Computer Science and Information Systems 21(1):395–418 https://doi.org/10.2298/CSIS230213002B

VINIA: Voice-Enabled Intent-Based Networking for Industrial Automation⋆

Raul Barbosa[1,2], João Fonseca[1,2], Marco Araújo[1], and Daniel Corujo[2]

[1] Capgemini Engineering, Porto, Portugal
[2] Universidade de Aveiro and Instituto de Telecomunicações, Aveiro, Portugal
dcorujo@av.it.pt

Abstract. Intent Based Networking (IBN) is a promising approach for automating and managing large and complex networks. Integrating Voice-enabled Virtual Assistants (VVAs) with IBN and Software Defined Networking (SDN) has improved network management efficiency and flexibility. However, there is still room for optimization improvement in installing intents in industrial scenarios. Network Orchestration Automation plays an important role within the Beyond 5G and 6G Networks, considering existing practices for orchestrating 5G Network Functions. This work presents an extended preliminary architecture for a voice-enabled IBN system called VINIA for industrial network automation. The new approach allows the configuration of more network assets (e.g., 5G networks), leveraging Network Orchestrators and Network Slice Managers, thus improving the system's capabilities. The results provide insights into this solution's potential benefits and limitations to enhance the automation of industrial networks' management and orchestration procedures.

Keywords: Intent, Intent-driven Management Service, Network Slicing, Industrial Virtual Voice Assistant

1. Introduction

A novel method for managing networks called Intent Based Networking (IBN) is gaining popularity due to its ability to automate and streamline the setup and maintenance of complex networks. IBN is especially well-suited for industrial scenarios with high dependability, security, and real-time performance demands. It is a desirable solution when traditional network management techniques can be time-consuming, error-prone, and challenging to scale [1]. One of the critical components of IBN are the Network Orchestrators (NOs), software systems that automate the configuration and management of devices based on high-level business objectives and policies [2].

Integrating Voice-enabled Virtual Assistants (VVAs) provides enhanced capabilities to IBN-based systems in industrial scenarios. VVAs allow skillful network administrators

⋆ This work is an extension of the paper presented in the INISTA 2022 conference: R. Barbosa and M. Araujo, "AI-driven Human-centric Control Interfaces for Industry 4.0 with Role-based Access",2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Biarritz, France, 2022, pp. 1-6, doi: 10.1109/INISTA55318.2022.9894247 and not-so-familiar users to interact with the network using speech, providing a more intuitive and user-friendly interface for managing the network. VVA can enhance overall efficiency and reduce human error in network management. Studies show the importance of adopting VVAs and IBN for Human-AI collaboration in Information and Communication Technology (ICT) supply chains [3, 4].

The integration of VVAs and IBN with Software Defined Networking (SDN) has been the subject of recent research in industrial network automation. Our previous work [5] explored the potential of using VVAs to simplify installing intents in industrial network scenarios. However, the work only considered the applications using SDN controllers as the primary network orchestration tool. In this extension paper, we aim to broaden the scope of our previous study by exploring the NOs/Network Slice Managers (NSMs) landscape, considering the requirements for 6G networks regarding Network Management and Orchestration Automation [6–9], and the current standard development organization's specifications and research. For that purpose, an analysis of the current NOs/NSMs is done, inferring the capability of these platforms to instantiate and control industrial networks more efficiently. As an outcome of this analysis, we present a new architecture for VINIA capable of efficiently orchestrating an industrial network through voice intents. This study is part of the work being developed in the project 6G BRAINS[3]: Bringing Reinforcement learning Into Radio Light Network for Massive Connections. Developing automation solutions for 6G Networks, leveraging AI/ML, and using Intent-based management by design is an important project requirement [10].

The rest of the document is structured as follows. In section 2, a study of relevant works on the topics covered in the paper is done. In section 3, we present the solution design given previously [5], extending the information about its implementation and results. Moreover, in this section, we draw some conclusions about state of the art and the drawbacks of the previous architecture to propose a more generic voice-enabled IBN system architecture that can interact with different types of SDN and Network Functions Virtualization Management and Orchestration (NFV MANO) systems. Finally, section 4 concludes and draws directions for future works.

2. Related Work

This section will review related work in speech recognition systems, IBN, and network management/orchestration to provide a foundation for our new architecture, considering current research and standardization efforts.

2.1. Speech recognition systems

Natural language access control policies can be unstructured and ambiguous; consequently, they cannot be directly implemented in an access control mechanism. In [11], a methodology is used to tackle this issue by applying linguistic analysis to parse natural language documents and annotate words to identify whether semantic arguments can be inferred from a given sentence. The authors claim that this methodology obtained results that can effectively identify access control policies with a precision of 79%.

---

[3] 6G BRAINS is an H2020 research and innovation action supported by the European Commission Horizon 2020 Programme under grant agreement number 101017226

Due to the massive policy scale and the number of access control entities in open distributed information systems available in Industry 4.0 systems, existing management decision engines for access authorization suffer from performance bottlenecks. Several solutions have been proposed to overcome bottlenecks. In [12], a permission decision engine scheme based on a random forest algorithm to construct a vector decision classifier is presented, for which the authors claim to have achieved a permission decision accuracy of around 92.6%. In [13], a method for improving the policy decision performance by eliminating conflicts is proposed, but with poor improvement in the performance. [14] offers a k-means clustering on the access control policy set, concluding that the order of the policies in the policy set significantly impacts the permission decision efficiency. [15] proposes a permission decision optimization method based on two tree structures: match tree and combination tree. The match tree uses a binary search algorithm to search for the policy matching the access request rapidly. The combination tree evaluates the access request based on the matching procedure. In [16], the authors propose a methodology to generate sets of realistic synthetic natural language access control policies to overcome the constraints due to the lack of appropriate data.

The previous text discusses challenges and solutions related to natural language access control policies in access authorization for different information systems. It investigates methods that use linguistic analysis to understand and interpret natural language, highlighting the importance of efficient policy decision engines in Industry 4.0's open-distributed systems. Building upon this understanding, the following discussion introduces three distinct frameworks, `SUSI AI`[4], `SEPIA Framework` [5], and `Mycroft AI`[6], contributing to the advancement of VVAs. These technologies offer innovative solutions for enhancing user interaction through voice and natural language while enabling the customization and control of the assistant's functionalities.

`SUSI AI` is an open-source VVA capable of interacting with the user through voice, using a Application Programming Interface (API). This VVA allows us to add more features to give the user greater control, allowing us to add, edit and remove skills. This type of VVA supports Linux, Android, and iOS and can be integrated with speakers and vehicles. It also allows the transformation of the user's data into JSON format and manipulation according to the user's intention.

`SEPIA Framework` means server-based, extensible, personal, and intelligent assistant with a Java server and a client that can run on various Android, iOS, Windows, Linux, and Mac platforms. The server is based on the REST architecture, and the clients use the HTTP protocol to communicate. Understanding natural language, dialogue management, and intention is done on the server. The client handles speech recognition and converts the voice into text, sending it to a SEPIA server to interpret and present the result to the user through text, which can be in JSON. Like the previously spoken technologies, SEPIA and the services already implemented allow us to create our commands.

The most relevant of the VVAs studied was `Mycroft AI`. This open-source VVA allows modifying, creating, and viewing code, enabling users to control the system. This type of VVA can be found on various systems, from Raspberry Pi, Windows, Android, and Mac. Besides these platforms, there are dedicated devices, Mark 1 and Mark 2. `Mycroft`

[4] SUSI AI.[online] Available at: https://github.com/fossasia/susi_server
[5] SEPIA Framework.[online] Available at: https://sepia-framework.github.io/
[6] Mycroft AI. [Online]. Available: https://mycroft.ai/

works by intents, once awakened, the user expresses the intention to the system, and it tries to interpret the intention and find the appropriate Skill. These abilities can be installed or removed by users and can be easily updated to expand functionality. In addition to these advantages, `Mycroft` has the particularity of transforming the user's request into JSON format, which is useful for implementing it on the system.

2.2. Intent Based Networking Management

An intent is a type of policy that expresses objectives without mentioning how they are implemented and can be considered [17]: *Portable* as it can be moved between the different controller and network implementations and remain valid; *Abstract* as it must not contain any details of a specific network.

Intents are like a set of rules and services that define the criteria for access and resource use. Each rule is composed of a set of conditions and actions. The first set defines when the rule is applicable, and as soon as it is active, it generates a set of actions to be implemented. Policies Based Networking Management works as a manager that separates the rules that control the system from its functionality. As a result, this system reduces maintenance costs and improves run-time flexibility. The research currently being performed into Beyond 5G/6G Networks validates the interest in this concept [18–20, 7]. We have focused on studying the application of intents in two different technologies: SDN and NFV MANO.

Intent-Based Networking Management in SDN Research work by the author Vijay Varadharajan [21] explores an architecture based on policies in SDN context using the ONOS controller. This architecture is run on the SDN controller as an application. The policy servers connected to the respective controller have five main components: Repositories, Policy Managers, Policy Evaluation Engine, Policy Enforcer, and Handle Creator. In Machado's work [22], the author introduces a policy authorization framework for SDN using a high-level language. This study focuses on policies targetting Quality of Service (QoS). By interpreting Service Level Agreements (SLAs), the system extracts Service Level Objectives (SLOs), which are considered requirements for assessing the network's performance. Douglas Comer and Adib Rastegarnia presented a framework, Open Software Defined Framework (OSDF) [23], which offers a high-level API that allows network users to configure and monitor the network to provide QoS. In addition, OSDF has mechanisms to analyze conflicts between policies to prevent two or more policies from conflicting when applied to the same targets. This framework's architecture consists of four components: Policy Storage Module, Policy Conflict Detection Module, Policy Parser Module, and High-level network operation services. The main focus of this study was the analysis of conflicts since it is essential to maintain the excellent performance of the networks.

Intent-Based Networking Management in ETSI MANO Platforms This work focuses on transitioning from a standalone SDN solution implemented in the previous work, augmenting it with capabilities to control End-to-End (E2E) NOs like ONAP[7] or OSM[8].

[7] ONAP.[online] Available at: https://www.onap.org/

[8] OSM.[online] Available at:https://osm.etsi.org/

These orchestrators bring several benefits to organizations managing complex networks: Improved Scalability, Enhanced Automation, Centralized Management, Increased Flexibility, and Improved Security.

ONAP supports Intents and can perform operations for quickly deploying an E2E Network that entails 5G Core and RAN Network Functions. Moreover, the platform allows for automatic operations through the closed-loop automation associated with efficient monitoring of resources. However, the complexity of this orchestrator limits its use, especially if the required computing resources aren't available.

OSM, an Open Source alternative developed by the European Telecommunications Standards Institute (ETSI), isn't capable of the same features, as it is a less complex/able NSM. However, since it is also a NO compliant with ETSI MANO Architecture, its adoption in European-funded projects (mainly Horizon programmes) is relevant. Several solutions build on top of OSM to automate the Network Management and Orchestration procedures [20, 24–27] due to its simplicity and low computing resources requirement. Moreover, in the ETSI OSM-hosted Hackfests Ecosystem Days, several demos of work unpublished/unreferenced in the literature are performed [9].

Telecom organizations can increase their networks' effectiveness and dependability while lowering operational expenses and enhancing security, augmenting existing SDN solutions with up-to-date NOs. NOs can deal with the requirements of contemporary networks, whether you are in charge of a sizable enterprise network or a network that supports vital infrastructure. The trend moves to more autonomous networks also brought new requirements to NOs, like the ability to manage E2E Network Slices. This led existent NOs to gain Network Slice Management capabilities [28, 29].

Standard Development Organizations Specifications and Research Automating procedures related to communication service configuration has been a research topic for several years. The need for specialized people to configure the network stacks has become a point of failure for both Communication Service Providers (CSPs) and Network Operators (NOPs) [30–32]. This has led the Telecommunication Industry to devise solutions to this problem. The 3rd Generation Partnership Project (3GPP) has been developing some research on Intent driven Management Service (Intent driven MnS) for mobile networks in the scope of the Technical Report (TR) 28.812 [33]. This effort, part of 3GPP release 17, offers an overview of the business requirements for adopting intents at different roles of the Telecom Vertical Industry. The document aligns with the recent works developed in the area of 5G, which capitalize on the concept of Network Slicing, a characteristic and a requirement of the architecture of 5G networks. The concept of management of Network Slices is one of the key points that mostly influences the TR. Specifically, 3GPP establishes that one or more Network Slices can deliver each Communication Service (CS). Starting from this concept, the definition for an Intent driven MnS specifies that it is a Service whose capabilities can be defined and managed via an Intent. 3GPP specified the consumers of the Intent as the CSP or Communication Service Consumer (CSC) building on top of the Network Slicing as a Service (NSaaS) concept defined in [34], allowing the CSP to provide management services to the CSC. This new paradigm part defined in 3GPP release 15 allows managing the resources associated with that slice. In the case of

[9] As an example, in the OSM-MR10, the author of [24] presented a Demo called Vertical's intent evolution at service runtime driving vCDN automated scaling

a Intent driven MnS, the CSC needs to know a service's characteristics without knowing the Network Slice's components. The 3GPP also specifies how CSP and NOPs can also leverage the use of Intents for the management of the Network Slices/Infrastructure in their domains. 3GPP's TR 28.812 also presents how closed-loop mechanisms can be used to automate tasks leveraging the Intent for effortless control of the CS to be provided to the CSC. It's important to reiterate that an intent specifies the CSC's objective, detailing the expected characteristics.

While the 3GPP has focused mainly on research that builds on the most recent advancements/needs of mobile networks, other Standard Development Organizations (SDOs) proposed using Intent for other use cases. TeleManagement Forum (TM Forum) introduced Intents in the scope of the Autonomous Networks project, which has created documents that detail the technical and business characteristics of an Autonomous Network [35–38]. The Introductory Guides of this organization explain the importance of this technology to different Vertical Industries [35, 39]. These documents facilitate understanding of the concept by people unfamiliar with network management and orchestration.

Other relevant research related to IBN can be visualized in Table 1.

Table 1. Description of IBN architecture scope and related works

Platforms	Functions / Features	References
SDN	Service model and orchestration	[40],[41],[42]
	Monitoring and resource exposure	[43], [44]
	Intent deployment and configuration	[45]
	Network orchestration	[46]
NFV	Service model and orchestration	[47]
	Intent deployment and configuration	[45]
	Network orchestration	[48]

2.3. Summary

While existing speech recognition systems and intent-based networking management approaches have made significant strides, notable gaps and limitations still require attention.

One major limitation in speech recognition systems is handling highly unstructured and ambiguous natural language access control policies. While methods like linguistic analysis have been proposed to parse and annotate words in natural language documents, achieving high precision in identifying access control policies remains challenging. The existing decision engines for access authorization may suffer from performance bottlenecks due to the massive policy scale and the number of access control entities in open distributed information systems. There's room for innovation focused on the bottlenecks, improving decision accuracy, and performance enhancement.

Regarding IBN management, existing implementations in SDN and Network Functions Virtualization (NFV) platforms have shown promise but face limitations. In SDN, while architectures based on policies have been explored, analyzing conflicts between policies remains a critical challenge to maintaining network performance. Furthermore, SDN solutions implemented with specific controllers may lack the scalability and flexibility needed to handle complex networks effectively. However, NFV platforms and NSMs have different capabilities and complexities, leading to trade-offs in their adoption. ONAP supports Intents and can automate E2E network deployment, but its complexity may hinder its use, particularly in resource-constrained environments. OSM is more suitable for specific scenarios due to its simplicity and lower computing resource requirements, making it the NSM of choice in EU-funded research projects. However, its capabilities may not match those of ONAP.

The heterogeneity of solutions and the significant gaps and limitations analyzed led us to our architecture proposal. Using a similar approach to the one proposed by Sli-MANO [20], where it interfaces with NFV MANO platforms like ONAP and OSM to enhance scalability, automation, and centralized management while enabling Intent-Driven Mobile Network Slices. Moreover, running on top of both orchestrators creates a more efficient, flexible, and effective network management system that can handle the requirements of contemporary networks, including those of 6G Networks. VINIA architecture's focus on Intent-Driven Mobile Network Slices aligns with the ongoing research efforts in 3GPP, emphasizing the importance of managing Network Slices in meeting specific Communication Service requirements. Additionally, the architecture aims to overcome the trade-offs between different approaches by integrating diverse capabilities from both ONAP and OSM, offering telecom organizations more options for effectively managing their networks while considering resource constraints and complexity. Combining VVAs with Intent-Based Networking in the proposed architecture also simplifies and facilitates intent installation, further enhancing network management.

3. System Design

This chapter will present the VINIA system, detailing the characteristics and respective descriptions, system workflow, and architectures. The system is divided into two components; the first is responsible for identifying the user, and the second implements the user's intention after the permissions validation.

3.1. Speech recognition system

This section outlines the design and characteristics of the system for successful development. The system employs pre-trained voice recognition models to identify users. User recognition is divided into training and test phases, differing mainly in the data processing step. The training phase involves data sent to modeling algorithms for training, while the test phase employs an already trained model for data classification. To provide an in-depth understanding of both phases and highlight their shared components, Figure 1 illustrates a detailed workflow.

1. **Data collection**-This is the initial process of collecting user information, namely the number of speeches they have and a list corresponding to the speeches' directories.
2. **Processing phase**-After loading the speech file, four approaches are applied: a) Trimming; b) Optionally, and if necessary, Data Augmentation; c) Feature Extraction; d)

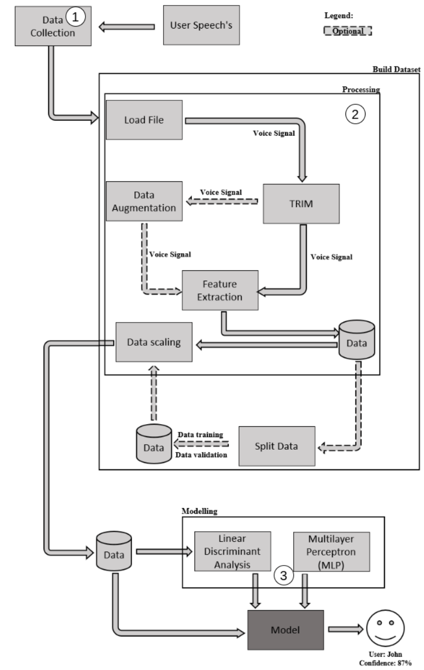


Fig. 1. VINIA's Speaker Recognition System: Workflow

Depending on the used algorithm, Multilayer Perceptron (MLP) or Linear Discriminant Analysis (LDA), optionally, we can split the Dataset in two: Data Training and Data Validation

3. **Modelling or testing**-We can follow two approaches, one for the modeling where the model is trained for future identification of users or for the pre-trained model where the data will be classified, that is, training and testing phases, respectively.

Regarding data construction (processing), we apply Trim, Data Augmentation, Feature Extraction, Data Scaling, and Split Data. We use Neural Network- MLP and LDA for modeling. Our choice of employing the MLP and LDA as our primary modeling techniques is based on their ample amount of work employing such techniques [49–53]. Our selection of MLP for modeling plays a pivotal role in our approach for several compelling reasons:

– It's a universal approximator, meaning it can effectively capture complex relationships within our data. This is of paramount importance when dealing with diverse and intricate audio data;
– In audio processing, extracting salient features may not be obvious to the human ear but can be efficiently identified by the model. The MLP empowers our system to uncover and use these data patterns;
– Complex tasks such as audio-based user identification using deep neural networks, including MLPs with multiple layers, require this method;

While the MLP constitutes the primary workhorse of our modeling approach, we have also complemented it with LDA for the following reasons:

- Discriminative Analysis: LDA is a proven technique for enhancing the separability of classes in data. This is particularly useful in user identification, where distinguishing between different users is a core objective;
- Dimensionality Reduction: LDA provides an avenue for dimensionality reduction, crucial in simplifying complex data while retaining essential discriminative information;

In conclusion, combining the MLP and LDA offers a balanced and comprehensive modeling approach. Together, these models qualify our system to effectively construct and process data for robust user identification, ensuring the accurate recognition of users even from complex audio data.

In the context of speech signal processing, it is crucial to discuss several key aspects involved in data construction, as these components lay the foundation for successful user identification models. These aspects include data trimming, data augmentation, feature extraction, and data scaling. Each plays a unique role in preparing the data for model training, ensuring that the system can effectively handle a wide range of scenarios. The data trimming process is designed to remove silent portions of an audio signal, particularly segments without speech. By eliminating these silent intervals, the overall duration of the speech signal is impacted. The method involves setting a threshold to identify silence zones, therefore segmenting the audio where the sound falls below 20 decibels. These identified areas are then labeled as silent zones, indicating the absence of speech. In machine learning, insufficient data for training models is a prevalent issue, especially when dealing with small datasets. To address this challenge, various data augmentation techniques can be employed, tailored to the specific nature of the data at hand. Data augmentation involves generating additional data artificially from the existing dataset. It is an optional process that is selectively employed based on the specific data requirements. When working with voice signals, which is our focus in this study, data augmentation techniques aim to create additional voice samples that expand the dataset. Common approaches for voice data augmentation include injecting noise and modifying pitch and speed. However, due to the context of voice recognition and the need to preserve the essential voice characteristics, the primary technique used here is noise injection. In this process, the original voice signal is analyzed to calculate its speech duration. Subsequently, artificial noise is generated and integrated into the original signal, effectively duplicating the data. This augmentation technique enhances the robustness and generalization of machine learning models trained on limited voice data. Feature extraction plays a pivotal role in voice signal processing, enabling the extraction of essential characteristics from the voice data. Features, in the context of voice signals, represent distinctive attributes that are crucial for voice analysis. These features can be broadly categorized into two types: temporal (time-domain features) and spectral (frequency-based features). Temporal features, which fall under the time-domain category, are relatively straightforward to extract and have direct physical interpretations. They include signal energy, zero-crossing rate, maximum amplitude, and minimum energy. These features provide insights into the temporal aspects of the voice signal and are important for tasks such as pitch and rhythm analysis. On the other hand, spectral features involve transforming the time-based voice signal into the frequency domain using mathematical techniques like the Fourier Transform. Spectral features encompass a range of attributes, including fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, and spectral roll-off. These features are invaluable for analyzing the frequency content of voice signals, contributing to tasks related to note recognition, pitch analysis, rhythm, and melody extraction. In our feature extraction process, we employ a comprehensive set of methods that cover both temporal and spectral domains. This diverse set of techniques allows us to capture various voice signal characteristics. Our selection includes Root-Mean-Square and Zero Crossing Rate for temporal analysis, which provides insights into the signal's energy and temporal dynamics. On the spectral side, we leverage various techniques such as Chroma features, Mel-Frequency Cepstral Coefficients (MFCC), Spectral Centroid, Spectral Contrast, and Spectral Rolloff. These spectral features enable us to delve into the frequency domain aspects of the voice signals, uncovering vital information about the voice, including pitch, tone, and spectral content. In the final step of our data processing pipeline, data scaling is applied to prepare the extracted features for model training. Data scaling aims to ensure that the feature values are transformed and exhibit consistent statistical properties. We employ the Standard Scaling approach, standardizing the data's distribution to have a mean value of zero and a standard deviation of one. This step aids in optimizing the performance and convergence of machine learning models, ensuring that features with different scales do not unduly influence the learning process. All these steps were validated in the paper that this journal extends [5]. All this voice processing and model creation is integrated into an API Representational State Transfer (REST) server that stores the machine learning model to identify users. So, until now, all the techniques for building the model and working on the voice have been presented. However, it is necessary to capture the user's voice; for that, the open-source VVA `Mycroft` is used, as shown in Figure 2.

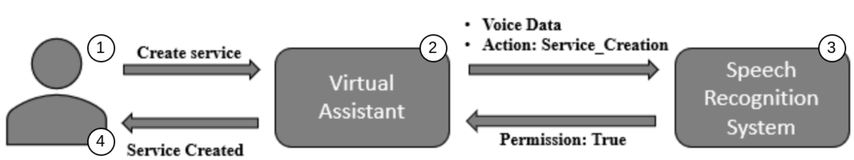


Fig. 2. Integration with Virtual Assistant

Here, the user (1) requests the creation of a service to `Mycroft` (2), which interprets the type of action required and sends it to the recognition server (3). The server recognizes the user by sending the signal captured to the machine learning model. After recognition, a query is made to the database to understand the user's roles and respective permissions. If the user has permission, the service is implemented (4). Otherwise, the user is informed that he has no permission. The system sequence diagram that illustrates the recognition and authorization process is shown in Figure 3.

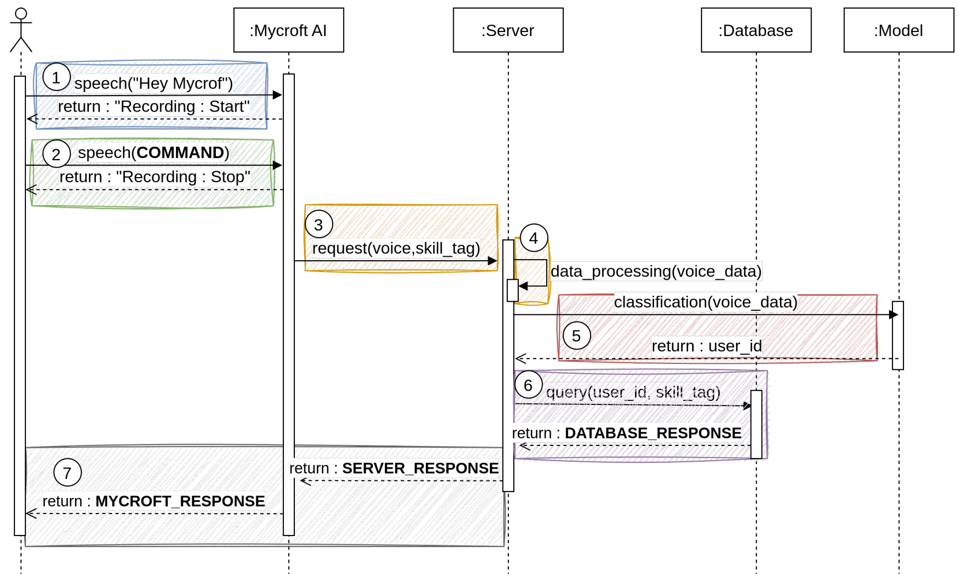


Fig. 3. Sequence System Diagram: Recognition and Authorization process

Detailing the sequence diagram, in (1), the user activates the VVAs using their default wake sentence *Hey Mycroft*. After being alerted, the recording of his command starts through a sound; the user communicates the desired command (2). The order is received on `Mycroft` and sent to the server via HTTP (3). The server receives the voice data and respective tag identifying which skill was triggered (user's action) (3). Next, voice data is processed (trim, feature extraction, and scaling) (4) and sent to the pre-trained model to identify (5). After that, a database query is performed to fetch information about the user to verify if the respective user has permission for the action required (6). Finally, based on previous decisions, the VVA `Mycroft` responds to the user about their permission (7).

3.2. Intent Based Network

Once the user is identified and allowed to implement the request, it goes to the IBN system, where a layer of intelligence is applied to the network, replacing manual network configuration processes and reducing the complexity of creating, managing, and enforcing network policies. To understand how this part of the system will work, the VINIA's IBN system architecture is shown in Figure 4.

The system architecture is designed to operate in a business context, where the user interacts with the VVA using their equipment (e.g., a computer). The VVA plays a pivotal role in maintaining communication between the system and the user to capture the user's commands. After being activated, the VVA saves the user's intention and forwards it to the

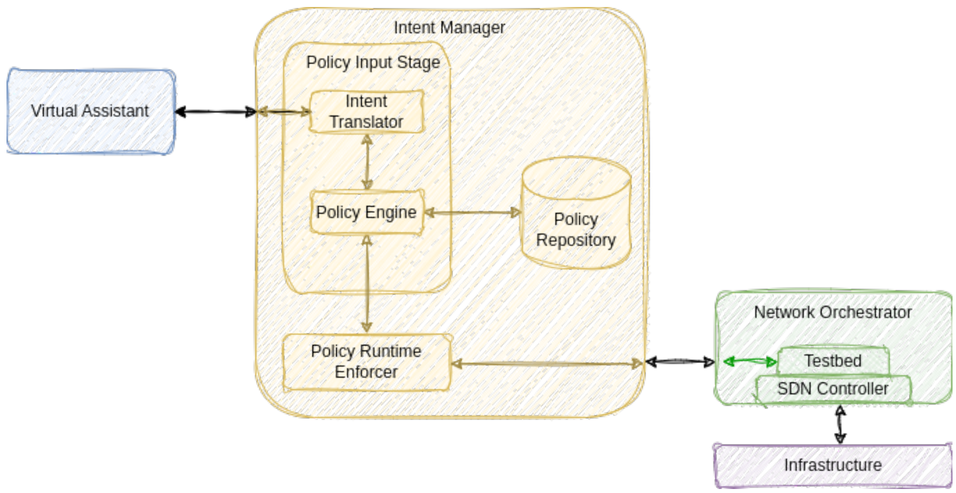


Fig. 4. VINIA's IBN System Architecture

Intent Manager to validate and implement it. The assistant is responsible for associating the policy with the user's intention and converting the high-level language into a machine language, sending it to the component where it will be analyzed.

The Intent Manager is a core system component, encompassing the Policy Input Stage, Policy Repository, and Policy Runtime Enforcer. The Intent Translator and Policy Engine components reside within the Policy Input Stage. The Intent Translator translates user intents from the VVA or other sources into a machine-readable format. The Policy Engine takes these translated intents and implements the logic and decision-making based on the policies. It ensures smooth communication between the VVA and the Policy Repository while also being accountable for analyzing policy conflicts to prevent clashes with existing network rules. In case of successful policy validation, this component forwards the rules to be implemented in the network infrastructure via Policy Runtime Enforcer. However, it initiates negotiation with the user in a conflict to resolve the issue.

The Policy Runtime Enforcer is a vigilant network monitor, continuously collecting and analyzing network information. It is critical in planning and executing policy implementations, considering operations and resources. The Policy Runtime Enforcer utilizes the data analysis results from the Policy Input Stage thresholds to take appropriate action if a policy fails.

The Testbed component is a REST server facilitating communication between the system and the SDN controller. It receives installation requests from the Policy Runtime Enforcer and attempts to install them in the SDN controller. Additionally, Testbed handles data collection requests that aid in applying monitoring algorithms and calculating variables for further study.

The SDN Controller obtains data from the network and implements any necessary rules. It communicates with the infrastructure, receiving essential information required for the system to make the desired changes. In this architecture, the OpenDayLight (ODL)[10] controller is utilized as the SDN controller.

The structure of intents represents all the information from the user structured so the system can interpret it. The system may receive several user requests, and creating a design that covers as much varied information as possible is crucial. The Intent structure template of our approach is shown in Listing 1.1.

Listing 1.1. Intent Structure

```
{
– IntentType:
– Intent Target:
– Intent State:
    Conditions:[
        Policy Type:
        Constraints:[
            Domains:[
                – Traffic Type:
                – Period:
                – Name:
                – Bool:
                – Access:
                – Performance:
            ]
        ]
    ]
}
```

Each request is assigned a type (for example: indicates whether the request is to set priorities or create services), a target (for example: what type of network), and a state (based on the state machine). Besides these three points, policies are defined according to the user's wishes. For each user-defined policy, we can have restrictions that indicate performance values, type of traffic, and who accesses the network, among others.

VINIA's Intent State Machine was drawn, Figure 5, considering the architecture and intent structure. It allows us to understand the life cycle of intent since it was captured by the VVA until its installation, not compromising the network operation. The intent state machine is divided into six sections. The validation phase and conflict analysis are part of the Policy input stage. The compilation and monitoring phases are processed in the Policy Runtime Enforcer module, and finally, the installation phase is the responsibility of the REST server `Testbed`.

In the initial phase, user requests are validated to ensure they include all required information for the desired action. If essential details are missing or inaccurate, an "invalid" status is assigned, necessitating user interaction. Conversely, a "valid" status is assigned if all necessary information is provided, allowing progression to the next phase.

The conflict phase is the second checkpoint in the state machine, comparing pre-validated attempts with stored requests in the database. Conflicts arise if requests are redundant or contradictory to existing information. When conflicts occur, a conflict state is assigned for user resolution. If there are no conflicts, the process advances to the compilation section.

During the compilation phase, the system translates user-requested policies from a high-level language into a format the controller interprets for installation. If the controller cannot support the desired information or operations, a compilation error occurs, which

[10] OpenDaylight Project — An Open Platform for Network Programmability. [online] Available at: https://www.opendaylight.org/

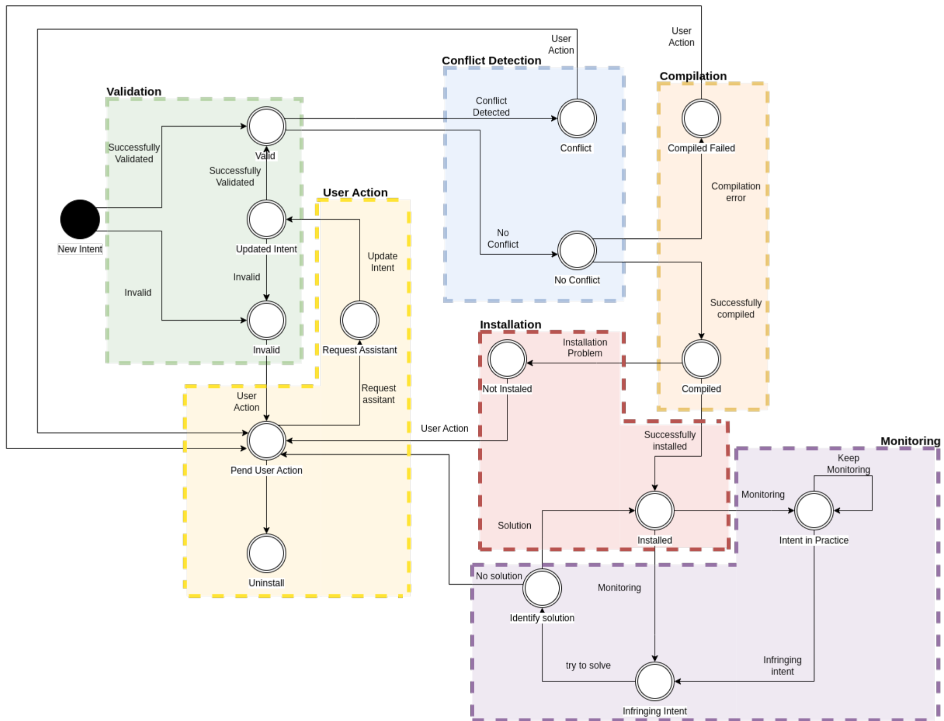


Fig. 5. VINIA's Intent State Machine

can be resolved with user assistance. Successful compilation leads to the subsequent installation phase.

Once the rules are obtained, the next step is to install them on the controller. If the controller supports the objectives and is installed, we move on to the monitoring phase. If the goals are not achievable because they may be offline or non-existent, the user is alerted that the application was not installed.

In the final phase, monitoring consists of a cycle that constantly checks if the existing requests in the database are being fulfilled or if any violation has occurred. If noncompliance occurs, the system automatically tries to identify how to solve the problem without human intervention. The user is alerted if the system does not specify a solution that corrects the problem. After the user is recognized and has permission, if the intent passes all these stages, the application is installed on the network and starts to be monitored.

3.3. Experimental Results

The results will be presented in two parts, the first referring to the speech recognition system results and the second to the intents system results.

Speech Recognition System To obtain suitable data for our study, it was necessary to collect datasets containing user speech. We utilized four distinct datasets, including NOIZEUS [54], TIMIT [55], LibrisSpeech ASR [56], and a custom dataset named SAFC.

In Table 2, these datasets are characterized by key attributes: dataset size, the number of speeches made by each user, and the average speech time.

Table 2. Dataset Properties

Name	Size (users)	Number of Speeches	Average Speech Time
NOIZEUS	6	5	2.6680 seconds
TIMIT	462	8	2.6726 seconds
LibrisSpeech	156	77 (average)	6.8771 seconds
SAFC	25	5	3.1416 seconds

As evident in Table 2, these datasets exhibit diverse characteristics, setting the stage for studying the scalability performance of the two selected algorithms concerning the growth in the number of users. This diversity also allows us to evaluate the performance of the selected approaches with respect to the amount of data per user.

In the context of industrial applications, where numerous users may be involved, it is imperative to ensure that a system can effectively handle a substantial user base. Scalability, measured as the performance of the different selected algorithms concerning the growth in the number of users, is of paramount importance. To meet this need, we have chosen "accuracy" as the primary performance metric for the study. Accuracy, in this context, serves as a critical metric due to its real-world relevance. In industrial scenarios, a high degree of user identification accuracy is vital to ensure the system functions correctly and securely. The primary objective of this study is to assess how well our selected algorithms perform in identifying users accurately, especially as the number of users grows. Accuracy reflects the system's ability to correctly classify and identify users in a multi-user environment, a crucial requirement in many practical applications. Hence, the choice of accuracy as a key metric is aligned with the practical context and the core objective of our study, which is to develop a robust and scalable user identification system suitable for real-world industrial applications. The primary study measures the performance of the different selected algorithms as a function of the growth in the number of users. Its performance is also analyzed with and without the use of data augmentation. The results are presented in Table 3, regarding the percentage of success rates.

We can conclude that both approaches perform well, and data augmentation is unnecessary in a system with minimal users and speeches per user. Regarding the experiments performed with the TIMIT dataset, we can conclude that both algorithms scale well with the increase of users, and data augmentation positively influences the performance of neural networks. Finally, to analyze the relation of the number of speeches per user, a final test of the LibrisSpeech dataset is carried out. This has 156 users; however, each has an average of 77 speeches. It is also necessary to be aware that each speech, on average, has approximately twice the work of the speeches of the other datasets. With these results, we observed that data augmentation has a negative performance on the LDA and a positive one on the neural network. This indicates that increased data per user influences the LDA performance negatively and the neural network positively.

Table 3. Dataset results

Dataset	Models	with Data Augmentation	without Data Augmentation
NOIZEUS	MLP	100%	100%
	LDA	100%	100%
TIMIT	MLP	100%	87,88%
	LDA	92,21%	95,67%
LibrisSpeech	MLP	100%	98,63%
	LDA	93,15%	93,84%
SAFC	MLP	100%	100%
	LDA	100%	100%

Intent System An illustrative scenario centered around service creation was formulated to showcase the outcomes related to the intention system. This scenario involved configuring a network topology comprising three switches and seven hosts, as depicted in Figure 6. Once created, there are no active services, as can be seen in Figure 7.

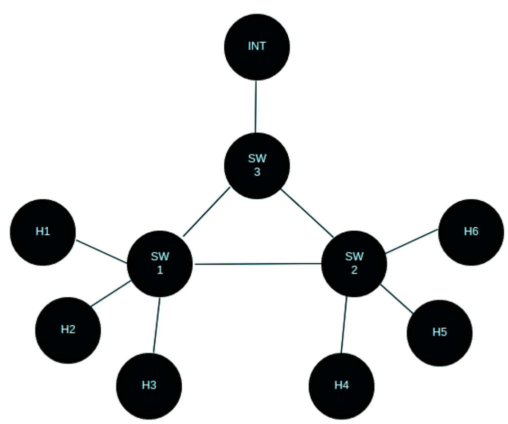


Fig. 6. Testing Environment

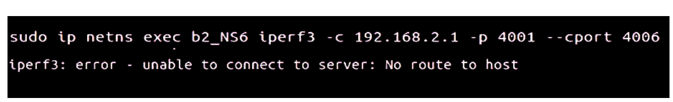


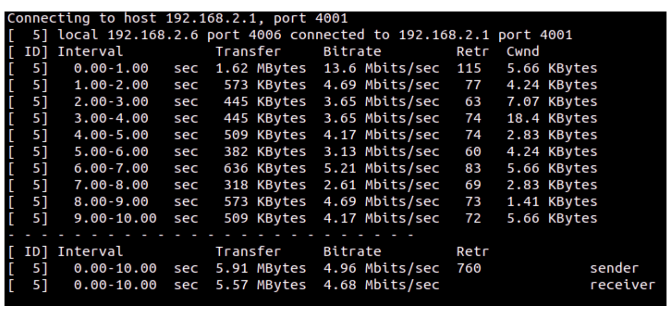
Fig. 7. Connection test between Host 1 and 6

The creation of a service depends on four parameters: the name of the service, who accesses the service, if the service needs an Internet connection, and finally, if there is a need to limit the bandwidth of the service. To understand how the user communicates these actions to the VVA, Mycroft, the subsequent dialogue is presented:

– User: ”Hey Mycroft, create a service, please.”
– Mycroft: ”Sure, what is the name of the service?”
– User: ”Call it s1.”
– Mycroft: ”Ok, please name the machines that will have access to the service.”
– User: ”Provides access to users 1 and 6.”
– Mycroft: ”Regarding internet access, does the service need internet access?”
– User: ”Yes, please.”
– Mycroft: ”One last question, do you want to define the performance of the service?”
– User: ”Yes, 4000 Mbits/sec.
– Mycroft: ”Thanks for the information, wait a bit while I implement the service.”

The user provides information to the VVA, which then initiates a state machine presented to validate, analyze conflicts, compile, and install the intent within the network upon obtaining permission. In the provided instance, the validation phase assesses whether complete data for service creation exists, verifying valid information in all mandatory fields required. Next, during the conflict analysis phase, the system checks if any information on this new service conflicts with any already implemented. It checks if the service name exists and if the hosts involved are active in other services. If any of these cases occur, the user is alerted. Once validated and without conflicts, the request goes to the build phase. In this phase, several requests are made to both the Testbed Server and the database to convert into rules, the policies from the user. If any requests fail, the compilation phase fails, and the user needs to be alerted. We move on to the installation phase when we get the rules compiled. In this phase, the rules are sent to the Testbed Server, which communicates with ODL to install them. After they are installed, they are stored in the database, and the user is notified that the application has been installed.

To test whether the service has been created between users 1 and 6, with Internet access and bandwidth limit set, we again use the `Iperf` tool in the same context mentioned above. The connection test can be visualized in Figure 8.



```
Connecting to host 192.168.2.1, port 4001
[  5] local 192.168.2.6 port 4006 connected to 192.168.2.1 port 4001
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.62 MBytes  13.6 Mbits/sec  115   5.66 KBytes
[  5]   1.00-2.00   sec   573 KBytes  4.69 Mbits/sec   77   4.24 KBytes
[  5]   2.00-3.00   sec   445 KBytes  3.65 Mbits/sec   63   7.07 KBytes
[  5]   3.00-4.00   sec   445 KBytes  3.65 Mbits/sec   74   18.4 KBytes
[  5]   4.00-5.00   sec   509 KBytes  4.17 Mbits/sec   74   2.83 KBytes
[  5]   5.00-6.00   sec   382 KBytes  3.13 Mbits/sec   60   4.24 KBytes
[  5]   6.00-7.00   sec   636 KBytes  5.21 Mbits/sec   83   5.66 KBytes
[  5]   7.00-8.00   sec   318 KBytes  2.61 Mbits/sec   69   2.83 KBytes
[  5]   8.00-9.00   sec   573 KBytes  4.69 Mbits/sec   73   1.41 KBytes
[  5]   9.00-10.00  sec   509 KBytes  4.17 Mbits/sec   72   5.66 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  5.91 MBytes  4.96 Mbits/sec  760             sender
[  5]   0.00-10.00  sec  5.57 MBytes  4.68 Mbits/sec                  receiver
```

Fig. 8. Connection test between Host 1 and 6(after service created)

Once the service has been created, the `Iperf` tool shows that it is possible to communicate between users 1 and 6 with Internet access and performance limited to 4000 Mbits/sec.

The presented solution showed good performance when applied to small-dimension networks. However, when dealing with more extensive networks, as in industrial contexts, the problem becomes more challenging to solve, increasing the waiting time between the initiation of the user request and its installation. In this way, the next chapter will introduce a new architecture so that the size of the network is not a problem for the installation of intents in the network.

3.4. Enabling E2E Network Orchestration in VINIA

Upon evaluation of the requirements for 6G networks [6] concerning the use of IBN, and considering the shortcomings of the architecture presented in Figure 4, we now propose a new architecture that enables End-to-End Network Orchestration in VINIA, in Figure 9. The architecture considers the latest developments in Network Automation & Orchestration and incorporates new insights and conclusions from the current state of the art. This architecture aims to provide a comprehensive and up-to-date framework offering significant improvements in efficiency and performance when dealing with Industrial networks. Moreover, it serves as a starting point for further research and development, and we believe it has the potential to make significant contributions to the field.

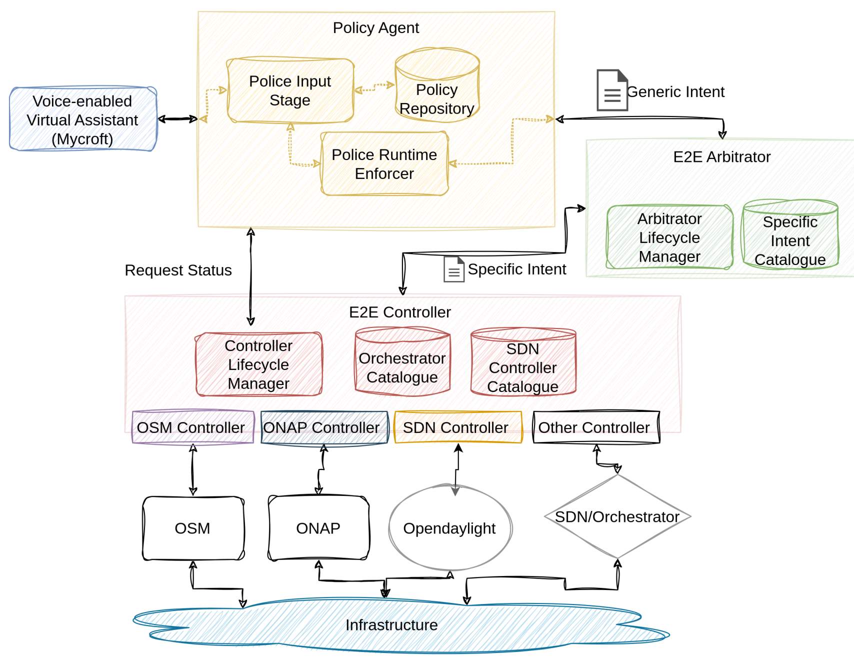


Fig. 9. New proposed architecture for VINIA

Regarding the user request capture and translation to an intent that involves the VVA and the Policy Agent, the architecture is unchanged from the one presented in Figure 4. The innovation comes when the compilation and installation process is initiated, where the Police Runtime Enforcer component is decomposed into several parts: E2E Arbitrator and E2E Controller. E2E Arbitrator is responsible for mediating between the high-level intents defined by network administrators and the underlying network infrastructure, ensuring that the intents are accurately translated. The E2E Arbitrator bridges the intent and network management layers, ensuring that the supporting Industrial network infrastructure is correctly configured and the desired network behavior is achieved. It receives a generic intent, as presented in Listing 1.1, then interprets the request and assumes what configurations are necessary to implement it using the Catalogue, which contains all the network information and what NOs are available and capable of implementing the intent. The output from this Arbitrator is a specific Intent with details on the components involved and which technologies to use to implement it. The E2E Controller is the intermediary between the E2E Arbitrator and the NOs, automating the translation of network intents into network configurations. It has several key components to help it manage the network infrastructure: Controller Lifecycle Manager, Orchestrators Catalogues, SDN Catalogue, and Orchestrators Controllers. The Controller Lifecycle Manager is responsible for managing the lifecycle of the E2E Controller, including initializing, updating, and terminating the controller as needed. The Orchestrator Catalogue is a repository of information about available NOs, including their capabilities, functionalities, and dependencies. As Orchestrator Catalogue, SDN Controller Catalogue is a similar repository for SDN controllers, providing information about available controllers and their capabilities. The last component, Orchestrators Controller, starts the installation process, sending the network configurations to the respective orchestrator. When the E2E Controller receives the Specific Intent, it uses the information stored in the Orchestrator Catalogue or SDN Catalogue(depending on the associated technology in the Arbitrator) to compile the intent in network configurations. Once the intent is compiled in network configurations readable by the respective orchestrator, the E2E Controller sends those configurations to the orchestrator to implement the intent by configuring the underlying network resources. This enables the Industrial network infrastructure to be managed and automated flexibly and efficiently, ensuring that the network behaves in the desired way.

4. Conclusion and Future Directions

Using high-level network intents to define desired network behavior, IBN enables network administrators to manage heterogeneous network infrastructures, including industrial ones, more efficiently and agilely. However, implementing IBN within traditional SDN contexts is challenging. Some issues are limited scalability, integration with other network components, and difficulty translating high-level intents. In VINIA's new architecture, we have explored various IBN implementation strategies, with a special focus on leveraging NOs like ONAP or OSM to enhance effectiveness and scalability.

The proposed VINIA's architecture towards supporting different components(including proprietary ones) allows for defining an E2E network. VINIA ensures support for Network Slices and compatibility with heterogeneous network infrastructure configurations in Industrial Networks(e.g., 5G and B5G Networks). Following VINIA's core definitions, automating the networks allows for the redefinition of the network operating cost. Moreover, as requirements evolve or expand, VINIA can integrate additional components, minimizing disruptions and ensuring sustained performance even as the network grows.

Moving forward, we will focus on validating the introduced architecture in the 6G Brains project, particularly through slice use cases [10] demonstrating its effectiveness in real-world scenarios. We aim to test the new components for each orchestrator type to ensure seamless integration with existing technologies and infrastructure. Through the use of orchestration in cooperation with monitoring information on the ongoing status of the network at the control plane, it is possible to have a disaggregated view of the network. The ability to control the different network segments of a E2E network allows for defining more complex intents to manage the different network's resources. The use of monitoring mechanisms as well as topology discovery mechanisms can be used to actively preclude QoS degradation. Towards this goal, in [10], a mechanism for enriching ONAP with information about the network was presented. Traditional SDN-based implementations are unaware of the E2E network. Therefore, using orchestrators such as OSM/ONAP improves the response to network changes, ensuring the E2E network can reconfigure quickly. Therefore, using orchestrators helps improve the dynamicity of the solution. Having a complete picture of the E2E network will allow for better monitoring, promoting the capability to detect points of failure/congestion. In the future, we aim to leverage the use of closed-control loops to detect problems in components managed by the VINIA system. In the scope of 6G BRAINS, we have been developing the first iteration of this concept, focused on the RAN. This concept relies on the metrics collected in the RAN for changing the type of network service provided in a specific RAN network slice.

Beyond network orchestration, we plan to expand the functionalities of the VVA as well as testing another VVA, Neon AI [11] as an alternative to the use of Mycroft. Furthermore, we plan to enhance the language understanding capabilities of VVAs, making them more inclusive by supporting multiple languages, accents, dialects, and colloquialisms, testing alternatives to Mozilla's DeepSpeech [12] Advances in natural language processing and machine learning techniques hold the potential to significantly improve VVA performance and accessibility for a broader range of users.

By addressing these aspects, we expect to overcome the constraints of conventional SDN-based IBN implementations, paving the way for a more efficient and integrated network management system. The continual evolution of IBN and VVA technologies will contribute to the progressive development and practical deployment of advanced network automation solutions in the industrial context.

Acknowledgments. This work was funded by the European Commission Horizon 2020 5G-PPP Program under Grant Agreement N. H2020-ICT-2020-2/101017226 "6G BRAINS: Bringing Reinforcement learning Into Radio Light Network for Massive Connections".

References

1. Kashif Mehmood, H. V. Kalpanie Mendis, Katina Kralevska, and Poul E. Heegaard. Intent-based network management and orchestration for smart distribution grids, 2021.
2. Talha Ahmed Khan and et al. Intent-based orchestration of network slices and resource assurance using machine learning. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–2, 2020.
3. Hong Chen, Ling Li, and Yong Chen. Explore success factors that impact artificial intelligence adoption on telecom industry in china. *Journal of Management Analytics*, 8(1):36–68, 2021.

[11] https://neon.ai/

[12] https://github.com/mozilla/DeepSpeech, which are used by both Mycroft AI and Neon AI.

4. Mounir Bensalem, Jasenka Dizdarević, Francisco Carpio, and Admela Jukan. The role of intent-based networking in ict supply chains. In *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*, pages 1–6, 2021.
5. Raul Barbosa and Marco Araujo. Ai-driven human-centric control interfaces for industry 4.0 with role-based access. In *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6, 2022.
6. Ömer Bulakçı (ed.) and et al. *Towards Sustainable and Trustworthy 6G: Challenges, Enablers, and Architectural Design*. Boston-Delft: now publishers, 2023.
7. Abdallah Moubayed, Abdallah Shami, and Anwer Al-Dulaimi. On end-to-end intelligent automation of 6g networks. *Future Internet*, 14(6), 2022.
8. Hatim Chergui, Adlen Ksentini, Luis Blanco, and Christos Verikoukis. Toward zero-touch management and orchestration of massive deployment of network slices in 6g. *IEEE Wireless Communications*, 29(1):86–93, 2022.
9. Karima Velasquez, David Perez Abreu, Marilia Curado, and Edmundo Monteiro. Resource orchestration in 5g and beyond: Challenges and opportunities. *Computer Communications*, 192:311–315, 2022.
10. João Fonseca and et al. 6g brains topology-aware industry-grade network slice management and orchestration. In *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pages 418–423, 2023.
11. Masoud Narouei and Hassan Takabi. Automatic top-down role engineering framework using natural language processing techniques. In Raja Naeem Akram and Sushil Jajodia, editors, *Information Security Theory and Practice*. Springer International Publishing, 2015.
12. Aodi Liu, Xuehui Du, and Na Wang. Efficient access control permission decision engine based on machine learning. *Secur. Commun. Networks*, 2021:3970485:1–3970485:11, 2021.
13. Fan Deng and Li-Yong Zhang. Elimination of policy conflict to improve the pdp evaluation performance. *Journal of Network and Computer Applications*, 80:45–57, 2017.
14. Said Marouf, Mohamed Shehab, Anna Squicciarini, and Smitha Sundareswaran. Adaptive reordering and clustering-based framework for efficient xacml policy evaluation. *IEEE Transactions on Services Computing*, 4(4):300–313, 2011.
15. Santiago Pina Ros, Mario Lischka, and Félix Gómez Mármol. Graph-based xacml evaluation. In *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*, SACMAT '12, page 83–92, New York, NY, USA, 2012. Association for Computing Machinery.
16. Manar Alohaly, Daniel Takabi, and Eduardo Blanco. Automated extraction of attributes from natural language attribute-based access control (abac) policies. *Cybersecurity*, 2, 12 2019.
17. Eni. Gr eni 003 - v1.1.1 - experiential networked intelligence (eni); context-aware policy management gap analysis. Technical report, ETSI, 2018.
18. Yiming Wei, Mugen Peng, and Yaqiong Liu. Intent-based networks for 6g: Insights and challenges. *Digital Communications and Networks*, 6(3):270–280, 2020.
19. Engin Zeydan and Yekta Turk. Recent advances in intent-based networking: A survey. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020.
20. Min Xie, Pedro Henrique Gomes, Jörg Niemöller, and Jens Patrick Waldemar. Intent-driven management for multi-vertical end-to-end network slicing services. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 1285–1291, Dec 2022.
21. Vijay Varadharajan, Kallol Karmakar, Uday Tupakula, and Michael Hitchens. A policy-based security architecture for software-defined networks. *IEEE Transactions on Information Forensics and Security*, 14(4):897–912, 2019.
22. Cristian Cleder Machado and et al. Policy authoring for software-defined networking management. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, pages 216–224, 2015.
23. Douglas Comer and Adib Rastegarnia. OSDF: A framework for software defined network programming. *CCNC 2018 - 2018 15th IEEE Annual Consumer Communications and Networking Conference*, 2018-Janua(October 2017):1–4, 2018.

24. Francesca Moscatelli and et al. Demo: Service-driven management of vertical services in heterogeneous network slices. In *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 1–3, 2021.
25. Khizar Abbas and et al. Ensemble learning-based network data analytics for network slice orchestration and management: An intent-based networking mechanism. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5, 2022.
26. Talha Ahmed Khan, Khizar Abbas, Afaq Muhammad, and Wang-Cheol Song. An intent-driven closed-loop platform for 5g network service orchestration. *Computers, Materials & Continua*, 70(3):4323–4340, 2022.
27. Flávio Meneses, Manuel Fernandes, Daniel Corujo, and Rui L. Aguiar. Slimano: An expandable framework for the management and orchestration of end-to-end network slices. In *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*, pages 1–6, 2019.
28. Sumbal Malik, Manzoor Ahmed Khan, Aadam, Hesham El-Sayed, Jalal Khan, and Obaid Ullah. Implanting intelligence in 5g mobile networks;a practical approach. *Electronics*, 11(23), 2022.
29. Serge Fdida, Nikos Makris, Thanasis Korakis, Raffaele Bruno, Andrea Passarella, Panayiotis Andreou, Bartosz Belter, Cédric Crettaz, Walid Dabbous, Yuri Demchenko, and Raymond Knopp. Slices, a scientific instrument for the networking community. *Computer Communications*, 193:189–203, 2022.
30. Peter Bailis and Kyle Kingsbury. The network is reliable: An informal survey of real-world communications failures. *Queue*, 12(7):20–32, jul 2014.
31. Nick Feamster and Hari Balakrishnan. Detecting BGP configuration faults with static analysis. In *2nd Symposium on Networked Systems Design & Implementation (NSDI 05)*, Boston, MA, May 2005. USENIX Association.
32. Maximilian Wilhelm. Bgp security and confirmation biases. online 08 Nov 2023.
33. 3GPP TSG. Telecommunication management; study on scenarios for intent driven management services for mobile networks. Technical Report 28.812, 3GPP, set 2022.
34. 3GPP TSG. Management and orchestration; provisioning. Technical Specification 28.531, 3GPP, jun 2021.
35. TM Forum. Autonomous networks scenario realizations. Introductory Guide 1230A, TM Forum, may 2021.
36. TM Forum. Autonomous networks industry standards. Introductory Guide 1230B, TM Forum, may 2021.
37. TM Forum. Autonomous networks reference architecture. Introductory Guide 1251, TM Forum, may 2021.
38. TM Forum. Study of telecom industry intent meta-modeling approaches. Introductory Guide 1259, TM Forum, may 2021.
39. TM Forum. Autonomous networks business requirements and framework. Introductory Guide 1218, TM Forum, dec 2022.
40. Adeel Rafiq, Asif Mehmood, and Wang-Cheol Song. Intent-based slicing between containers in sdn overlay network. *Journal of Communications*, pages 237–244, 01 2020.
41. Federica Paganelli, Francesca Paradiso, Monica Gherardelli, and Giulia Galletti. Network service description model for vnf orchestration leveraging intent-based sdn interfaces. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–5, 2017.
42. Walter Cerroni and et al. Intent-based management and orchestration of heterogeneous openflow/iot sdn domains. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–9, 2017.
43. Yoshiharu Tsuzaki and Yasuo Okabe. Reactive configuration updating for intent-based networking. In *2017 International Conference on Information Networking (ICOIN)*, pages 97–102, 2017.

44. Arthur Selle Jacobs, Ricardo José Pfitscher, Ronaldo Alves Ferreira, and Lisandro Zambenedetti Granville. Refining network intents for self-driving networks. In *Proceedings of the Afternoon Workshop on Self-Driving Networks*, SelfDN 2018, page 15–21, New York, NY, USA, 2018. Association for Computing Machinery.
45. Fred Aklamanu and et al. Intent-based real-time 5g cloud service provisioning. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2018.
46. Rami Akrem Addad, Diego Leonel Cadette Dutra, Miloud Bagaa, Tarik Taleb, Hannu Flinck, and Mehdi Namane. Benchmarking the onos intent interfaces to ease 5g service management. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018.
47. Adeel Rafiq, Asif Mehmood, Talha Ahmed Khan, Khizar Abbas, Muhammad Afaq, and Wang-Cheol Song. Intent-based end-to-end network service orchestration system for multi-platforms. *Sustainability*, 12(7), 2020.
48. Khizar Abbas, Muhammad Afaq, Talha Ahmed Khan, Adeel Rafiq, and Wang-Cheol Song. Slicing the core network and radio access network domains through intent-based networking for 5g networks. *Electronics*, 9(10), 2020.
49. Orken Mamyrbayev, Nurbapa Mekebayev, Mussa Turdalyuly, Nurzhamal Oshanova, Tolga Ihsan Medeni, and Aigerim Yessentay. Voice identification using classification algorithms. In Yang (Cindy) Yi, editor, *Intelligent System and Computing*, Rijeka, 2019. IntechOpen.
50. Meenu Yadav, Dr. Vinod Kumar Verma, Chandra Yadav, and Jitendra Verma. Mlpgi: Multilayer perceptron-based gender identification over voice samples in supervised machine learning. In *Applications of Machine Learning*, pages 353–364, 05 2020.
51. Mucahit Buyukyilmaz and Ali Osman Cibikdiken. Voice gender recognition using deep learning. In *Proceedings of 2016 International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA2016)*, pages 409–411. Atlantis Press, 2016/12.
52. Orken Mamyrbayev, Nurbapa Mekebayev, Mussa Turdalyuly, Nurzhamal Oshanova, Tolga Ihsan Medeni, and Aigerim Yessentay. Voice identification using classification algorithms. In Yang (Cindy) Yi, editor, *Intelligent System and Computing*, chapter 3. IntechOpen, Rijeka, 2019.
53. M Murugappan, Nurul Qasturi Idayu Baharuddin, and S Jerritta. Dwt and mfcc based human emotional speech classification using lda. In *2012 International Conference on Biomedical Engineering (ICoBE)*, pages 203–206, 2012.
54. Philip Loizou. NOIZEUS: A noisy speech corpus for evaluation of speech enhancement algorithm.
55. John S. Garofolo and et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus.
56. Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2015-Augus:5206–5210, 2015.

Raul Barbosa is currently pursuing his PhD at the University of Aveiro and is affiliated with the Telecommunications Institute. His research centers around the development of AI-based systems to enhance networks in autonomous driving, industrial settings, and other vertical domains. Alongside his academic pursuits, Raul holds the role of Professional II-Connectivity & NW Engineer in Capgemini's R&D Telecom team, actively contributing to various national and international research projects.

João Pedro Fonseca is a PhD Student enrolled in the MAPi Programme, and associated with the Telecommunications Institute. He currently pursues research on Vertical-aware and Dynamic System Orchestration. He is a Connectivity & NW Engineer at the R&D Telecom team of Capgemini, contributing to national and international research projects.

Marco Araújo is a telecommunications expert with over 15 years of experience working with telecom operators like NOS, T-Mobile, Vodafone, Oi, Vivo, and Telefonica. He holds a PhD from Stockholm University, specializing in applications of telecommunications systems engineering for the automotive and aerospace sectors. Leading R&D projects such as Horizon Europe and collaborating with the European Space Agency, Marco manages budgets as a senior manager in Portugal's R&D department, ensuring innovative progress.

Daniel Corujo is an Associate Professor at the University of Aveiro and a Researcher at the Telecommunications Institute, both in Portugal. He pursues research on programatic mobile networking in national and international research projects, holding several research papers, standards and patents. He is Vice-chair of the IEEE COMSOC PT Chapter.

Received: February 13, 2023; Accepted: November 15, 2023.