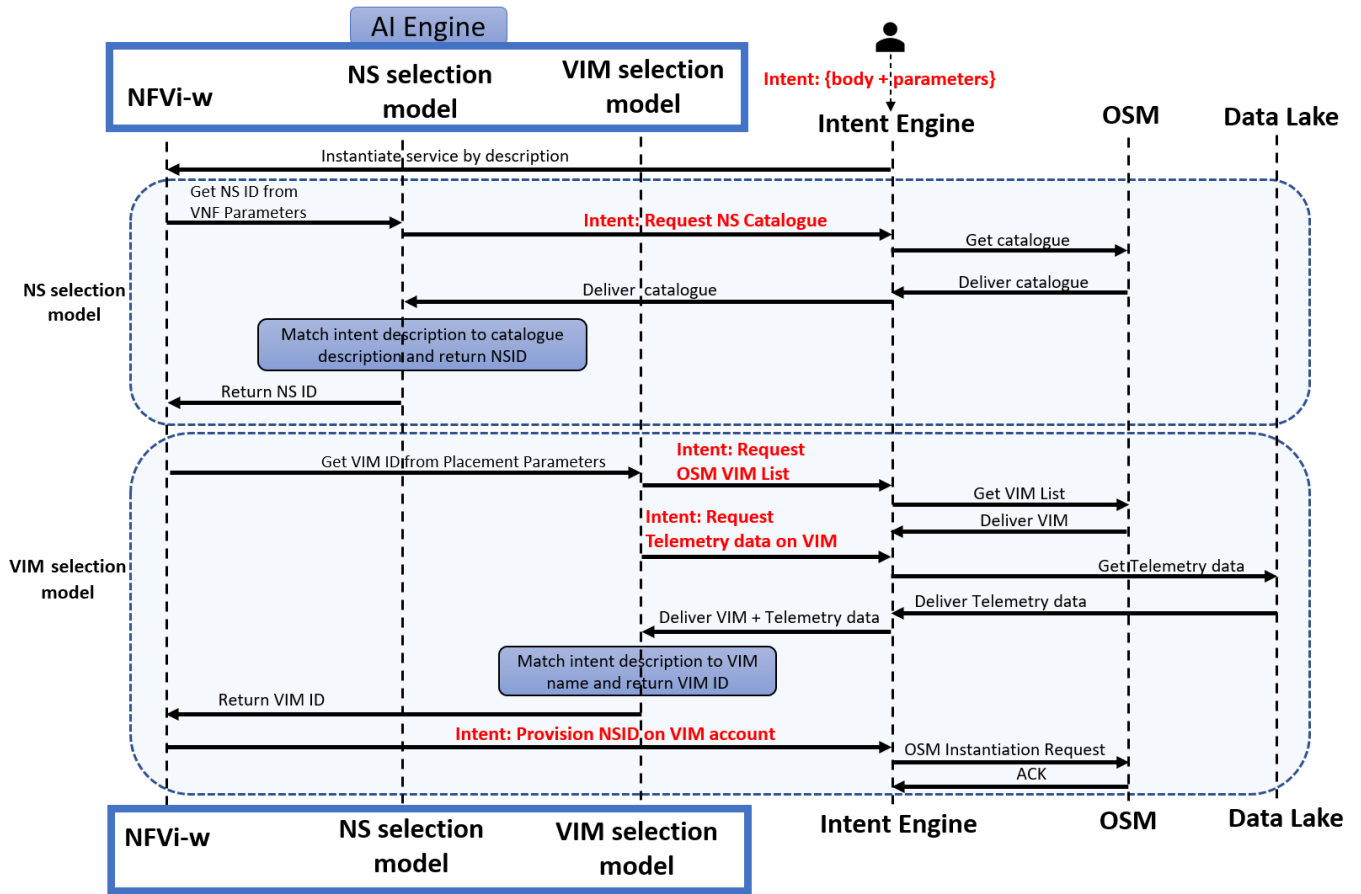
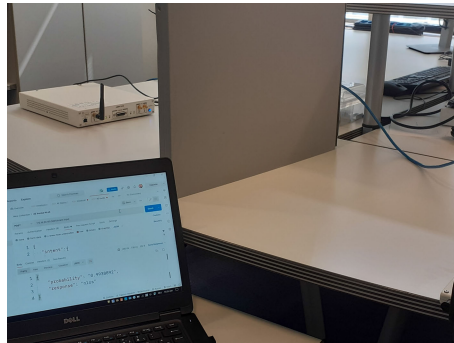


FIGURE 9. Intent based NFVaaS use case workflow.

```

POST localhost:5001/intent-input
{
  "intent": {
    "request": "Create a slice using Slice Creation Workflow",
    "parameters": {
      "name": "nova",
      "user-list": [
        {
          "imsi": "001035432100005"
        }
      ],
      "location": {
        "latitude": 0.0,
        "longitude": 0.0
      },
      "technology": {
        "AMARISOFT_CELL",
        "Some_ACCESS"
      }
    }
  }
}
    
```



(a) Intent based slice provisioning through POSTMAN based interface (b) Intent based NLoS identification, with object blocking line of sight (c) Intent based network service deployment, depicting the robot mounted 360° camera

FIGURE 10. Snapshots of the three use case demonstrators available in [36].

Table 4 reports the intent execution times for each use case. Next, we discuss each use case in detail.

An implementation and demonstration of our intent based slice provisioning is available in [37], where we can see the full slice provisioning process including the intent engine, the slice manager and the RAN controller. Fig. 10(a) depicts a snapshot of our demonstration where we can see how the slice

provisioning intent is issued through a POSTMAN interface. To benchmark the intent execution time we repeat 10 slice provisioning experiments and compute an average time of approximately 3 minutes, which is a reasonable time considering the fact that provisioning a slice requires re-configuring 5G cells and APs as well as spanning new core network related VNFs.

**TABLE 4.** Average intent execution time per use case.

	Slice provisioning	NLoS identification	NS deployment
Average intent execution time	≈ 3 minutes	≈ 1 second	≈ 4 minutes

Regarding the NLoS identification use case, an implementation and demonstration is available in [38]. Fig. 10(b) provides a snapshot of the demonstrator where we can see an obstacle blocking the communication path between the UE and gNB. We can see in the demonstrator how upon triggering a request by the operator (through POSTMAN interface), the “LoS” decision on the link condition and its corresponding probability is returned. Next, when the direct link is blocked by a metal object, the returned decision switches to “NLoS.” Table 4 reports the average time from intent provisioning to NLoS detection to be of one second, which is reasonable given the fact that the model is trained offline and only needs to feed the CIR and return the decision on the link condition. Furthermore, the accuracy of the NLoS identifier model has been reported in [34] and [39] and is around 95%.

The NFV use case as demonstrated in an *Intent based end-to-end NFVI-as-a-service* deployment is captured in the snapshot shown in Fig. 10(c). This is associated to a smart tourism service, in which surveillance video from a 360° camera mounted on a mobile robot is requested remotely using intents generated by a public safety officer’s monitoring device. To implement this service a virtual media forwarding unit is provisioned at the edge in the museum, which forwards the robot camera feed to the appropriate user device. As shown in Table 4, the average time to instantiate a video service from when the intent is initiated to NS creation is 234 seconds (≈4 minutes). The layers of interactions between and within OSM, intent engine and the image size account for this duration. A video demonstration of this use case is available in [36].

## V. CONCLUSION

There is a strong interest in designing systems that can simplify the interactions between humans and complex digital systems through the use of natural language based interfaces. Intent based networking has been devised as the architectural framework to achieve this simplification in the management of mobile networks, where advances in this regard are especially needed in the area of private 5G networks owned by verticals that often lack skilled 5G personnel.

In this paper, we have introduced the 5G-CLARITY intelligence stratum that offers an NLP interface to private 5G network operators. Our proposed intelligent stratum features an Intent Engine that receives as input intents in the form of English level sentences, and an AI engine that hosts ML models, which encode complex API workflows and provide the low-level context missing in the original intent definition.

We describe a prototype implementation of our intelligence stratum and demonstrate it with three private 5G network use cases, namely a slice provisioning use case, an indoor positioning use case, and a network service deployment use case. All use cases are benchmarked in terms of intent provisioning time.

As part of our future work, we plan to continue investigating how recent advances in NLP, e.g. based on the use of Large Language Models (LLMs) [27], can be leveraged to further simplify the operation of private 5G networks. For example, an LLM could learn to map a user intent directly to a complex sequence of API endpoints, thus simplifying the need for the manually handcrafted workflow models that are needed in our current system to support different use cases.

## ACKNOWLEDGMENT

(Joseph McNamara, Daniel Camps-Mur, Meysam Goodarzi, Hilary Frank, Lorena Chinchilla-Romero contributed equally to this work.)

## REFERENCES

- [1] J. Prados-Garzon, P. Ameigeiras, J. Ordóñez-Lucena, P. Muñoz, O. Adamuz-Hinojosa, and D. Camps-Mur, “5G non-public networks: Standardization, architectures and challenges,” *IEEE Access*, vol. 9, pp. 153893–153908, 2021.
- [2] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, “A survey on intent-driven networks,” *IEEE Access*, vol. 8, pp. 22862–22873, 2020.
- [3] A. Collet, A. Banchs, and M. Fiore, “LossLeap: Learning to predict for intent-based networking,” in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 2138–2147.
- [4] T. Cogalan, D. Camps-Mur, J. Gutierrez, S. Videv, V. Sark, J. Prados-Garzon, J. Ordóñez-Lucena, H. Khalili, F. Canellas, and A. Fernandez-Fernandez, “5G-CLARITY: 5G-advanced private networks integrating 5G NR, WiFi, and LiFi,” *IEEE Commun. Mag.*, vol. 60, no. 2, pp. 73–79, Feb. 2022.
- [5] *Technical Specification Group Services and System Aspects; Telecommunication Management; Study on Scenarios for Intent Driven Management Services for Mobile Networks*, document TR 28.812, V17.1.0, 3GPP, 2020.
- [6] *Management and Orchestration; Intent Driven Management Services for Mobile Networks*, document TR 28.312, V17.1.1, 3GPP, 2022.
- [7] *Experiential Networked Intelligence (ENI); InTent Aware Network Autonomicity (ITANA)*, document ETSI GR ENI 008, V2.1.1, 2021.
- [8] *Zero-Touch Network and Service Management (ZSM); Reference Architecture*, document ETSI GS ZSM 002, V1.1.1, 2020.
- [9] *5G Clarity 5G ZORRO ETSI ZSM POC Demo*. Accessed: Feb. 8, 2023. [Online]. Available: [https://www.youtube.com/watch?v=heU\\_ceO315s](https://www.youtube.com/watch?v=heU_ceO315s) accessed: Feb. 8, 2023
- [10] *Zero-Touch Network and Service Management (ZSM); Means of Automation*, document ETSI GR ZSM 005, V1.1.1, 2020.
- [11] *Autonomous Networks—Reference Architecture*, document TM Forum 859 IG1251, V1.0, 2021.
- [12] *Intent in Autonomous Networks*, document TM Forum IG1253, 2022.
- [13] *Intent Common Model*, document TM Forum IG1253A, V1.1.0, 2021.
- [14] *Intent Extension Models*, document TM Forum IG1253B, V1.0.0, 2021.
- [15] *Intent Life Cycle Management and Interface*, document TM Forum IG1253C, V1.1.0, 2022.
- [16] *Intent Manager Capability Profiles*, document TM Forum IG1253D, V1.0.0, 2021.
- [17] L. Velasco, M. Signorelli, O. G. De Dios, C. Papagianni, R. Bifulco, J. J. V. Olmos, S. Pryor, G. Carrozzo, J. Schulz-Zander, M. Bennis, R. Martinez, F. Cugini, C. Salvadori, V. Lefebvre, L. Valcarenghi, and M. Ruiz, “End-to-end intent-based networking,” *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 106–112, Oct. 2021.

- [18] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Network slice life-cycle management for 5G mobile networks: An intent-based networking approach," *IEEE Access*, vol. 9, pp. 80128–80146, 2021.
- [19] T. A. Khan, A. Muhammad, K. Abbas, and W.-C. Song, "Intent-based networking platform: An automated approach for policy and configuration of next-generation networks," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 1921–1930.
- [20] *Open Source MANO*. Accessed: Feb. 8, 2023. [Online]. Available: <https://osm.etsi.org/>
- [21] *FlexRAN*. Accessed: Feb. 8, 2023. [Online]. Available: <https://mosaic5g.io/flexran/>
- [22] *M-CORD*. Accessed: Feb. 8, 2023. [Online]. Available: <https://opennetworking.org/m-cord/>
- [23] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.
- [24] A. Chaudhari, A. Asthana, A. Kaluskar, D. Gedia, L. Karani, L. Perigo, R. Gandotra, and S. Gangwar, "VIVoNet: Visually-represented, intent-based, voice-assisted networking," 2019, *arXiv:1904.03228*.
- [25] *Alexia Skills Toolkit*. Accessed: Feb. 8, 2023. [Online]. Available: <https://github.com/alexa/ask-toolkit-for-vscode>
- [26] B. K. Saha, L. Haab, and L. Podleski, "Intent-based industrial network management using natural language instructions," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2022, pp. 1–6.
- [27] T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, and C. J. Cai, "PromptChainer: Chaining large language model prompts through visual programming," in *Proc. CHI Conf. Human Factors Comput. Syst. Extended Abstr.*, Apr. 2022, pp. 1–10.
- [28] *5G-CLARITY Semantic Data Aggregator*. Accessed: Feb. 10, 2023. [Online]. Available: <https://github.com/giros-dit/semantic-data-aggregator>
- [29] L. Fallon, S. van der Meer, and J. Keeney, "Apex: An engine for dynamic adaptive policy execution," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, Apr. 2016, pp. 699–702.
- [30] J. McNamara, E. Aumayr, L. Fallon, and E. Fallon, "A flexible interpreter for intent realisation," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, Apr. 2022, pp. 1–6.
- [31] A. Betzler, D. Camps-Mur, and M. Catalan, "G-ADRR: Network-wide slicing of Wi-Fi networks with variable loads in space and time," *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 3986–4000, Nov. 2022.
- [32] S. Dwivedi, R. Shreevastav, F. Munier, J. Nygren, I. Siomina, Y. Lyazidi, D. Shrestha, G. Lindmark, P. Ernstrom, E. Stare, and S. M. Razavi, "Positioning in 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 38–44, Dec. 2021.
- [33] J. Ordonez-Lucena. (Jul. 2021). *5G-CLARITY Deliverable D4.2 Validation of 5G-CLARITY SDN/NFV Platform, Interface Design With 5G Service Platform*. [Online]. Available: [https://www.5gclarity.com/wp-content/uploads/2021/11/5G-CLARITY\\_D42.pdf](https://www.5gclarity.com/wp-content/uploads/2021/11/5G-CLARITY_D42.pdf)
- [34] M. Goodarzi, V. Sark, N. Maletic, J. G. Teran, G. Caire, and E. Grass, "DNN-assisted particle-based Bayesian joint synchronization and localization," *IEEE Trans. Commun.*, vol. 70, no. 7, pp. 4837–4851, Jul. 2022.
- [35] D. Borsatti, W. Cerroni, G. Davoli, and F. Callegati, "Intent-based service function chaining on ETSI NFV platforms," in *Proc. 10th Int. Conf. Netw. Future (NoF)*, Oct. 2019, pp. 144–146.
- [36] *5G-CLARITY*. Accessed: Feb. 10, 2023. [Online]. Available: <https://www.youtube.com/@5g-clarity458>
- [37] *5G-CLARITY Intent Based Slice Management Demonstration*. Accessed: Feb. 8, 2023. [Online]. Available: <https://www.youtube.com/watch?v=5Jsc2ds-etI&t=43s>
- [38] *5G-CLARITY Intent Based NLoS Identification Demonstration*. Accessed: Feb. 8, 2023. [Online]. Available: <https://www.youtube.com/watch?v=FgdyJBpIQ>
- [39] J. Ordonez-Lucena, D. Camps-Mur, L. Chinchilla, J. Perez-Romero, M. Goodarzi, A. Purwita, A. Garcia, E. Aumayr, C. C. Meixner, R. Bian, T. Cogalan, and M. Ghoraiishi, "5G-CLARITY deliverable D4.2 validation of 5G-CLARITY SDN/NFV platform, interface design with 5G service platform, and initial evaluation of ML algorithms," Tech. Rep., 2022. [Online]. Available: [https://www.5gclarity.com/wp-content/uploads/2021/11/5G-CLARITY\\_D42.pdf](https://www.5gclarity.com/wp-content/uploads/2021/11/5G-CLARITY_D42.pdf)



**JOSEPH MCNAMARA** received the B.Sc. (Hons.) degree in software development from the Technological University of the Shannon, Ireland, where he is currently pursuing the Ph.D. degree in policy-based network management. On completion of the undergraduate, he was awarded funding to continue studies under the Irish Research Council Enterprise Partnership Scheme, in 2018. In the following years, he has contributed to several Horizon 2020 and Horizon Europe Projects as a Task Leader. He is a Senior Researcher with Ericsson LMI, Ireland. His research interests include the mitigate aspects of the complexity for humans who wish to influence policy-based closed loop management, particularly intent-based networking, intelligent controllers, real-time analytics, and adaptive policy.



**DANIEL CAMPS-MUR** received the master's and Ph.D. degrees from the Polytechnic University of Catalonia (UPC), Spain, in 2004 and 2012, respectively. He is currently leading the Mobile and Wireless Internet (MWI) Group, i2CAT, Barcelona, Spain. In addition, he is the Technical Coordinator of the 5G-PPP Project 5G-Clarity, which investigates convergence of 5G NR, WiFi, and LiFi. Previously, he was a Senior Researcher with NEC Network Laboratories, Heidelberg, Germany. His research interests include 5G and 6G mobile networks, software-defined networking, and communications protocols for the Internet of Things.



**MEYSAM GOODARZI** received the B.Sc. degree in electrical communication engineering from the K. N. Toosi University of Technology, Iran, in 2013, the M.Sc. degree in communications and multimedia engineering from the Friedrich-Alexander University of Erlangen-Nuremberg, Germany, in 2017, and the M.Sc. degree in economics from the University of Paris 1 Panthéon-Sorbonne, France, in 2022. He is currently pursuing the Ph.D. degree with the IHP GmbH–Leibniz-Institute for Innovative Microelectronics, Humboldt University of Berlin, Germany. His current research interests include resource allocation, signal processing, applications of machine learning and probabilistic graphical models in wireless communications (specifically synchronization and localization), and network economics.



**HILARY FRANK** (Graduate Student Member, IEEE) received the B.Sc. degree from the University of Port Harcourt, Nigeria, and the M.Sc. degree in mobile and high-speed telecommunication networks from Oxford Brookes University, U.K. He is also an alumnus of the United States Telecommunications Training Institute. He is currently a Research Associate and writing-up his Ph.D. thesis with the Smart Internet Laboratory, High Performance Networks Group, Electrical and Electronic Engineering Department, University of Bristol, U.K. His research interests include 5G non-public networks (5G NPN), multi-connectivity driven wireless networks, and techno-economic assessment of 5G networks.



**LORENA CHINCHILLA-ROMERO** received the B.Sc. and M.Sc. degrees from the University of Granada (UGR), Granada, Spain, in 2018 and 2020, respectively. She is currently pursuing the Ph.D. degree with the WiMuNet Laboratory Research Group, Department of Signal Theory, Telematics and Communications, UGR. Her research interests include private 5G networks, radio resource management, and machine learning.



**ADRIANA FERNÁNDEZ-FERNÁNDEZ** received the degree in telecommunications and electronic engineering and the Ph.D. degree in telematics engineering from Universitat Politècnica de Catalunya (UPC). She is a Senior Researcher with i2CAT Foundation, Spain. Before joining i2CAT, she was a Postdoctoral Researcher with the BAMPLA Research Group, UPC. She is the author/coauthor of over 20 papers in peer-reviewed international journals and conference proceedings.

She has been awarded several scholarships from the Spanish Government through the Research Training Program (FPI). Her current research interests include management, orchestration, and network slicing in 5G networks, as well as on NFV and cloud-native technologies. Her research interests include communication network modeling and optimization, energy-aware routing, SDN, and traffic engineering.



**FERRÁN CAÑELLAS** received the degree in telecommunication systems engineering from Universitat Politècnica de Catalunya (UPC), in 2016, and the M.Sc. degree in telecommunications from Danmarks Tekniske Universitet (DTU), in 2018. He is a Senior Research and Development Engineer with i2CAT Foundation. Before joining i2CAT, he was a Research Assistant with DTU. Currently, his research is focused on the experimentation with multiple wireless access technologies, including IEEE 802.11 and cellular networks. His research interests include SDN and multipath TCP.

technologies, including IEEE 802.11 and cellular networks. His research interests include SDN and multipath TCP.



**SHUANGYI YAN** (Member, IEEE) received the B.E. degree in information engineering from Tianjin University, Tianjin, China, in 2004, and the Ph.D. degree in optical engineering from the Xi'an Institute of Optics and Precision Mechanics, CAS, Xi'an, China, in 2009. He is a Senior Lecturer with the High Performance Networks Group, Smart Internet Laboratory, University of Bristol. From 2011 to 2013, he was with Hong Kong Polytechnic University, Hong Kong, as a Postdoctoral

Researcher, investigating on the spectra-efficient long-haul optical transmission systems and low-cost short-range transmission systems. In July 2013, he joined the University of Bristol. His research focuses on machine-learning applications in dynamic optical networks and 5G beyond networks, programmable optical networks, and data center networks.

...