Intent-Based 5G Core Deployment Using Generative AI

Dimitrios Brodimas
Dept. of ECE
University of Patras
Patras, Greece
0000-0002-7696-4418

Nikolaos Tzanis
Dept. of ECE
University of Patras
Patras, Greece
0000-0002-5219-1676

Alexios Birbas
Dept. of ECE
University of Patras
Patras, Greece
0000-0002-9468-7215

***Abstract*—The concept of Network-as-a-Service is based on the automatic deployment and dynamic reconfiguration of next-generation networks in order to meet the needs of the respective stakeholders. Towards that path, orchestrating both telco and computational resources across the different network domains, imposes a great challenge even for the network experts. This paper explores an intent-based management framework designed to simplify this procedure. By receiving high-level network requirements through descriptive text and supplementary images representing the desired outcome, the framework ingests and translates them into network configuration files. These files facilitate the seamless deployment of a open-source 5G Core, without any human intervention, tailored to the stakeholders' needs. The proposed framework achieves this by employing multimodal Generative AI models, particularly Large Language Models, to bridge the gap between the user's intent and network configuration.**

***Index Terms*—Intent-Based Networking, Large Language Models, Multimodal Generative Artificial Intelligence, Orchestration**

I. Introduction

As mobile networks become omnipresent, the need for customization and programmability grows, especially to meet the diverse needs of the various industries. Those specific needs are often dynamic and can shift over time, depending on the context. Hence, it is vital to offer Network-as-a-Service (NaaS) to provide the necessary abstraction for flexibility and adaptability in future networks. For the network operators this indicates that a fundamental enhancement of the existing orchestration platforms must take place to effectively deliver NaaS. These next-generation orchestrators must efficiently manage end-to-end deployments, optimize network management, energy consumption, and service delivery, while also ensuring digital inclusion for non-technical stakeholders.

This rapidly evolving landscape of mobile networks attracted many users with limited domain knowledge who seek advanced network services, but disregard the complexity of the underlying network implementation. This has heightened the need for user-friendly mechanisms to ingest their requirements, and effortlessly transform them into network configurations. Intent-Based Networking (IBN) was conceived to address this need. While the IBN concept has been around already for some time, the recent emergence of Large Language Models (LLMs) and Generative Artificial Intelligence (GenAI) has made this prospect feasible, by bringing yet a new era in network architecture and management.

To that extent, this work proposes a framework that capitalizes on the capabilities of cutting-edge AI tools, to deliver in the most seamless manner a customizable open-source 5G Core (5GC) deployment, introducing a major part of the NaaS concept. Specifically, this framework supports:

- incorporation of stakeholders' intent through a User Interface (UI)
- context distillation from high-level descriptions and images by employing Multimodal GenAI models
- configuration of an open-source 5GC based on the captured intent
- deployment of the 5GC on an actual edge-cloud infrastructure

The rest of the paper is structured as follows: Section II describes the related work regarding the 5GC, along with IBN. Section III presents the proposed framework in detail. In Section IV, the solution is validated through an indicative proof of concept (PoC). Finally, Section V comprises the conclusion and future work.

II. Related Work

A. Intent-Based Networking

Intent refers to a set of specific goals and desired outcomes that a network should achieve, expressed in a high-level language, without detailing on how these should be implemented. IBN focuses on creating networks that are easier to manage and operate, reducing the need for user intervention [1].

IBN has attracted considerable interest across various areas, including theoretical models, architectural designs, practical implementations, and the integration of AI [2]. Organizations like the Linux Foundation Networking (LFN) and the European Telecommunications Standards Institute (ETSI) are actively engaged in defining [3], standardizing [4], and developing intent-based components for their orchestration frameworks [5].

The different functionalities of IBN can be divided in two main clusters: Intent Fulfillment and Intent Assurance. The first one includes the processes of intent ingestion, translation, and orchestration with emphasis on providing the functions

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

and interfaces required for the intent to reach the network and be implemented (the work performed in this paper solely explores this aspect of IBN). In addition to the Fulfillment, Assurance is responsible for the functions and interfaces that allow users to check that the network indeed achieves the given objectives provided by the Fulfillment [2], [6].

In terms of intent ingestion and translation, various initiatives have taken place. An early implementation model was developed in [7], which mapped user-provided keywords to those usable by a software-defined network (SDN) controller. In another effort, the authors of [8] introduced an intent language that was converted into P4 templates for network configuration. Similarly, in [9], intents were translated into Network Service Descriptors (NSDs) using generative adversarial neural networks (GANs). More complex IBN use cases have emerged recently, as LLMs have grown in popularity. In [10], a GenAI model was used to offer network policies for the network to consume. However, to the best of our knowledge, there has not been any stable implementation specifically targeting customizable 5G Core deployment as a Service, over a diverse edge-cloud environment. This serves as the primary motivation behind this paper.

B. 5G Core Deployment

In alignment with the need for adaptable and scalable networks based on open standards, 3GPP [11] introduced the fifth generation (5G) of mobile networks. Those networks are based on the Service-Based Architecture (SBA), which is designed to be cloud-native and modular. It allows the core network functionalities, such as authentication, mobility management, and data plane operations, to be implemented as self-contained software components called Network Functions (NFs). These NFs can be deployed on commercial off-the-shelf (COTS) hardware within cloud environments without any requirement for dedicated hardware [12].

These standards are implemented by different advanced open-source projects, the most important of which are Open5GS [13] and free5GC [14]. Open5GS offers robust solutions for building and managing NR/LTE mobile networks up to 3GPP Release 16, while free5GC focuses on implementing the 5GC network as defined by 3GPP Release 15 [15] and beyond. Both solutions embrace the SBA and offer flexible deployments of the different NFs according to the requirements of the vertical applications, through standardized service-based interfaces (SBI) [16]. These implementations offer great flexibility but require deep knowledge of the SBA and SBI standards.

SD-Core [17], is another significant open-source project in this domain, that introduces a disaggregated mobile core optimized for public cloud deployment, seamlessly integrating with distributed edge clouds. It also leverages standard 3GPP interfaces, making it compatible with conventional mobile core deployments. This is feasible through the integration with Aether, Open Networking Foundation's 5G Connected Edge platform for private mobile connectivity and edge cloud services.

Nokia enhances these implementations by providing a comprehensive Core Software-aaS solution, as a network template, which includes pre-integrated NFs commissioned, operated, and maintained by the provider [18]. These pre-integrated NFs are configured and optimized for one or more use cases alleviating the end user from complicated low level network configurations. This service-oriented model simplifies integration, supports a wide range of use cases, and ensures continuous service, maintenance and upgrades through a single subscription, but is limited to the catalog of predefined configurations that can be ordered through documented APIs by the user.

III. PROPOSED FRAMEWORK

In this section, we introduce a deployment framework that leverages the potential of state-of-the-art genAI tools for intelligent deployment and orchestration of the 5GC, as depicted in Fig. 1. Within this proposed framework, a user initiates the flow by expressing his high-level business intent/description, along with a supplementary file, requesting a 5GC deployment. This supplementary file can be an image depicting how the different functions of the 5GC are spread across the infrastructure. The request is processed by the Intent Fulfillment Component (IFC), which parses the input for insights and then forwards it, along with the derived insights, to the LLM component for context analysis. Then the negotiation phase begins between the user and the LLM (through a UI) in order to fine-tune the deployment. When the user is satisfied with the setup, the framework creates a Contract containing all the needed information for the Deployment Agent to create an instance over the underlying infrastructure, divided in different Kubernetes clusters across the cloud continuum.

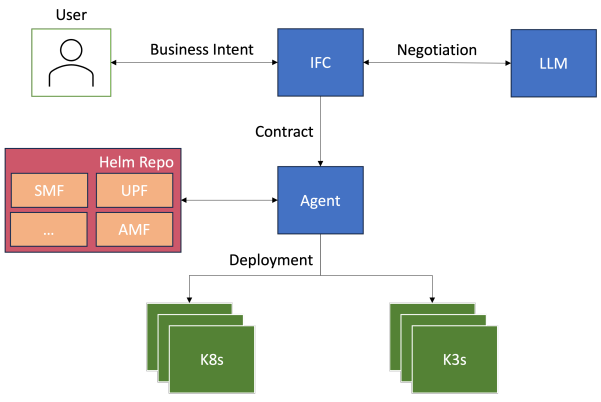

Fig. 1. Proposed framework's architecture

A. Intent Fulfillment Component

This element is burdened with the role of a facilitator within the framework by acting as the external access point for user interaction. As input, it receives a high-level description of the required network characteristics and/or a deployment-related image. The text input is collected through a chat-like

interface where the user communicates with the LLM in high-level language, until they agree upon the requirements and the available resources. This comprises the implementation of the negotiation phase. The input image is collected through a user friendly drug and drop zone, and then it gets transformed to base64 encoding in order to bring it to a suitable format for the LLM to consume. This new image is then bound by the LLM, providing the needed context.

Once the user is satisfied with the result of the negotiation, he presses the "Deploy" button in the UI, initiating the Contract Creation Phase. The outcome of this phase is a contract file in JSON format, which is forwarded to the Deployment Agent. It is also stored for management of the overall deployment life-cycle, e.g., modification or deletion of the 5GC artifacts, after the completion of the overall deployment.

To create the contract, the IFC parses the last two prompts of the negotiation phase, to extract all needed information regarding the deployment, such as the location of the 5G control plane functions and the user plane function (UPF), the number of UPFs, etc. These values are used to populate a predefined contract template, with the specified values for the different 5G control plane and UPF instances.

B. Large Language Model

GenAI, particularly LLMs, have gained significant interest in the field of natural language processing (NLP) due to their ability to generate coherent and meaningful text. These models are trained on vast datasets, which enables them to understand and produce human-like responses across a range of inquiries. Being built based on transformer deep learning techniques, LLMs have demonstrated high effectiveness in numerous NLP tasks. The transformer network serves as the central part of an LLM, consisting of multiple layers of self-attention mechanisms and feed-forward neural networks. When generating responses, the model utilizes the self-attention mechanism to determine the relevance of various words or tokens within a sequence. The input text for LLMs is typically tokenized into smaller units like words or subwords, with each token associated with a distinct numerical representation called an embedding. This method allows the model to analyze the text and identify relationships between tokens. During inference stage, the model processes a sequence of tokens as input and produces a probability distribution for possible subsequent tokens, ultimately choosing the token with the highest probability as its output.

In order to enhance the general pretrained model's knowledge with specific insights and context related to the networking domain, the few-shot learning [19] technique is employed, due to the limited and insufficient depth of the available datasets in this field. This transfer learning method leverages a minimal amount of information, including definitions, descriptions, examples, and behavior specifications (e.g., similar to a network engineer), as prompts [20]. These prompts are used to provide context and customize the models, without needing to format the model's entire parameter set. Moreover, this approach enables the integration of new tasks to the model without erasing or disrupting previous ones. As a result, it facilitates the model's ability to handle and respond to a wide variety of 5G deployments and orchestration goals.

C. Deployment Agent

The Deployment Agent is a straightforward automation engine that leverages the Helm Charts API [21] to deploy 5GC network functions on the specified Kubernetes clusters. Each 5GC function is packaged into separate Helm Charts to align with the SBA. The Agent processes the Contract provided by the IFC as input, configuring the necessary variables in the Helm Values files. Depending on the Contract's location field, the Agent selects the corresponding cluster configuration files and proceeds to deploy the NFs. To streamline the deployment and management of 5G instances, a common namespace is created across all involved clusters using the contract's unique ID. Once the deployment phase is completed, the Agent provides this unique ID to the user for future reference.

IV. PROOF OF CONCEPT

This section aims to present an end-to-end PoC of the described framework. Specifically, it leverages GenAI to transform an expression of high-level business requirements to an actual 5GC deployment. The testbed hosting this experiment and enabling the overall endeavor consists of:

- A server with 16CPUs , 16GB RAM, 1TB disk and one NVIDIA RTX A5000 to host the proposed framework,
- A K8s cluster with two VMs each one equiped with 4vCPUs, 8GB RAM and 80GBs disk, and
- Two K3s clusters, each of them comprised by two Raspberry Pis model 4b 8GB.

Both the IFC and the Deployment Agent are developed as containarized applications and they are deployed over Kubernetes on the server. As for the LLM, the framework leverages the Ollama API and three different models: GPT-3.5-turbo, Mistral-7b and Bakllava-7b. The GPT-3.5-turbo model is reached via an HTTPS remote internet connection, while the other two models are locally deployed. The framework's generated outcome is fed to the previously mentioned Kubernetes clusters of both conventional K8s and K3s distributions, representing the cloud (location represented by Athens) and the edge sites (location represented by Patra and Ioannina) of the continuum respectively.

Open5Gs is selected as the open source 5GC to support the experiment. In terms of radio access network (RAN) equipment to test to overall deployment, UERANSIM [22] is employed to simulate the functionalities of both the gNBs and the UEs.

A. Training Phase

In order for the LLM to be able to answer to the user's request adequately, some context must be provided. This is performed through the few-shot learning process explained in the previous section. Some of the most useful prompts

TABLE I
PROMPT EXAMPLES

5G definition
You serve as a 5G network assistant. You can deploy, modify and delete 5G Core instances. The User can specify where any of the 5GC NFs will be deployed. The default location of all the NFs is on the cloud.
Continuum definition
There are three clusters available. The clusters are located in Athens, Patras and Ioannina. The cluster in Athens is considered the cloud. The clusters in Patras and Ioannina are considered the edge.
Functionalities
Local breakout means UPF at a defined location. Two slices request means two UPF with different APNs.

provided to the LLM at the beginning of each session can be found in Table I.

These prompts are divided into three major categories to make it easier for the LLM to understand the context. The first one explains the role and the basic 5G insights to the LLM, the second one describes the underlying computing infrastructure, while the third one provides the context of the additional functionalities of the 5G system that may influence the deployment procedure.

B. Negotiation Phase

During the first step of this phase, the user selects from a drop-down menu the LLM that will be deployed to assist. If at any stage of the negotiation phase a complementary image is uploaded to accompany the text input, such as the one in Fig. 2, then the LLM locks to Bakllava model, as it is the model selected to serve multimodal inputs.

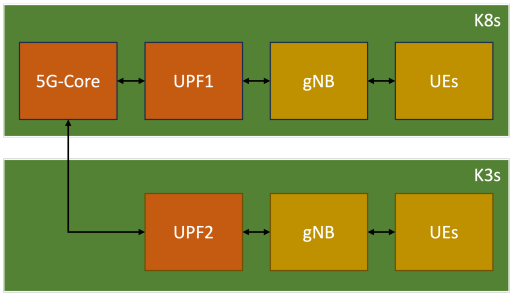

Fig. 2. Deployment overview

Then, the user provides the high-level business intent to the IFC which then proceeds on passing it to the LLM. The LLM processes the input and responds based on the context provided in the Training Phase. The messages' exchange keeps on until the user is satisfied with the provided 5G deployment. Then he presses the deploy button and the IFC moves on to the Contract Creation Phase. The produced Contract is then forwarded the to the Deployment Agent to implement and fulfill the intent. A clear illustration of this interaction can be found in Fig. 3.

In terms of performance of the LLMs deployed for the experimentation, GPT-3.5-turbo demonstrated superior context awareness and response quality across both text and image inputs, largely due to its vast number of parameters. However, its limitation on token usage per session imposed a constraint on the overall endeavor, leading to a shift towards local models. Mistral-7b delivered the second-best results in text-only scenarios but lacked the capability to process images. Bakllava-7b, while providing the lowest answer quality, supplemented its responses with context derived from the image inputs. Both Mistral-7b and Bakllava-7b were chosen in their medium-sized versions, balancing between performance and required resources.

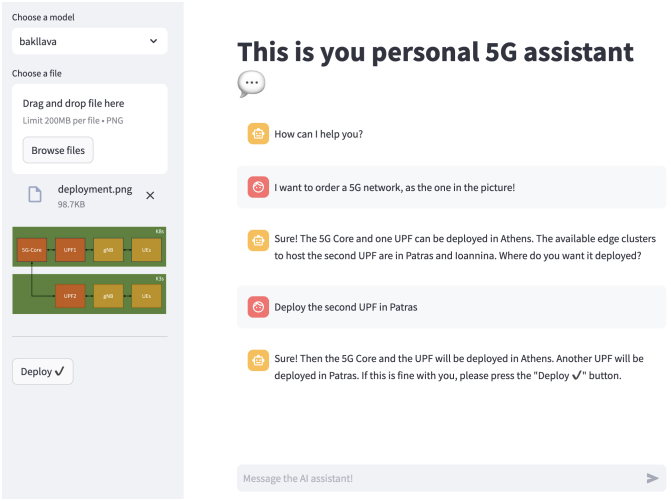

Fig. 3. Negotiation Phase

C. Contract Fulfillment Phase

In this final stage, the Deployment Agent captures the Contract and iteratively parses it to extract all the included information. The Agent deploys the UPF instances first as their services IPs need to be populated to the session management function (SMF) to connect them with the rest of the 5GC and handle them. The location indicates which kubeconf file has to be used, while the "contract_UUID" is used to create a specific namespace dedicated for the specific Contract. The access point name (APN) is a configuration parameter indicating early slicing support.

```
{ "contract_UUID": "77",
    "UPF": [
    { "name": "UPF_1",
      "location": "Patras",
      "APN": "internet"
    },
    { "name": "UPF_2",
      "location": "Athens",
      "APN": "ims"
    }
  ],
  "SMF": [
    {"name": "SMF",
      "location": "Athens",
      "APN": ["internet","ims"]
    }
  ],
```

```
  "AMF": [...]
}
```

In a similar way, for the rest of the 5GC deployment a namespace with the Contract's unique_ID is created, the SMF function is configured to include all the UPF related information and the serving APNs. When the 5GC is successfully deployed the external service IP of the access and mobility management function (AMF) is collected in order to be provided to the simulated gNBs to connect with the 5GC. Finally, to validate the produced outcome, the gNBs and the UEs are manually deployed, using the UERANSIM Helm Charts.

```
apel@apel-server1:~$ kubectl get pods -n contract-77 -o wide
NAME                                READY   STATUS    RESTARTS       AGE    IP             NODE
S
open5gs-amf-7556cc7f66-zpsjd        1/1     Running   0              114m   10.20.126.44   apel-server1
open5gs-ausf-57788544b-6lz2r        1/1     Running   0              114m   10.20.126.51   apel-server1
open5gs-bsf-745cb86bcb-ww5fv        1/1     Running   0              114m   10.20.126.47   apel-server1
open5gs-mongodb-59589b5b66-pm6hs    1/1     Running   0              114m   10.20.126.52   apel-server1
open5gs-nrf-7669b557fb-jkh2x        1/1     Running   0              114m   10.20.222.81   apel-server2
open5gs-nssf-cf6cc6f86-pcc7j        1/1     Running   0              114m   10.20.126.46   apel-server1
open5gs-pcf-f588f5f88-l5mtj         1/1     Running   2 (114m ago)   114m   10.20.222.80   apel-server2
open5gs-populate-74dddd57fd-ng8d8   1/1     Running   0              114m   10.20.222.77   apel-server2
open5gs-scp-79c67d4699-6t4lb        1/1     Running   0              114m   10.20.222.66   apel-server2
open5gs-smf-6d69446f55-rtxqw        1/1     Running   0              114m   10.20.222.76   apel-server2
open5gs-udm-7c46f6bf56-rt4bk        1/1     Running   0              114m   10.20.126.48   apel-server1
open5gs-udr-8549ff85f-6hcjl         1/1     Running   2 (114m ago)   114m   10.20.222.74   apel-server2
open5gs-upf-6c7f7659f6-lqhfc        1/1     Running   0              114m   10.20.126.45   apel-server1
open5gs-webui-58b49ffd6b-kckf9      1/1     Running   0              114m   10.20.126.50   apel-server1
ueransim-gnb-84b5886896-dk2dh       1/1     Running   0              112m   10.20.126.60   apel-server1
ueransim-gnb-ues-ff594cb8b-667tr    1/1     Running   2 (112m ago)   112m   10.20.126.57   apel-server1
```

Fig. 4. Pods deployed at the Cloud

The JSON file shown in Fig. IV-C is a part of the Contract generated by the IFC component after the negotiation phase. This Contract was then utilized during the actual deployment process, which was coordinated by the Deployment Agent. The complete set of the requested pods for the 5GC, gNBs, and UEs can be seen in Fig. 4 and Fig. 5, accessed through the respective clusters using the Kubernetes API.

```
ubuntu@k3s-test-1:~$ kubectl get pods -n contract-77 -o wide
NAME                                READY   STATUS    RESTARTS      AGE   IP           NODE
open5gs-upf-54fcc876d8-9mtc6        1/1     Running   0             16m   20.10.1.16   k3s-test-2
ueransim-gnb-cdf9649df-8rxfv        1/1     Running   0             15m   20.10.1.21   k3s-test-2
ueransim-gnb-ues-6c55cf58d6-6wlv6   1/1     Running   2 (15m ago)   15m   20.10.0.19   k3s-test-1
```

Fig. 5. Pods deployed at the Edge

V. Conclusion

In this paper, we introduced an intent-based framework capable of deploying a 5GC across the edge-cloud continuum. This framework utilizes a chat-like interface, allowing users to iteratively express their intent by communicating with one of three available LLMs supporting multimodal input. This procedure results in the creation of a deployment Contract, which is then handled by the automation engine enforcing it on the underlying infrastructure. The proposed solution was validated using open source tools (Open5GS, Kubernetes, Ollama) widely adopted by the scientific community, but can be easily adapted for use with other open-source or proprietary solutions.

The tangible network deployment provided as an outcome by the overall framework confirmed that user requirements, expressed in a high-level and intuitive manner, were accurately translated into network operations, bridging the gap between users' intent and the network configuration.

Future work identified by the authors includes testing and validation of the proposed framework with actual RAN equipment and UEs. Further improvements could focus on enhancing model training, shifting from few-shot learning to techniques like retrieval-augmented generation (RAG) or model fine-tuning, to improve the framework's ability to adapt to user intent in more complex scenarios, supporting advanced configurations and network setups.

Acknowledgment

This work has been partially supported by the "Horizon Europe" Project: P2CODE (grant agreement No. 101093069).

References

[1] C. Li, O. Havel, A. Olariu, P. Martinez-Julia, J. Nobre, and D. Lopez, "Rfc 9316: Intent classification," USA, 2022.
[2] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, "A survey on intent-driven networks," *IEEE Access*, vol. 8, pp. 22 862–22 873, 2020.
[3] ETSI GS ENI, "System Architecture," 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ENI/001_099/005/03.01.01_60/gs_eni005v030101p.pdf
[4] ETSI GR ZSM, "Intent-driven autonomous networks; Generic aspects," 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/011/01.01.01_60/gr_ZSM011v010101p.pdf
[5] ONAP, "Support for Intent Framework and Intent Modeling," 2021. [Online]. Available: https://wiki.onap.org/display/DW/Support+for+Intent+Framework+and+Intent+Modeling
[6] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura, "Rfc 9315: Intent-based networking - concepts and definitions," USA, 2022.
[7] Y. Han, J. Li, D. Hoang, J.-H. Yoo, and J. W.-K. Hong, "An intent-based network virtualization platform for SDN," in *2016 12th International Conference on Network and Service Management (CNSM)*, 2016, pp. 353–358.
[8] M. Riftadi and F. Kuipers, "P4I/O: Intent-Based Networking with P4," in *2019 IEEE Conference on Network Softwarization (NetSoft)*, 2019, pp. 438–443.
[9] K. Abbas, M. Afaq, T. A. Khan, A. Mehmood, and W.-C. Song, "IBNSlicing: Intent-Based Network Slicing Framework for 5G Networks using Deep Learning," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2020, pp. 19–24.
[10] K. Dzeparoska, J. Lin, A. Tizghadam, and A. Leon-Garcia, "LLM-Based Policy Generation for Intent-Based Management of Applications," pp. 1–7, 2023.
[11] 3GPP, "3GPPP." [Online]. Available: https://www.3gpp.org/
[12] L. Morand, "OpenAPIs for the Service-Based Architecture," May 2022. [Online]. Available: https://www.3gpp.org/technologies/openapis-for-the-service-based-architecture
[13] "Open5GS." [Online]. Available: https://open5gs.org/
[14] "free5GC." [Online]. Available: https://free5gc.org/
[15] L. Morand, "3GPPP Release 15," April 2019. [Online]. Available: https://www.3gpp.org/specifications-technologies/releases/release-15
[16] "ETSI TS 129 500; 5G System; Technical Realization of Service Based Architecture," August 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/129500_129599/129500/-18.06.01_60/ts_129500v180601p.pdf
[17] "SD-Core." [Online]. Available: https://opennetworking.org/sd-core/
[18] "Core network Software as a Service." [Online]. Available: https://www.nokia.com/networks/core/5g-core/core-saas/
[19] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, jun 2020. [Online]. Available: https://doi.org/10.1145/3386252
[20] Y. Zhu, Y. Wang, J. Qiang, and X. Wu, "Prompt-Learning for Short Text Classification," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–13, 2023.
[21] "Helm Charts." [Online]. Available: https://helm.sh/
[22] A. Gungor, "UERANSIM." [Online]. Available: https://github.com/-aligungr/UERANSIM