The Role of Intent-Based Networking in ICT Supply Chains

Mounir Bensalem, Jasenka Dizdarević, Francisco Carpio, and Admela Jukan

Technische Universität Braunschweig, Germany

{mounir.bensalem, j.dizdarevic, f.carpio, a.jukan}@tu-bs.de

Abstract—**The evolution towards Industry 4.0 is driving the need for innovative solutions in the area of network management, considering the complex, dynamic and heterogeneous nature of ICT supply chains. To this end, Intent-Based networking (IBN) which is already proven to evolve how network management is driven today, can be implemented as a solution to facilitate the management of large ICT supply chains. In this paper, we first present a comparison of the main architectural components of typical IBN systems and, then, we study the key engineering requirements when integrating IBN with ICT supply chain network systems while considering AI methods. We also propose a general architecture design that enables intent translation of ICT supply chain specifications into lower level policies, to finally show an example of how the access control is performed in a modelled ICT supply chain system.**

I. INTRODUCTION

In the recent years intent-based networking (IBN) has been the emerging networking paradigm, levering the integration of AI/ML techniques with network orchestration to improve upon the network automation, complexity and control functionalities. Compared to traditional network management solutions it has shown great potential in numerous aspects, reducing the time for network configuration deployment while improving on its scalability [1], increasing both the efficiency and reliability, as well as reducing costs in particularly complex industrial networks [2] and integrating security through its intent functions [3]. The key behind IBN's increasing abilities to solving complex network tasks lies is in their high-level policies specification capabilities that allow network administrator to define goals and demands on how the network should behave.

One of the particularly significant IBN potential application domains are ICT supply chains, since the fast-growing technological evolution and the digital transformation behind Industry 4.0 and ICT supply chains have created a gap between traditional networking solutions and these emerging complex networks. As now ICT supply chains can span across multiple vendor infrastructures and devices, assuming different configuration requirements and capabilities with continuous addition of new functionalities, the interest in novel network management and orchestration approaches has been growing rapidly. The utilization of IBN in this domain could facilitate automation, control and orchestration processes, while improving on the network robustness and operation dynamicity.

This paper introduces an IBN approach to automating the configuration management and control of ICT supply chain networks through high-level intents. The objective is to first analyse the key engineering requirements in integrating IBN approach with these complex networks, considering the questions of centralized vs decentralized orchestration, influence of different AI techniques, and manner of specifying high level intents to ICT supply chain policies. To this end we propose an architectural design of intent based translator for ICT supply chains, as well as an example in the context of supply chain assets access control.

The rest of the paper is organized as follows. Section 2 describes the background and motivation behind using intent-based approaches. Section 3 gives an overview of engineering requirements, the general architectural design for utilization of IBN in ICT supply chains, and a use case scenario for assets access control. Finally, Section 4 concludes the paper.

II. BACKGROUND AND MOTIVATION

In this section we analyse the common architectural building blocks of any intent-based system, according to the already existing solutions. While the design of intent-based translators can follow guidelines depending on the scenario to be applied, this analysis was used as the initial step in proposing an IBN architectural design of the ICT supply chain system. Table I shows a comparative analysis of the intent abstraction layer design for ICT systems. Several proposals from the literature are investigated to understand the main software components responsible on interfacing with the user, storage, management, policies, networking, monitoring, and enabling intelligence. It is important to mention that few proposals highlighted the role of AI and ML in translating user intents [4] and improving policies using insight from telemetry [5]–[7].

A. Intent-Based Translator Design

1) High-Level Language: The definition of a high level intent-based language is essential to allow the user to express its requirements to the system. Multiple works in the literature try to define this type of language, for instance, Riftadi et al [8] employed Nile, a network intent language that provides an intermediate layer between natural language and lower-level policies. In [15] a high level declarative policy language is proposed to control the architecture of software-defined networking (SDN) based on the notion of functional reactive programming where policies are used to state the change in

Work	Interface	Management	Storage	Policies	Network	AI/ML	Monitoring	Application
[8]	GUI	intent parser & compiler	service library	policy builder	switch controller	no	no	switches programmability
[9]	extended NBI	intent manager & compiler	vocabulary store	conflict checking	SDN controller	no	yes	—
[10]	GUI, Alexa	intent engine & compiler	MariaDB		floodlight controller	—	no	network visualization
[11]	UI	intent manager & compiler	service registry	service integration element	SDN controller	no	no	SDN applications
[12]	GUI, Alexa	intent engine	database		Ryu controller	no	—	self-Healing SDN
[7]	NBI	intent manager	policy store	policy configurator	resource controller	LSTM, GAN	yes	orchestration of network slicing
[5]	NBI	policy parser	policy store	conflicts management	SDN controller	heuristics	yes	SDN-packet processing
[13]	REST API	intent engine	no	PCE	Ryu Controller	no	no	mobile backhauling
[6]	GUI	intent manager	policy store	policy configurators	FlexRAN, open MANO	DL, GAN	yes	network slicing
[4]	slack API	intent manager & compiler	ontology and RDF	conflict checking	—	LSTM	—	high-speed networks
[14]	NETCONF	extractor & compiler	—	convertor	—	no	no	IoT device configuration
our work	GUI	intent manager & compiler	policy store, intent store, telemetry	policy configurator	resource, controller	RL, LSTM	yes	ICT supply chain

TABLE I: Comparative analysis of intent-based architectural components

network conditions. In [5], it is defined an Open Software Defined Framework (OSDF) policy language used to provide a high level API in order to express network requirements by managers and network administrators. [16] uses a Controlled Natural Languages (CNL), which sets restrictions on inputs by the definition of grammar model. In this case, CLN was employed to control blockchain selection queries, as mapping requirements to a particular blockchain needs expertise and knowledge about the existing blockchain implementations. Recent development of natural language processing (NLP) algorithms [10], [12] show how voice-assisted technologies have been used to manage the network as well.

2) Interfaces: The initial works on intent-based interfaces support imply their abilities of managing internal and external interactions independently from the underlying network technologies. In the approach by Szyrkowiec et al [17], the intents issued by a client application, are submitted through a representational state transfer (REST) northbound interface (NBI). Similarly, in the work by Zeydan et al [18], intent based NBI of NIC allows descriptive way to obtain what is desired from the network using any kind of protocols, including OpenFlow, BGP, and Netconf. In the work by Pham et al [11], the controllers NBI and core services are used as the atomic layer in the service orchestration because the application wants to retrieve abstract network topology provided by the controller. The intent layer in Han et al [9] provides extended North Bound Interface (NBI) that is used

by various applications.

Mahtout et al [4], deployed EVIAN, an approach for using ML for intent-based provisioning, the EVIAN client provides an intelligent chatbot interface, via Slack, to users in order to understand their network requirements and science needs.

3) Manager/Parser: In the [2], authors have recognized a core component of IBNs, which interacts with most of the other components, naming it an Intent Processor. This core component is responsible for accepting high level intents expressed by users (applications) using a semantic language or through other appropriate user interfaces. For instance, it can receive input in plain English and, in turn, it can translate that into say, Algebraic Modelling Language, which can be easily understood by a mathematical solver. In [5], the similar functionality has been been identified under the name Policy Parser, responsible for analyzing application-based policies.

4) Policy Builder: In the work by Abbas et al [6] on creating core network slices through IBN, the policy configurator extracts the information related to resources from the graph, provided by the intent manager and converts the specifications into a slice template. That slice template is according to the network orchestrator acceptable format; for example, for the OSM core, the policy configurator can give the slice template in form of a JSON file which contains the information related to the deployment of network functions and their mapping.

5) Compiler: The intent compiler is one of the key functionalities related to IBN system design, with the task of

converting the translated specifications or policies into technical configurations expected to realise the intent. As such it has been a focus of numerous research works. In [14] it is used to generate a low-level configuration XML format from the low-level configuration data using context-free grammar CFG. In [19] extracted data is converted to network security function required data via Data Converter. In [8] an element called P4 code generator is proposed to generate a working P4 program with the input of a policy graph introduced by the user as an intent, while in [20] generic constraints are converted into technology-specific ones, to be later applied by network controller.

6) Knowledge Base (KB): The KB is a database that stores all the necessary information for the intent based system, such as predefined policies, intent vocabularies and telemetry used for improving policy building and intent translation process. In the recent IBN works, KB has been mostly recognised under the name policy store [5]–[7]. Its main tasks include storing and retrieving application-based network policies that a user enters to the system. As such, it has been used to access the current active policies in the system, allowing network administrators to create, read, update, delete policies at runtime. In systems like [8], the policy store represents an action library containing P4 code templates, that can be parsed by the intent engine into a policy graph.

B. Application in ICT systems

With an increased interest in the automation of network orchestration, the emerging IBN approaches are being extensively analysed for its utilization in numerous application fields, from vehicular, IoT and 5G scenarios, to its potential in provisioning intent-based security policies and services.

One of the most interesting application assumed to benefit of IBN has been the Vehicle Network Management with its distributed and self-organizing networks composed of diverse and enormous number of high-speed vehicles; Here, IBN-based policies are expected to manage vehicular networks towards an intelligent network behavior according to the underlying applications or IoT workflows, as well as help in minimizing the energy consumption and reducing of network latency. An example can be seen in [21], where the authors designed an IBN control framework over the SDN architecture in the vehicular edge computing ecosystem, achieving the set goals of improving on network energy efficient and latency reduction. network latency. High-level programming abstractions and declarative languages [13] are posed to significantly benefit mobile networks and applications as well, due to the increase in their complexity over the years. This can be seen in the design and prototype implementation of an Intent-based mobile backhauling interface for 5G networks presented in [13] and in [22], where a wireless home automation system with automatic spoken word into text commands translation was proposed.

The possibilities of smart network management and control with IBNs have also been appealing with industrial solutions, with Huawei [23] developing an IBN product called Intent-

Driven Network (IDN) for its cloud solution. Likewise Open-DayLight Controller Platform has been involved in numerous strategies on enabling the utilization of intents with Network Intent Composition (NIC) [24], Group-based Policy (GBP) intent abstraction [25], and NEMO [26] language.

The full potential of IBN approach is probably most evident in network security, where the security can be incorporated as an integral part of intent-based networking across intent, automation, and assurance functions. With their continuous loop frameworks, IBNs have proven to provide ongoing protection and alignment to security policy and compliance requirements [3]. Moreover, IBN implementation can provide traffic segmentation based on business roles for connections to remote or cloud computing sites to simplify threat protection and reduce the attack surface [3]. A policy-driven security orchestration framework were presented in has also been presented in [27] in SDN/NFV-aware IoT scenarios.

III. System Design for ICT Supply Chains

A. Requirements Engineering

Deployment of IBN network orchestration and policy-based automation solutions in ICT supply chain management carries a great potential. But, in order to leverage this potential the future system design needs to adequately address some of the main requirements which affect both of these domains, and can be assumed to be critical in their integration.

Centralized/decentralized orchestrating approach: When opting for a straightforward automation and deployment of intents, with the total control being handed over to a single node with the complete knowledge of the network, the choice would be a centralized network orchestrator. This approach comes at a price of that the central node machine running the orchestrator would have to have significant storage and processing power in order to maintain information for potentially very large network. With a decentralized approach, distributed orchestrators maintain an incomplete information of the network, allowing higher levels of scalability but introducing its own challenge regarding the efficient processing of intents across multiple orchestrating nodes [1]. A similar discussion is led in assessing the advantages and disadvantages of centralized and decentralized control and orchestration strategies in todays complex ICT supply chains environments has to be considered during the design phase of the management system. The more traditional centralized approaches which often assume complete information sharing among different SC organizational system components with the a top hierarchical decision making authority have been noted to offer more stable environments, faster decision making and easier coordination. On the other side decentralized approach, with orchestration and decision making left to individual SC organizational system components, benefits the SC efficiency.

Intent specification: One of the crucial issues in IBN has been on how to actually specify high-level intents, and here there is still a lack of standardized solutions. And while there are ongoing efforts towards improving the ways intents can
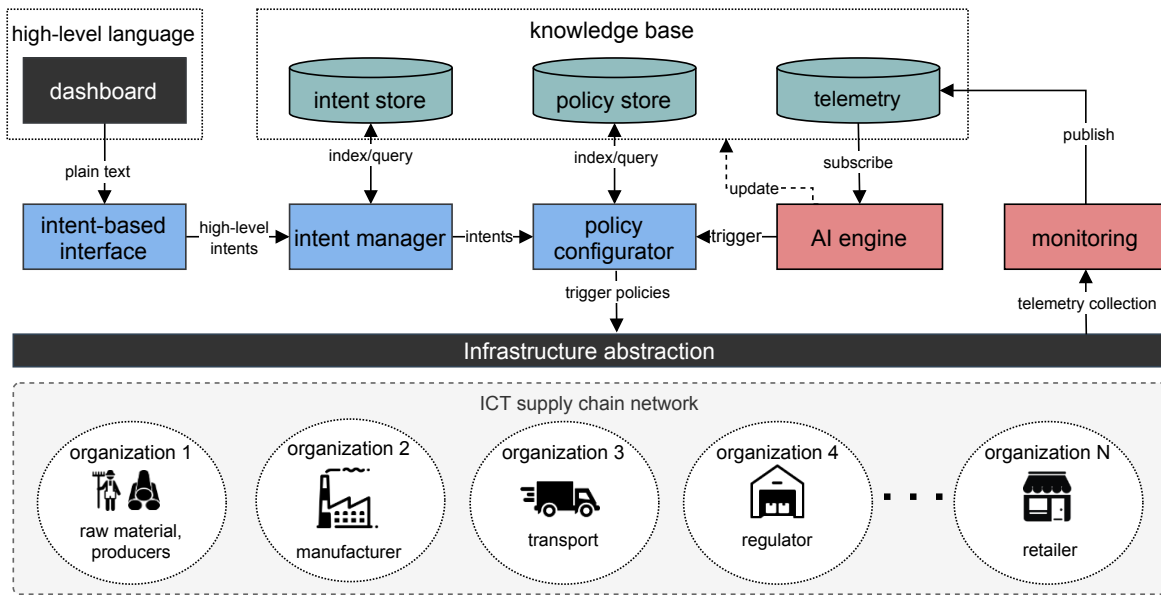
Fig. 1: Intent-based translator design for ICT supply chain networks

be described and approaching the naturally spoken languages (e.g NEMO and INDIRA), how this will relate to translating intent-based policies is to be discussed, but a general language model will be required in order to provide a scalable solution for different ICT supply chain systems.

AI technologies: Both IBN and ICT supply chains have been greatly influenced by the advances in the domains of AI and ML. In case of AI-powered IBNs, the performances have been improved by using AI in network analysis and intent implementation as well as using AI-based heuristics for solving constrained optimization problems. Here the challenge lies in a critical issue for any AI/ML algorithm - the lack of data. To tackle this we can assume that IBN will have to operate for certain amount of time to have enough data available, so it will be necessary to assess how to use data collected from the pre-IBN deployment. Moreover, when leveraging the potential of AI techniques for IBN applications in supply chain management (SCM) there are two additional questions that need to be addressed. Firstly, the question of leveraging the potential of natural language processing (NLP) [28] in improving humanmachine interactions in SCM should be addressed. Secondly, as supply chains are particularly important from industrial networks perspective, in case of an incident it is a requirement to obtain the explanation behind it, which can be difficult in case it is a consequence of an IBN automated algorithm reaction [2].

Closed-loop verification: With closed-loop verification method, IBNs maintain the enforced network state and configuration through intents, which can be expressed as series of smaller network element configurations [29]. The problem lies in the complexity of this process, which assumes constant system and network configuration adjustments in order to satisfy desired system requirements and can result in autonomous remediation errors or unwanted delays. The complex system

architectures as the ones behind SC scenarios will need to improve on this method, as a seemingly small configuration adjustments can affect the entire network across.

Single vs Multi-Domain: Majority of the IBN deployment and research work focuses on single-domain network approach, with the orchestrator centralization. However, this kind of approach is insufficient in addressing the SCM scenarios, as they normally extend across multiple companies, decision-makers and network domains. To address this particular issue there is a need for shifting towards a more multi-domain oriented IBN approach, where an intent framework will be implemented in a manner that would ensure end-to-end network orchestration across multiple potential SC domains.

B. Architecture Design

The intent based translator aims to automate the configuration management of ICT supply chain networks. Thus, we propose a general software architecture for the intent abstraction layer, as shown in Fig.1. Supply chains include various organizational partners that own different parts of the supply chain and need to cooperate in order to successfully deliver a final product or service. The management solution of such network will allow different network administrators to easily configure and control the network via high-level intents. A plain text, that is restricted by a predefined high-level language, is input by the user, using a dashboard that communicates the information via an intent-based interface. This interface is developed as to facilitate the human machine interactions, considering the ability to control the infrastructure using high-level intents for security assessment, network service management and control. From here, as shown in Fig.1, high-level intents are translated and parsed by the intent manager into a structured format, using natural language processing techniques. A knowledge base is
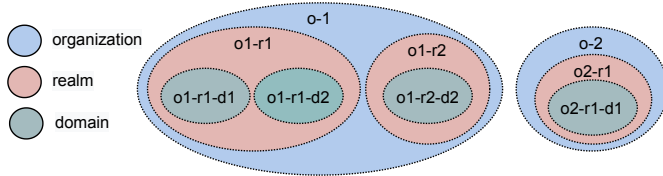
Fig. 2: Supply chain abstraction building block

then used to store the intent vocabularies in a json format, where it is used by the intent manager to determine the intent specifications. The vocabularies inside the knowledge base can be enriched dynamically, improving the system scalability and capability to manage supply chains of different contexts. After producing a structured format of an intent with user requirement specifications, the intent is sent to a so-called policy configurator component, responsible on matching the user requirements with existing policies that can be applied in the infrastructure. This component will then check the existing policies stored in the knowledge base in a defined policy store that can be updated and accessed by developers and experts in order to define new policies. The created policies are mapped by the policy configurator into a lower level format, after checking for possible conflicts among them. The policy configurator is also responsible for triggering the network controller, using defined APIs. In proposed system we also consider the possibility of smartly reconfiguring the network based on insights from the system. Metrics are collected by monitoring tools from all the network elements, and used to periodically update the knowledge base, which includes telemetry as well as storing policies and intent vocabularies. A component called AI Engine is responsible on analyzing the data coming from the system, predicting the network status, and automatically configuring the policies.

C. Example of Supply Chain Assets Access Control

An overview of the SC abstraction proposed in our example is illustrated in Fig. 2. Organizations are in the top of the hierarchy and represent companies, consortiums or institutions that belongs to the supply chain. Realms consist of an environment within an organization that shares common security policies and rules. Domains are a set of assets with certain relationship, for instance, they share the same network, location or infrastructure.

1) Intent Language Definition: We define a simple grammar for SC asset access control as an intent language, based on a context-free grammar standard: the Extended Backus-Naur Form (EBNF) [30]. The defined grammar in Grammar 1 is a CNL that allows the system administrator to grant access to users for a certain asset in the system, restricted to specific organization, realm, or domain at a certain working time. This proposal is a general definition of the grammar, and can be enriched with more details related to the nature of an asset, the composition of a supply chain, the structure of a company, and finally the type of users.

2) Intent Manager: It parses the input plain text into a json file using the defined high-level language and regular

$\langle intent \rangle$	::=	$\langle users \rangle$ is $\langle permission \rangle$ to access to $\langle asset \rangle$ $\langle spot \rangle$ at $\langle timeframes \rangle$
$\langle users \rangle$	::=	$\langle user \rangle$ (, and) $\langle user \rangle$
$\langle user \rangle$	::=	[a-z0-9]+
$\langle permission \rangle$	::=	allowed | blocked
$\langle asset \rangle$	::=	domain | realm | organization
$\langle spot \rangle$	::=	[a-z0-9]+
$\langle timeframes \rangle$	::=	$\langle timeframe \rangle$ (, and) $\langle timeframe \rangle$
$\langle timeframe \rangle$	::=	$\langle morning\ shift \rangle$ | $\langle late\ shift \rangle$ | $\langle night\ shift \rangle$
$\langle morning\ shift \rangle$	::=	[6:00-13:59]
$\langle late\ shift \rangle$	::=	[14:00-21:59]
$\langle night\ shift \rangle$	::=	[22:00-5:59]

Grammar 1: SC Assets Access Control Intent

expressions (regex approach). The parsing process can be improved using ML techniques to allow the user to employ multiple vocabulary options. The intent manager reads the intent store from the knowledge base and try to match the existing vocabularies and structures with the input text. Finally a json file with user specifications is generated and sent to the policy configurator.

3) Policy Configurator: It matches the requirements specifications received from the intent manager with the existing policies stored in the policy store. It solves conflict between intents and generates valid policies. Conflicts can happen between policies while compiling intent specifications. A user can be restricted from access to assets in realm *o1-r1* using a following intent : *user-x is not allowed to access to realm o1-r1*. The policy configurator will map the intent into high level policies such as:

> *check user-x in database of Users*
> *block user-x to access assets in o1-r1*
> *alert admin in o1*

When the administrator wants to give the user access to assets in organization *o1*, the following intent can be written: *user-x is allowed to access to organization o1*. This intent will create a conflict with the previous intent during the policy translation, as it will allow the user to access to realm *o1-r1*, which is inside the organization *o1*. In order to solve this issue, conflict checking needs to be considered and implemented. A solution to the mentioned conflict can be translated to the following high-level policies:

> *check user-x in database of Users*
> *allow user-x to access assets in o1 except o1-r1*
> *alert admin in o1*

Other conflicts should be analysed considering the organizational structure, the existing policies and the possible human intervention.

4) Knowledge Base: It consists of three main components: intent store, policy store, and telemetry. The intent store is used to store a dictionary of vocabularies and options that are required to parse intents. The policy store represents the policy graphs defined by experts, in our case the policy graph represent the access to assets based on the organizational hierarchy. And finally the telemetry is used to store the needed metrics to detect anomalies in the system.

5) AI Engine: NLP techniques, such as long-short time memory (LSTM), can be implemented to enhance the proposed CNL and provide more flexibility to the user while writing the intents. The parsing of the input plain text considers a rich dictionary of vocabularies and expressions that can mean the same input specifications. AI techniques can also be used to enhance the policy configuration by solving policy conflicts. Metrics collected from system and stored in the knowledge base can be used to predict if a user has to be blocked or not based on his monitored behavior, and ML techniques are a great candidate for such task of pattern recognition and security assessment.

IV. CONCLUSION

This paper proposes a general design of an intent-based translation system that manages ICT supply chain networks, considering AI enabling technologies to ensure a valid translation of user requirements, network security and scalability. With our system design we intend to move the complexity of network management, which requires very high level of expertise from the system administrator by automating policy building process through AI and ML techniques. We analysed an use case example of an ICT supply chain management task - access control policy configuration using high level intents. Also, we discussed the engineering requirements and challenges in integrating IBN approach with ICT supply chains, in terms of network orchestration complexity, the application of AI techniques, and the ambiguity of defining high level intents for ICT supply chain policies.

REFERENCES

[1] E. Shry, "The state of intent-based networking," 2019. [Online]. Available: http://www.cse.wustl.edu/~jain/cse570-19/ftp/intent/index.html

[2] B. K. Saha *et al.*, "Intent-based networks: An industrial perspective," in *Proceedings of the 1st International Workshop on Future Industrial Communication Networks*, 2018, pp. 35–40.

[3] B. Laliberte, "The journey to intent-based networking: Ten key principles for accelerating adoption," 2018.

[4] H. Mahtout *et al.*, "Using machine learning for intent-based provisioning in high-speed science networks," in *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*, 2020, pp. 27–30.

[5] D. Comer and A. Rastegatnia, "Osdf: An intent-based software defined network programming framework," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, 2018, pp. 527–535.

[6] K. Abbas *et al.*, "Slicing the core network and radio access network domains through intent-based networking for 5g networks," *Electronics*, vol. 9, no. 10, p. 1710, 2020.

[7] T. A. Khan *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*. IEEE, 2020, pp. 1–2.

[8] M. Riftadi and F. Kuipers, "P4i/o: Intent-based networking with p4," in *2019 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2019, pp. 438–443.

[9] Y. Han *et al.*, "An intent-based network virtualization platform for sdn," in *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, 2016, pp. 353–358.

[10] A. Chaudhari *et al.*, "Vivonet: Visually-represented, intent-based, voice-assisted networking," *arXiv preprint arXiv:1904.03228*, 2019.

[11] M. Pham and D. B. Hoang, "Sdn applications-the intent-based north-bound interface realisation for extended applications," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE, 2016, pp. 372–377.

[12] M. Jain *et al.*, "Intent-based, voice-assisted, self-healing sdn framework," *Journal of Network Communications and Emerging Technologies (JNCET)*, vol. 10, no. 2, 2020.

[13] T. Subramanya *et al.*, "Intent-based mobile backhauling for 5g networks," in *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, 2016, pp. 348–352.

[14] C. Chung and J. P. Jeong, "A design of iot device configuration translator for intent-based iot-cloud services," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2020, pp. 52–56.

[15] A. Voellmy *et al.*, "Procera: A language for high-level reactive network control," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 4348. [Online]. Available: https://doi.org/10.1145/2342441.2342451

[16] E. J. Scheid *et al.*, "A controlled natural language to support intent-based blockchain selection," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2020, pp. 1–9.

[17] T. Szyrkowiec *et al.*, "Automatic intent-based secure service creation through a multilayer sdn network orchestration," *Journal of Optical Communications and Networking*, vol. 10, no. 4, pp. 289–297, 2018.

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

[19] J. Yang and J. P. Jeong, "An automata-based security policy translation for network security functions," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 268–272.

[20] P. Sköldström *et al.*, "Dismi-an intent interface for application-centric transport network services," in *2017 19th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2017, pp. 1–4.

[21] A. Singh *et al.*, "Intent-based network for data dissemination in software-defined vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[22] S. Amrutha *et al.*, "Voice controlled smart home," *International Journal of Emerging Technology and Advanced Engineering*, vol. 5, no. 1, pp. 272–275, 2015.

[23] Huawei, "Huawei intent-driven network white paper," 2019, accessed: 04.05.2021. [Online]. Available: https://bit.ly/3nOKPue

[24] OpenDayLight, "Network intent composition (nic) user guide," 2016, accessed: 04.05.2021. [Online]. Available: https://bit.ly/3tiN36b

[25] ——, "Group based policy user guide," 2016, accessed: 04.05.2021. [Online]. Available: https://bit.ly/3b26dXv

[26] onosproject, "Nemo language," 2016, accessed: 04.05.2021. [Online]. Available: https://bit.ly/3eShzOX

[27] A. M. Zarca *et al.*, "Semantic-aware security orchestration in sdn/nfv-enabled iot systems," *Sensors*, vol. 20, no. 13, p. 3622, 2020.

[28] R. Toorajipour *et al.*, "Artificial intelligence in supply chain management: A systematic literature review," *Journal of Business Research*, vol. 122, pp. 502–517, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014829632030583X

[29] L. Pang *et al.*, "A survey on intent-driven networks," *IEEE Access*, vol. 8, pp. 22862–22873, 2020.

[30] R. S. Scowen, "Generic base standards," in *Proceedings 1993 Software Engineering Standards Symposium*. IEEE, 1993, pp. 25–34.