Intent-Based Management of Next-Generation Networks: an LLM-centric Approach

Abdelkader Mekrache, *Member, IEEE,* Adlen Ksentini, *Senior Member, IEEE,* and Christos Verikoukis, *Senior Member, IEEE.*

***Abstract*—Intent-Based Networking (IBN) management has emerged as an alternative approach to simplify network configuration and management by abstracting the complexities of low-level configurations. Existing IBN solutions typically rely on human-readable structures like JSON or YAML to define Intents, which still require expertise in understanding these structures. A natural evolution of IBN is to use natural language instead of defined structures. However, this approach introduces complexities related to natural language understanding. Fortunately, Large Language Models (LLMs) offer a promising solution. In this paper: (i) We propose a novel LLM-centric Intent Life-Cycle (LC) management architecture designed to configure and manage network services using natural language. The architecture spans the complete Intent LC, encompassing decomposition, translation, negotiation, activation, and assurance; (ii) We identify key open issues and challenges related to IBN within our proposed architecture; (iii) We demonstrate the effectiveness of the architecture by developing a component within the EURECOM 5G facility [1], leveraging LLMs to implement the essential Intent LC procedures; (iv) We validate the proposed system through real-world deployment, showcasing its capability to define, decompose, translate, and activate Intents using natural language.**

***Index Terms*—Intent-based networking, Intent life-cycle, natural language, large language models, human feedback.**

I. Introduction

Intent-based networking (IBN) plays a crucial role in enabling autonomous networks by specifying goals and constraints at a higher level to the Network Management System (NMS) [2]. It introduces the notion of "Intent," which represents an abstract operational goal provided as input to the NMS. Subsequently, the latter generates the necessary low-level configurations to fulfill these Intents. Although IBN is a relatively new term and technology, significant efforts have been dedicated to defining and standardizing it, including the 3rd Generation Partnership Project (3GPP) [3], the European Telecommunications Standards Institute (ETSI) [4], and the TM Forum [5]. In these standards, high-level Intents are specified using human-readable structures like JSON or YAML. For instance, ETSI standards [6] define it using the Network Service Descriptor (NSD), a JSON structure designed to configure network services.

However, the current model of expressing Intents still requires significant effort in writing these structures, demanding a detailed comprehension of the model specified by the North-Bound Interface (NBI). This process is not always straightforward, and adhering to the structure of these NBIs is time-consuming. A natural evolution for IBN is to move beyond the use of human-readable strucures and transition towards natural language. An example of an Intent request using natural language for deploying a Communication Service (CS) for a 5G network serving eXtended Reality (XR) users is:

Example I.1

I need a network composed of three XR applications: an augmented reality content server, a mixed reality collaboration platform, and a virtual reality simulation engine. Each application requires 4 vCPU and 2 Gigabytes (Gbytes) of memory. All XR applications are connected using 5 Gbytes/sec links. The clients are connected through a 5G network located in the Nice area and tolerate a maximum latency of 5 ms.

In this example, the Intent comprises different sub-Intents specific to the technological domain involved in supporting this CS. Indeed, one part is dedicated to the needed computing resources to run the mentioned applications on the Cloud/Edge, another one for networking, and the last part to the Radio Access Network (RAN). Once the Intent is specified, there are many required steps to reach the deployment and propagation of the Intent's objective up to the different infrastructures composing the technological domains (Cloud/Edge, networking, and RAN). These steps correspond to the Intent Life-Cycle (LC) management procedures: *(i) Intent decomposition* that extracts the information on each technological domains that need to be involved in the deployment phase; *(ii) Intent translation* that translates decomposed Intents to Infrastructure-Level Intents (ILIs) specific to each domain. Then, *(iii) Intent negotiation*, *(iv) Intent activation*, and *(v) Intent assurance* are the steps that enforce the Intent on the technological domain infrastructure.

Intents LC management based on natural language represents the simplest form of IBN-enabled system communication with the network. Nevertheless, natural language's unstructured and ambiguous nature poses challenges for IBN systems in extracting essential information and producing precise low-level configurations. Additionally, users from diverse regions can add complexity, necessitating an approach capable of interpreting Intents across multiple human languages, even in the presence of grammatical errors. Fortunately, with the rapid explosion in the Natural Language Processing (NLP) area, Large Language Models (LLMs) become very powerful in understanding human languages.

A. Mekrache is with EURECOM, France (e-mail: abdelkader.mekrache@eurecom.fr).

A. Ksentini is with EURECOM, France (e-mail: adlen.ksentini@eurecom.fr).

C. Verikoukis is with ISI/ATH and University of Patras, Greece (chverik@gmail.com).

To address the gap in the literature regarding Intent LC management, this paper introduces a novel, LLM-centric high-level architecture designed to manage Intents LC from an End-to-End (E2E) perspective. This involves defining, handling, and enforcing Intent objectives. Our architecture leverages cutting-edge AI advancements, particularly LLMs, to tackle all previously mentioned E2E Intent LC procedures. We demonstrate the efficiency of this architecture by upgrading the EURECOM 5G facility [1] with natural language Intent handling, i.e., decomposing and translating Intents using LLMs. Subsequently, Intent activation and assurance are managed using its existing functionalities [1]. This upgrade relies on open-source, state-of-the-art LLMs incorporating Human Feedback (HF) to learn from previous experiences. The major contributions of this paper are as follows:

- *End-to-End IBN LC Management Architecture:* We propose a comprehensive, LLM-centric architecture that spans all stages of IBN from Intent definition, decomposition, and translation to activation and assurance. This architecture ensures seamless interaction with the NMS for all users, including those with limited domain knowledge who will use natural language instead of ILIs. It leverages the latest AI advancements, particularly LLMs.
- *Identification of Key Open Issues and Challenges:* Within our architecture, we identify and discuss the main open issues related to IBN that need to be addressed. These challenges are crucial for advancing beyond the current state-of-the-art and providing a solid foundation for future research and development in this area.
- *LLM-Based Intent Decomposition and Translation System:* To showcase the effectiveness of the proposed architecture, we develop an innovative system within the 5G EURECOM facility [1] using LLMs for Intent decomposition and translation. This system utilizes few-shot learning and HF to transform natural language Intents into ILIs.
- *Real-World Deployment and Testing:* Our proposed framework is implemented and tested in real-world scenarios using a single NVIDIA A100 GPU with 40GB of vRAM, leveraging the Code Llama LLM [7]. This demonstrates the practical applicability of our architecture on the 5G facility [1], enabling the configuration of network services using natural language.

The remaining sections of this paper are structured as follows: Section II describes related works on IBN. Section III presents the high-level architecture to handle Intent's LC, open issues and challenges, and our Intent decomposition and translation solution. Section IV provides an analysis of the solution's performance. Section V discusses limitations and outlines future work. Finally, Section VI concludes the paper.

II. Related works

IBN is a transformative approach to network management that prioritizes user Intent and establishes a dynamic, adaptable network ecosystem [2]. Introduced relatively recently, IBN is actively being standardized by organizations like 3GPP, ETSI, and TM Forum. Each of these organizations has formed dedicated study groups to explore IBN. In 2018, 3GPP started standardization efforts, introducing the concept of Intent-driven management and proposing an Intent-driven management service for managing 5G networks and services [3]. Concurrently, ETSI established the zero-touch network and service management working group to delve into Intent-based automation and Intent-based service orchestration [4]. Finally, the TM Forum, as part of its autonomous networks framework, is standardizing IBN in [5]. However, all these standards groups focus on what and why Intents are needed, while how an Intent objective is written using natural language, translated, and enforced on the infrastructure is still missing.

Various research efforts have addressed numerous challenges in the various components of IBN. For instance, the authors of [8] tackled the Intent translation process by developing a chatbot that elicits context-specific information from users of packet-optical networks. To address Intent activation, researchers in [9] proposed an Intent negotiation framework to resolve conflicts arising from limited resource availability in IBN. Additionally, *Zheng et al.* [10] proposed an Intent assurance solution for data center IBN systems, utilizing specific data preparation procedures and Machine Learning (ML) models for time series forecasting. Furthermore, several research works have proposed E2E IBN architectures. For example, *Velasco et al.* [11] presented a scalable 5G architecture by integrating secure ML-powered IBN, fostering application-level resilience and intelligence. Concurrently, with the explosion of LLMs in the NLP domain, the research community is shifting towards utilizing and standardizing them in the context of networking and telecommunications.

A few studies have employed LLMs in IBN [12, 13]. Researchers in [12] aimed to generate Python code from natural language Intent with LLMs, enabling infrastructure management. However, they relied on a customized Python library to execute the generated program, limiting its applicability to other infrastructure management systems. Our system generates standardized ILIs, such as NSD and RAN Descriptor (RAND), aimed at universal acceptance across public and private 6G infrastructures adhering to these standards. This is achieved through a two-stage process: decomposing the Intent into different domains and generating standardized ILI for each domain. This approach ensures compliance with various standards across multiple domains using a single Intent. Authors of [13] utilize LLMs, along with other AI techniques, to achieve autonomous fault localization, strategy generation, and strategy verification. However, establishing an E2E Intent LC management for CS deployment in 6G systems (combining Intent decomposition, negotiation, translation, activation, and assurance) remains an open challenge. To address this gap, our work aims to leverage the latest advancements in LLMs to support E2E Intent LC management. We demonstrate the effectiveness of the proposed architecture by implementing the decomposition and translation steps in E2E Intent LC management using LLMs and rely on the existing functionalities of the EURECOM 5G facility [1] for activation and assurance. Furthermore, both [12, 13] rely on OpenAI's closed-source ChatGPT LLM, which lacks full control compared to open-source LLMs utilized in our approach.

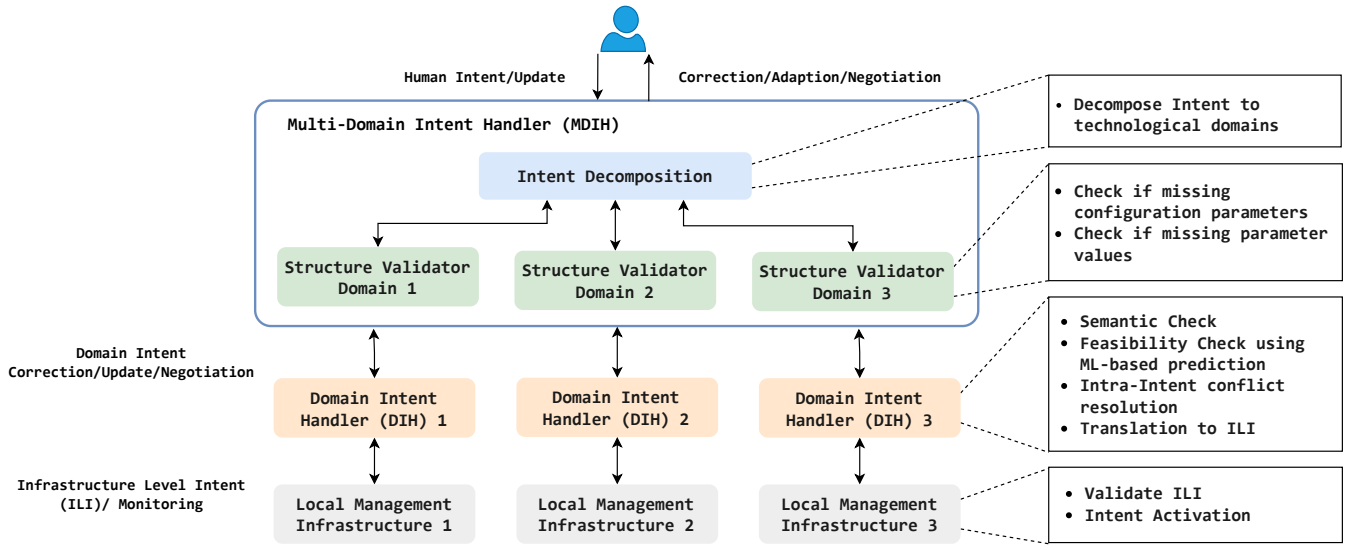

Fig. 1: High-level architecture design to handle natural language-based Intents LC.

III. IBN LC Management System Design

In this section, we begin by introducing our high-level architecture that aims to handle the LC of Intents from their definition up to their admission and deployment over the infrastructure, considering the context of multi-domain CS deployment in 5G and beyond. Then, we expose the different open challenges related to the realization of the architecture's objectives and discuss some research directions. Finally, we introduce novel solutions that address two critical challenges related to natural language-based decomposition and translation employing an LLM-centric approach. We demonstrate the application of the proposed solutions using the 5G facility [1] as an example to deploy a CS spanning over two technological domains: Cloud/Edge and RAN.

A. Proposed Architecture

To effectively address Intent LC, we propose a high-level architecture for natural language-based Intents, as depicted in Fig. 1. In this system, the manager or user interacts with the Multi-Domain Intent Handler (MDIH) component to deploy a CS across various infrastructure domains. The MDIH assists the manager or user in correcting, updating, or negotiating the Intent to ensure that the latter can be deployed. This interaction is envisioned to be similar to using a chatbot. Each infrastructure domain is managed by a Local Management System (LMS) that utilizes ILI to deploy the corresponding CS components on the infrastructure. An example of ILI is the NSD defined by the ETSI NFV group [6] for deploying services on virtualized platforms, or the helm chart model used by Kubernetes[1] for deploying cloud-native services. Our proposed system is designed to be infrastructure and network architecture agnostic, offering native support for all technological domains comprising 6G, i.e., cloud, edge, transport, and RAN. The Intent-based architecture illustrated in Fig. 1 aims to abstract the underlying infrastructure as much as possible. The only infrastructure-specific aspect lies in the translation step (at the DIH), which is essential for generating ILI per domain. In many cases, ILI also abstracts the complexities of the underlying layers. For example, the ETSI NSD uses blueprints to abstract various computing information (see Fig. 2 for an NSD example).

In the proposed solution, the CS is defined using natural language. No restriction on the formulation model (or grammar) is imposed; the user is free to define the Intent by using natural language. The Intent is then handled by a MDIH, which belongs to the CS provider and is integrated into the OSS/BSS as a component. The internal structure of MDIH includes an Intent decomposition module, tasked with decomposing the Intent into sub-parts corresponding to each domain. The Intent decomposition module should understand the semantic (i.e., the meaning) of the Intent to decompose it; in Fig. 1, we assume three domains, e.g., Cloud/Edge Intent, networking Intent, and RAN Intent. While considering three domains in this example, the proposed approach can be generalized to a higher number of domains. To decompose the Intent, we rely on LLMs, known for their ability to understand natural language and extract semantic information. After decomposition, each subpart is sent to the structure validator module, which validates the Intent's structure for each technological domain. The validation process checks that all required parameters and associated values to deploy a CS on a domain, are included in the Intent. For instance, an Intent to deploy a CS on Cloud/Edge should indicate the software image location and resources needed for deployment. If a parameter or value is missing, the structure validator request the user to correct the Intent. The process ends when the Intent structure is valid. As stated earlier, a chatbot-like system interacts with the user, providing proposals to correct the Intent formulation until final validation. The chatbot approach corrects the Intent's structure, avoids intra-Intent conflicts, and ensures the Intent's feasibility on the infrastructures. The objective is to prevent any structural error before forwarding

[1]https://kubernetes.io

the request to the Domain Intent Handler (DIH).

Subsequently, each domain-specific Intent is handled separately by a DIH, which has the role of doing the semantic validation of the Intent and the translation to ILI specific to an LMS handling a domain-specific infrastructure. The different actions the DIH has to execute are in the following order:

1) Check if there are intra-Intent conflicts between the objectives. For instance, if the Intent requests radio throughput higher than a value (V1) and interference lower than a value (V2), then a conflict exists as increasing the radio throughput will also increase the interference due to the usage of a higher data rate modulation scheme. In this case, a correction of the Intent through the chatbot is proposed to the end user to avoid conflicts. When the Intent is conflict-free, the following action is triggered; otherwise, an update is requested from the Intent owner through the chatbot.
2) Check if the Intent can be satisfied, which involves the feasibility check process. Indeed, the latter aims to check if the Intent objectives can be fulfilled on the domain-specific infrastructure considering the available resources. This step is essential as it is equivalent to an admission control process, avoiding that the Intent is deployed while its objectives cannot be sustained. This requires that the DIH uses a prediction of the resource evolution of an infrastructure aiming to decide if the Intent can be accepted. In this context, an external module to the DIH will run AI/ML prediction models that consume monitoring information collected from the LMS. The predicted model is built per LMS.
3) Translate the Intent to ILI and activate the Intent by sending it to the LMS. The translation process will be done using an LLM-based module that translates natural language to ILI as expected by LMS. The LMS will validate the ILI; if any error is detected, a request to correct the ILI is sent to the DIH. This closed control loop between the DIH and LMS allows the improvement of the LLM KB to enhance future translation accuracy.
4) Request the collection of Intent's Key Performance Indicators (KPIs) to start the Intent assurance step, which consists of validating the Intent performance and achieving the requested service level agreement. Besides, the assurance step consists of deriving LC management decisions using reinforcement learning or policy-based approaches to correct the performance if a degradation is detected or correct inter-Intent conflicts.

B. Challenges and Open Issues

Although the proposed architecture offers a promising E2E framework for Intent LC management, its realization faces several challenges. These challenges encompass various aspects of Intent processing procedures, including Intent decomposition and structure validation, Intent semantic validation and intra-Intent conflict resolution, as well as Intent translation.

1) Intent decomposition and structure validation: Intent task decomposition poses a significant challenge in IBN, prompting researchers to explore novel AI-based methods. However, the complexity and ambiguity of natural language make decomposing tasks particularly challenging. On the other hand, validating the Intent structure involves ensuring the presence of required parameters in the user's Intent. This presents multiple challenges: How do we extract the corresponding Intent for each domain? How do we validate the Intent structure? How do we interact with the user? LLMs are actively being developed to support various NLP tasks. Owing to their context detection and human language comprehension abilities, they can be utilized for intent decomposition to understand which Intent concerns which domain. This feature is beneficial in 6G, as the latter involves heterogeneous technological domains. Subsequently, structure validation can be achieved through conventional rule-based methods, such as human intervention (via HF), or by employing context-aware techniques to extract essential parameters from user inputs. For instance, named entity recognition is beneficial for detecting specific required parameters. Nonetheless, incorporating HF can introduce security vulnerabilities when dealing with erroneous or manipulated input. To address these risks, a recommended approach is to establish robust HF validation mechanisms. For instance, one effective strategy is to engage experts to validate feedback before it is used by the system.

2) Intent semantic validation and intra-Intent conflict resolution: On the one hand, semantic validation involves verifying that the user's Intent aligns with the capabilities of the underlying system. Researchers have employed NLP techniques, such as LLMs, and Knowledge Graphs (KGs), to identify key concepts and relationships in a given text [14]. They then utilize AI/ML approaches, such as deep learning, to predict and verify conformity with the underlying system for feasibility checks. Notably, LLMs can be used synergistically with AI/ML to perform semantic validation and feasibility checks, respectively. On the other hand, conflicts may arise between users and service providers due to a misalignment between service requirements and the capabilities of underlying resources. To address these conflicts, Intent negotiation modules have been developed to initiate negotiation processes [9]. These modules generate alternative Intents based on resource availability, allowing users to accept or reject them. Challenges include resolving conflicting objectives within Intents and assessing the feasibility of infrastructure. Despite proposed frameworks, standardization in Intent negotiation is still being developed. In this context, LLMs represent a promising option due to their advanced reasoning capabilities. However, LLMs require substantial knowledge before making decisions, necessitating further training. Consequently, one approach is to emphasize creating training and evaluation telecommunications datasets. Subsequently, training LLMs on these datasets, such as [15], could create expert LLMs in this domain capable of efficiently making informed decisions to resolve conflicts within Intents.

3) Intent translation: IBN systems are a significant area of research interest, with researchers proposing new AI-based methods to translate users' Intents into network configurations, operations, and maintenance strategies. Tools such as chatbots [8] have emerged to simplify the Intent translation process. Additionally, reasoning approaches have been used

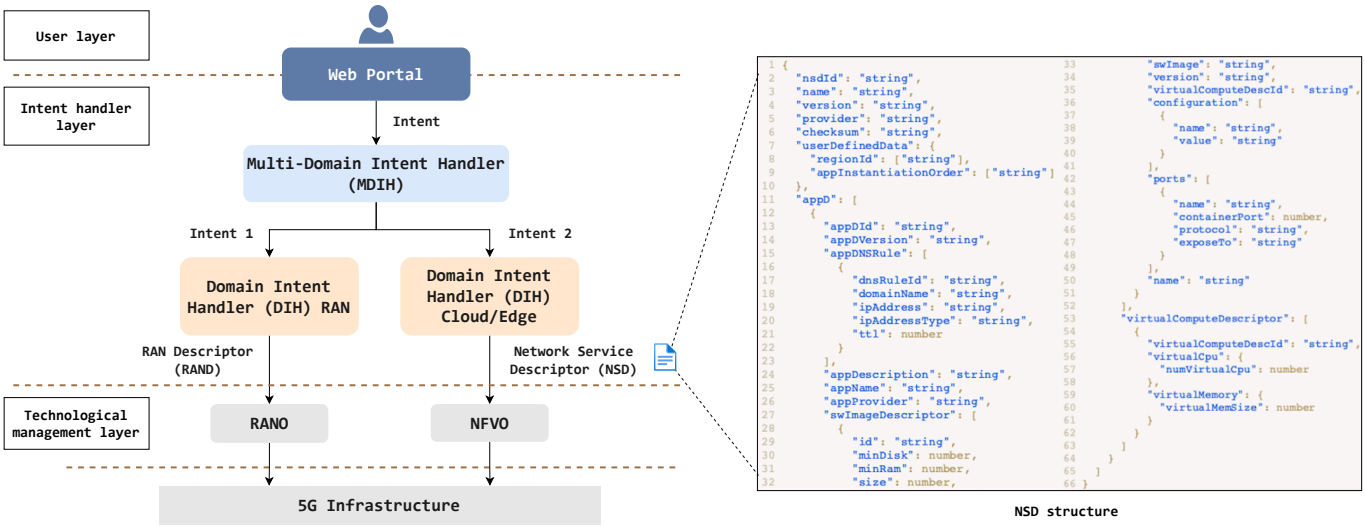

Fig. 2: Natural language-based Intent LC management in the EURECOM 5G facility [1].

in conjunction with NLP to improve translation performance. On the other hand, LLMs also offer promising reasoning abilities for translation tasks. However, in the context of IBN, a few challenges remain: How do we translate natural language Intents into ILIs? How do we validate the correctness of the generated ILIs? First, LLMs can be adapted to this task using fine-tuning or few-shot learning. However, this requires having a high-quality KB that contains Intents and ILIs couples, which is scarce. Second, rule-based methods such as HF or advanced NLP based on deep learning approaches can be used to validate the results of the LLMs.

C. Our Intent Decomposition and Translation framework

Fig. 2 depicts an instantiation of the architecture outlined in Fig. 1, focusing on deploying natural language Intents using the EURECOM 5G facility [1]. Two technological domains are considered: Cloud/Edge and RAN. Initially, a natural language Intent is processed by the MDIH, using an LLM to decompose it into Cloud/Edge Intent and RAN Intent. These are then forwarded to respective DIHs, also employing an LLM to translate domain-specific Intents into ILIs. In the Cloud/Edge domain, the ILI is represented by an NSD; in the RAN domain, it is a RAND. Although only the NSD JSON structure is illustrated in Fig. 2, the RAND is also implemented using JSON and configures RAN parameters such as slicing, throughput, maximum latency, bandwidth parts, etc. These configurations described by the RAN Intent pertain to non-real-time configuration, equivalent to the O1 configuration of Open-RAN (O-RAN)[2]. Subsequently, the NSD is sent to the Network Function Virtualization Orchestrator (NFVO), responsible for deploying network services within the 5G infrastructure. Simultaneously, the RAND is delivered to the RAN Orchestrator (RANO), which operates similarly to the Service Management Orchestration (SMO) of O-RAN, managing RAN service deployment within the 5G infrastructure. In this latter, the RAND configuration is enforced through the SMO/RAN Intelligent Controller (RIC) using rApps and xApps. To illustrate, Example I.1 will be decomposed into two Intents: First, the Cloud/Edge Intent would be: *"I want three applications: an augmented reality content server, a mixed reality collaboration platform, and a virtual reality simulation engine. Each application requires 4 vCPU and 2 Gbytes of memory."* Second, the RAN Intent would be: *"The clients are connected through a 5G network located in the Nice area and tolerate a maximum latency of 5ms."*. Then, the system will create: (i) an NSD containing three applications with the requirement in CPU and RAM for each one; and (ii) a RAND containing one xApp. This xApp will be deployed on top of the RIC by the RANO and will manage the radio resource allocation to ensure satisfying a 5ms latency for all users.

Fig. 3 illustrates the detailed LLM-based decomposition and translation framework. Both the MDIH and DIHs utilize the same LLM-based system design. This shared design employs a three-stage process to establish an efficient Intent-to-2-domain-Intents pipeline (for MDIH) and Intent-to-ILI pipeline (for DIHs). In Stage 1, historical examples are retrieved from the central decomposition KB: KB_d (for MDIH), or domain KBs: Cloud/Edge KB_c; and RAN KB_r (for DIHs). In Stage 2, these examples are utilized to decompose the Intent (for MDIH) or generate the ILI (for DIHs). Here, an LLM is employed through in-context learning. Finally, in Stage 3, users provide HF regarding the quality of the decomposition (for MDIH) or the generated ILI (for DIHs). This feedback is incorporated into the corresponding KB if validated by an administrator. Below, we discuss each stage in more detail. We will use the index $j \in \{d, c, r\}$ to represent the three systems.

1) *Few-shot examples extraction:* In this stage, the input, denoted as Q_{uj}, is processed by a sentence transformer model. The model's objective is to retrieve relevant examples from the corresponding KB_j where each entry is organized in a tuple structure as (Q_{ij}, A_{ij}). Here, Q_{ij} represents the queries corresponding to historical inputs, and A_{ij} comprises LLM responses, which have been validated either through manual insertion or HF.

[2]https://www.o-ran.org

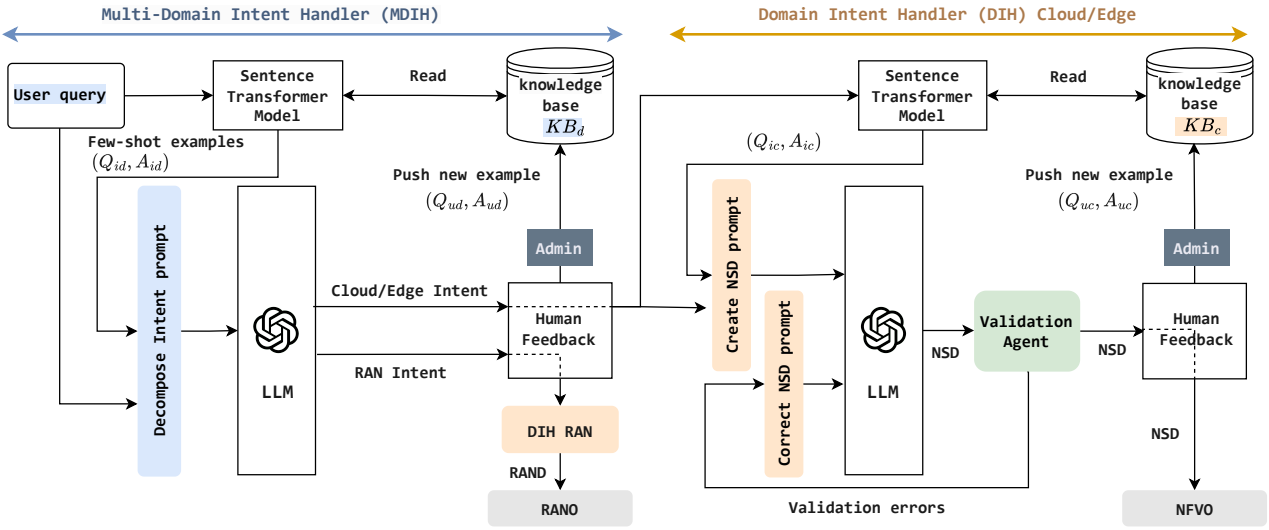

Fig. 3: LLM-based Intent decomposition and translation system.

The retrieval of examples involves measuring the cosine similarity between the new input query Q_{uj} and each query Q_{ij} within KB_j. The cosine similarity score indicates the degree of similarity between the vectors representing the queries. It is calculated as:

$$S(Q_{uj}, Q_{ij}) = \frac{Q_{uj} \cdot Q_{ij}}{\|Q_{uj}\| \|Q_{ij}\|} \quad (1)$$

where $\cdot$ represents the dot product of the query vectors, and $\| \cdot \|$ denotes the vector's Euclidean norm. Subsequently, the AI model extracts a set of n tuples, denoted as Top_n, representing the most similar historical examples. These tuples, in the form (Q_{ij}, A_{ij}), are then forwarded to the next stage for further processing.

2) *Intent decomposition or Intent translation:* The previously generated few-shot examples are assembled into a prompt, which serves as an input for the LLM. This prompt provides explicit instructions to guide the LLM in its task, with a clear directive stating either *"Your job is to decompose Intent"* or *"Your job is to generate the ILI."* Alongside this instruction, the prompt includes additional rules defined by the administrator, the few-shot examples (Q_{ij}, A_{ij}), and the new input Q_{uj}. Subsequently, the LLM will generate the decomposition or the ILI. However, it is essential to note that effective performance requires the LLM to have a substantial context window, particularly when dealing with a large number of examples (n is significant). Besides, the LLM should be pre-trained on code-related data, given its mission to translate Intent into JSON structures. In this context, we have selected the CodeLlama model [7]. A single ILI can comprise more than 2k tokens, and providing numerous few-shot examples necessitates a substantial LLM context window. CodeLlama models offer a maximum context window of 100k tokens, making them well suited for our use case.
3) *Intent translation validation:* In Intent translation, the validation agent ensures the LLM's output aligns with the ILI's structure through three steps: *syntax validation* for ILI JSON conformity, *semantic validation* using regular expressions to check parameter types and values, and *correlation validation* for parameter relationship consistency. It also confirms ILIs compatibility with EURECOM's NFVO/RANO APIs. Validated ILIs are forwarded to users via the GUI for HF; otherwise, the LLM receives a prompt to correct the NSD with specific instructions on detected errors.
4) *Human feedback:* Indirect Reinforcement Learning from HF (IRLHF) is a crucial component of our system. When a user is satisfied with either the Intent decomposition or the generated ILI, they can provide HF to the administrator for inclusion in the KB_j. Otherwise, the user can correct the LLM response and include the corrected version. This feedback, which consists of the query and the correct response (Q_{uj}, A_{uj}), is used to generate more accurate and complete Intent decompositions or ILIs in the future.

IV. Performance Evaluation

The section is structured into two subsections: *Experimentation setup*, which details the experimental setup, and *Experimentation results*, which presents and analyzes the performance of our framework.

A. Experimentation Setup

Our experimental setup consists of two machines, each equipped with 36 Intel(R) Xeon(R) Gold 6240R CPUs running at 2.40GHz. The second machine also has an Nvidia A100 GPU with 40GB of vRAM. The first machine runs the Kubernetes-based test cluster and the EURECOM 5G facility components, whereas the second machine hosts the LLM-based system. The same LLM is used for both Intent decomposition and translation. We use the LangChain[3]

[3]https://www.langchain.com

framework for handling LLMs and ChromaDB[4] to store KB embeddings. We gathered our foundational KBs data from a variety of sources, including EURECOM's past and ongoing research projects. We compared several other popular open-source LLMs, including Mistral 7B and Llama 13B, and found that the CodeLlama model with 34B parameters[5] produced the best results in both decomposition and translation tasks. We used the MPNet v2 sentence transformer model[6]. We set the maximum number of tokens to 3k, as the longest ILI in our initial KBs is 2k tokens long. We set n to 10 for the decomposition task and 4 for the translation task, as it is the maximum number of few shot examples that fit on the GPU. Additionally, we set the LLM temperature to 0.1. It is important to note that these parameters are flexible and can be adjusted to meet future requirements.

B. Experimentation Results

This subsection evaluates the system's performance in the context of Intent decomposition and translation. The evaluation focuses on decomposing natural language Intents into the two technological domains (Cloud/Edge, RAN) and subsequently translating Cloud/Edge Intents into NSDs. The evaluation of RAN Intent translation is omitted due to space constraints. However, since both the Cloud/Edge and RAN domains rely on NF-based components, the evaluation results from the Cloud/Edge domain can be generalized to the RAN domain. In order to gather performance data, feedback was solicited from 10 volunteering users within EURECOM. Each volunteer assessed our platform by creating 10 CSs and providing feedback on Intent decomposition and Cloud/Edge Intent translation.

1) Intent decomposition and translation feedback: To enable IBN, our system must understand users' Intents. We validated this understanding with a rating approach. After each experiment, volunteers evaluated both the Intent decomposition and translation steps using a scale of 0 to 5, with 5 being the highest score. Fig. 4 shows the average user rating from 10 CS creations. Using averages helps mitigate individual biases among users. If one or two users provide biased ratings, these are balanced out in the mean calculation. Our approach demonstrated significant efficiency in Intent decomposition, reflected by an initial rating score of about 4.5. However, users initially expressed dissatisfaction with the Intent translation component, requiring modifications before submitting NSDs to the NFVO. Despite this, the average rating consistently exceeded 3.5, indicating that only minor adjustments were necessary, primarily related to configuring unfamiliar parameters. Over time, the system learned from feedbacks and past examples, generating NSDs identical to those desired by users.

2) Intent decomposition and translation latency: Fig. 5 illustrates the relationship between the number of requested Cloud/Edge applications and the time required to decompose Intents and generate a valid and correct NSD using the LLM-based system. From this figure, we can observe that the

[4]https://www.trychroma.com/
[5]https://huggingface.co/TheBloke/CodeLlama-34B-Instruct-GPTQ
[6]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

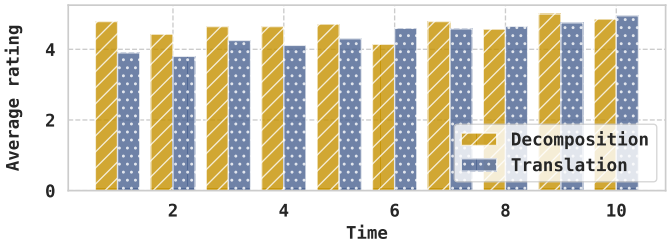

Fig. 4: Mean rating score throughout time.

Intent decomposition process adds only a few seconds to the overall E2E time (yellow surface). This is because the process involves breaking down the Intent into smaller parts, resulting in generating almost the same number of tokens in every decomposition task. However, as the number of requested applications increases, the translation time also increases (blue surface). As illustrated in Fig. 5, the E2E time exceeds 2 minutes when translating requests containing more than 3 Cloud/Edge applications. This is because the system generates more tokens for each Cloud/Edge application in the NSD, resulting in increased Intent translation time.

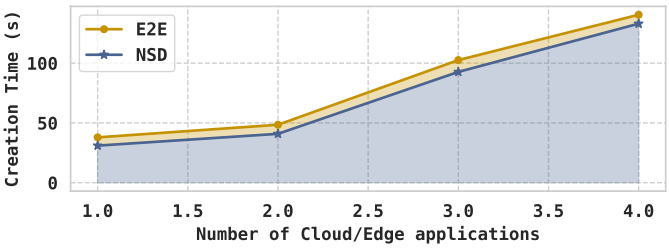

Fig. 5: Impact of the number of applications on decomposition and translation time.

3) Validation agent iterations: Throughout all NSD generations, the validation agent was executed only once for each generation. This demonstrates that the output of the CodeLlama LLM consistently adheres to the NSD structure from the first iteration, showcasing CodeLlama's effective learning of the format only from few-shot examples. This proficiency highlights its ability to generate correct JSON structures due to its training on code-related data.

V. Limitations and Future Work

Our work has successfully demonstrated the feasibility of automatic Intent decomposition and translation by combining few-shot learning using an open-source state-of-the-art LLM with HF. However, to achieve a robust E2E Intent LC management system, several areas of future development remain:

- *Intent activation and conflict resolution:* The LLM-generated ILIs may not align with available infrastructure resources. To address this, an ML-based feasibility check will be integrated into the DIHs, and an Intent negotiation mechanism will be developed to collaborate with users.
- *Trust activation:* Users should have the option to activate trust at a later stage, allowing them to submit requests

without viewing intermediate steps. This feature will be integrated into the system to enable fully automated Intent decomposition and translation, directly deploying applications from natural language-based Intents.

- *Structure validation enhancement:* HF is currently used to validate LLM outputs, which can be time-consuming. Therefore, advanced ML-based techniques, such as NER, will be explored to perform comprehensive structural and semantic validation of LLM outputs for decomposition and translation actions.
- *LLM improvements:* The current execution time is acceptable, but processing dense Intent requests can lead to prolonged processing and timeouts. Techniques to accelerate the inference process will be investigated. Additionally, in advanced development phases, a single small LLM will be trained to master Intent translation using the resulting KBs. This will enable the direct generation of ILIs from initial user Intent.

VI. Conclusion

This paper presented an innovative Intent LC management architecture that revolutionizes network configuration and management by enabling natural language interaction with networks. By leveraging cutting-edge advancements in AI, particularly LLMs, the proposed architecture automates the entire Intent LC, from Intent decomposition and translation to Intent negotiation, activation, and assurance. This approach significantly simplifies network management, eliminating the need for expertise in low-level configurations and enabling network administrators to focus on high-level network objectives. To validate the proposed solution's effectiveness, we presented an initial implementation and evaluated it in real-world deployments. The results demonstrate the architecture's capability to translate natural language Intents into actionable network configurations.

Acknowledgment

This work is supported by the European Union's Horizon Program under the 6G-Intense project (Grant No. 101139266).

References

[1] Sagar Arora et al. "A 5G Facility for Trialing and Testing Vertical Services and Applications". In: *IEEE Internet of Things Magazine* 5.4 (2022), pp. 150–155.

[2] Engin Zeydan and Yekta Turk. "Recent advances in intent-based networking: A survey". In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE. 2020, pp. 1–5.

[3] 3GPP Technical Specification Group Services and System Aspects. *Study on scenarios for Intent driven management services for mobile networks, Telecommunication management*. Tech. rep. 2019.

[4] ETSI. *Zero Touch Network and Service Management (ZSM) Means of Automation*. 2018.

[5] *TM Forum TMF921A - Intent Management API*.

[6] ETSI. *Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; Network Service Descriptor File Structure Specification*. ETSI Group Specification GS NFV-SOL 007. 2019.

[7] Baptiste Rozière et al. "Code Llama: Open Foundation Models for Code". In: *arXiv preprint arXiv:2308.12950* (2023).

[8] Celso H Cesila et al. "Chat-IBN-RASA: Building an Intent Translator for Packet-Optical Networks based on RASA". In: *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE. 2023, pp. 534–538.

[9] Yogesh Sharma et al. "Intent Negotiation Framework for Intent-Driven Service Management". In: *IEEE Communications Magazine* 61.6 (2023), pp. 73–79.

[10] Xiaoang Zheng and Aris Leivadeas. "Network assurance in intent-based networking data centers with machine learning techniques". In: *2021 17th International Conference on Network and Service Management (CNSM)*. IEEE. 2021, pp. 14–20.

[11] Luis Velasco et al. "End-to-end intent-based networking". In: *IEEE communications Magazine* 59.10 (2021), pp. 106–112.

[12] Jieyu Lin et al. "AppleSeed: Intent-Based Multi-Domain Infrastructure Management via Few-Shot Learning". In: *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE. 2023, pp. 539–544.

[13] Jingyu Wang et al. "Network Meets ChatGPT: Intent Autonomous Management, Control and Operation". In: *Journal of Communications and Information Networks* 8.3 (2023), pp. 239–255.

[14] Chao Feng, Xinyu Zhang, and Zichu Fei. "Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs". In: *arXiv preprint arXiv:2309.03118* (2023).

[15] Ali Maatouk et al. "Teleqna: A benchmark dataset to assess large language models telecommunications knowledge". In: *arXiv preprint arXiv:2310.15051* (2023).

BIOGRAPHIES

Abdelkader Mekrache is a PhD condidate at EURECOM's Communication Systems Department. His primary focus is advanced Network Management frameworks in next-generation wireless networks under the supervision of Prof. Adlen Ksentini. He is an active participant in collaborative research, and notably contributes to the OpenAirInterface (OAI) project. His research interests include Next-Generation Networking, 5G Core Network, Network Management & Orchestration, Artificial Intelligence, and Reasoning Algorithms for 5G networks and beyond.

Adlen Ksentini (Senior Member, IEEE) is a professor in the Communication Systems Department of EURECOM. He is leading the Network softwarization group activities related to Network softwarization, 5G/6G, and Edge Computing. Adlen Ksentini's research interests are Network Sofwerization and Network Cloudification, focusing on topics related to network virtualization, Software Defined Networking (SDN), and Edge Computing for 5G and 6G networks. He has been participating to several H2020 and Horizon Europe projects on 5G and beyond, such as 5G!Pagoda, 5GTransformer, 5G!Drones, MonB5G, ImagineB5G, 6GBricks, 6G-Intense, Sunrise-6G and AC3. He is the technical manager of 6G-Intense and AC3, on zero-touch management of 6G resources and applications, and Cloud Edge Continuum, respectively. He is interested in the system and architectural issues but also in algorithm problems related to those topics, using Markov Chains, Optimization algorithms, and Machine Learning (ML). Adlen Ksentini has given several tutorials in IEEE international conferences, IEEE Globecom 2015, IEEE CCNC 2017/2018/2023, IEEE ICC 2017, IEEE/IFIP IM 2017, IEEE School 2019. Adlen Ksentini is a member of the OAI board of directors, where he is in charge of OAI 5G Core Network and O-RAN management (O1, E2) for OAI RAN activities.

Christos verikoukis received the Ph.D. degree from Technical University of Catalonia (UPC), Barcelona, Spain, in 2000. He is currently an Associate Proferssor with the Univerisity of Patras and an Collaborating Faculty member with the ISI/ATH. He has authored 158 journal papers and over 200 conference papers. He is also a co-author of three books, 14 chapters in other books, and two patents. He has participated in more than 40 competitive projects, and has served as a project coordinator of several funded projects from the European Comission and of national projects in Greece and Spain. He is currently the IEEE ComSoc GITC vice-chair and the editor-in-chief of the IEEE Networking Letters.