Security Closed-Loop Automation in 6G Networks

Por

Pedro Miguel Vieira Ferreira

Orientador: Paula Cristina Ribeiro Coutinho de Oliveira

Co-orientador: João Paulo Fonseca da Costa Moura

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no
DR – I série–A, Decreto-Lei n.º 74/2006 de 24 de Março e no
Regulamento de Estudos Pós-Graduados da UTAD
DR, 2.ª série – Deliberação n.º 2391/2007

Security Closed-Loop Automation in 6G Networks

Por

Pedro Miguel Vieira Ferreira

Orientador: Paula Cristina Ribeiro Coutinho de Oliveira

Co-orientador: João Paulo Fonseca da Costa Moura

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Engenharia Electrotécnica e de Computadores, de acordo com o disposto no
DR – I série–A, Decreto-Lei n.º 74/2006 de 24 de Março e no
Regulamento de Estudos Pós-Graduados da UTAD
DR, 2.ª série – Deliberação n.º 2391/2007

Orientação Científica :

Paula Cristina Ribeiro Coutinho de Oliveira

Professora Auxiliar do
Departamento de Engenharias
Escola de Ciências e Tecnologia
da Universidade de Trás-os-Montes e Alto Douro

João Paulo Fonseca da Costa Moura

Professor Auxiliar do
Departamento de Engenharias
Escola de Ciências e Tecnologia
da Universidade de Trás-os-Montes e Alto Douro

Acompanhamento do trabalho :

Manuel José Torres Sousa da Cunha

Arquiteto de Software do
Departamento de Operações
Optare Solutions S.A.

"We become what we think about."

Earl Nightingale (1921 – 1989)

"The only constant in cybersecurity is change"

Kevin Mitnick(1963 – 2023)

-

Closed-Loop Automation in Next-Generation Networks

Pedro Miguel Vieira Ferreira

Submitted to the University of Trás-os-Montes and Alto Douro
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computers

Abstract — The advent of 5G and upcoming 6G technologies has revolutionised mobile networks, offering unprecedented speed, connectivity, and versatility. However, this evolution introduces complex security challenges that demand innovative solutions. This thesis presents the design, implementation, and evaluation of a Security Closed-Loop Automation (SCLA) system aimed at enhancing network security in modern telecommunications infrastructures. Focusing on the Security Decision (SD) component, the system integrates advanced inference engines, machine learning algorithms, and automation tools to enable real-time threat detection and mitigation, provider-level performance monitoring, and robust performance under high-stress conditions.

The SCLA system adopts a dual-layer architecture comprising domain-level and end-to-end (E2E) instances, facilitating comprehensive security management across various network domains such as EDGE, cloud, and RAN. The SD component employs both forward and backward chaining methodologies within a CLIPSPy-based inference engine, leveraging data from the Security Data Collection (SDC) and Security Data Analytics (SDA) components. The integration of Kafka for data streaming and Ansible for automated deployment enhances the system's scalability and adaptability.

Extensive testing was conducted to evaluate the system's effectiveness. In real-time threat detection at the domain level, the SD component achieved average inference times of approximately 6.7 milliseconds, successfully identifying and mitigating simulated Distributed Denial of Service (DDoS) attacks. For provider-level compliance monitoring, the E2E SCLA demonstrated 100

The results validate the SCLA system's capability to enhance network security through automated, intelligent responses to threats and performance deviations.

The modular architecture and innovative integration of technologies position the system as a viable solution for securing next-generation mobile networks. Limitations such as the focus on DDoS attacks and the use of synthetic data highlight areas for future research. Recommendations include expanding threat coverage, deploying the system in real-world environments, integrating advanced machine learning models, and enhancing scalability through parallel processing and resource optimisation. In conclusion, this thesis contributes to the field of network security by providing a practical, effective approach to automated security management in 5G and beyond networks. The SCLA system lays a solid foundation for ongoing research and development, addressing critical security challenges and supporting the evolution of secure, resilient telecommunications infrastructures.

Key Words: 5G security, 6G networks, Closed-Loop Automation (CLA), Network security, Real-time threat detection, Inference engine

Closed-Loop Automation in Next-Generation Networks

Pedro Miguel Vieira Ferreira

Submetido na Universidade de Trás-os-Montes e Alto Douro
para o preenchimento dos requisitos parciais para obtenção do grau de
Mestre em Engenharia Electrotécnica e de Computadores

Resumo — A chegada das tecnologias 5G e futuras 6G revolucionou as redes móveis, oferecendo velocidades, conectividade e versatilidade sem precedentes. Contudo, esta evolução introduz desafios de segurança complexos que exigem soluções inovadoras. Esta tese apresenta o design, implementação e avaliação de um sistema de Automação de Ciclo Fechado de Segurança (Security Closed-Loop Automation - SCLA) com o objetivo de aprimorar a segurança das redes nas infraestruturas de telecomunicações modernas. Focando-se no componente de Decisão de Segurança (Security Decision - SD), o sistema integra motores de inferência avançados, algoritmos de machine learning e ferramentas de automação para permitir a deteção e mitigação de ameaças em tempo real, monitorização de desempenho a nível de fornecedor e desempenho robusto sob condições de alto stress.

O sistema SCLA adota uma arquitetura de dupla camada composta por instâncias a nível de domínio e de ponta a ponta (End-to-End - E2E), facilitando uma gestão abrangente da segurança através de diversos domínios de rede como EDGE, cloud e RAN. O componente SD emprega metodologias de encadeamento direto e inverso dentro de um motor de inferência baseado em CLIPSPy, aproveitando dados dos componentes de Colecção de Dados de Segurança (Security Data Collection - SDC) e Análise de Dados de Segurança (Security Data Analytics - SDA). A integração do Kafka para streaming de dados e do Ansible para implementação automatizada aumenta a escalabilidade e adaptabilidade do sistema.

Foram realizados testes extensivos para avaliar a eficácia do sistema. Na deteção de ameaças em tempo real a nível de domínio, o componente SD alcançou tempos médios de inferência de aproximadamente 6,7 milissegundos, identificando e mitigando com sucesso ataques simulados de Distributed Denial of Service (DDoS).

Para a monitorização de conformidade a nível de fornecedor, o SCLA E2E demonstrou uma precisão de deteção de 100% na identificação de fornecedores de serviços não conformes, aplicando prontamente ações corretivas com impacto mínimo na continuidade do serviço. Sob condições de alto stress, o componente SD manteve um desempenho consistente, processando até 10.000 relatórios por segundo sem degradação significativa, confirmando assim a escalabilidade e robustez do sistema. Os resultados validam a capacidade do sistema SCLA de aprimorar a segurança das redes através de respostas automatizadas e inteligentes a ameaças e desvios de desempenho. A arquitetura modular e a integração inovadora de tecnologias posicionam o sistema como uma solução viável para a segurança das redes móveis de próxima geração. Limitações como o foco em ataques DDoS e o uso de dados sintéticos destacam áreas para pesquisas futuras. As recomendações incluem a expansão da cobertura de ameaças, a implementação do sistema em ambientes reais, a integração de modelos avançados de machine learning e o aumento da escalabilidade através do processamento paralelo e otimização de recursos.

Em conclusão, esta tese contribui para o campo da segurança de redes ao fornecer uma abordagem prática e eficaz para a gestão automatizada da segurança em redes 5G e futuras. O sistema SCLA estabelece uma base sólida para pesquisas e desenvolvimentos contínuos, abordando desafios críticos de segurança e apoiando a evolução de infraestruturas de telecomunicações seguras e resilientes.

Palavras-Chave: Segurança 5G, Redes 6G, Automação de Ciclo Fechado (CLA), Segurança de Redes, Deteção de Ameaças em Tempo Real, Motor de Inferência

Acknowledgements

I would like to begin by thanking my family for supporting me and providing me with the opportunity to pursue and reach this step in my educational path, even through the hardships it has brought them.
I would also like to thank the people involved in this dissertation/project. These include my advisors Dr. Paula Cristina Ribeiro Coutinho de Oliveira and Dr. João Paulo Fonseca da Costa Moura as well as my advisor Eng. Manuel José Torres Sousa da Cunha which also acted as my tutor during the internship that began this process.
I am also immensely grateful to my team at Optare Solutions, from the highest-ranking senior, to my colleagues "in the trenches", that guided me throughout this project and helped me develop into a better professional/researcher.
Last but not least, I am also thankful to Optare Solutions for their belief and generous support towards me, which allowed me the opportunity to work on a project like this.

UTAD, Pedro Ferreira
Vila Real, 06 de Novembro de 2024

Funding

This work has been partially funded by the "Ministerio de Asuntos Económicos y Transformación Digital" and the European Union-NextGenerationEU in the frameworks of the "Plan de Recuperación, Transformación y Resiliencia" and of the "Mecanismo de Recuperación y Resiliencia" under reference 6G-OPENSEC-SECURITY (TSI-063000-2021-58).

General Index

Figure Index

Glossary

Glossary

Acronyms List

Sigla	**Expansão**
AI	*Artificial Intelligence*
ANNs	*Artificial Neural Networks*
CLARA	*Closed-Loop Automation for Resilient Architectures*
CLA	*Closed-Loop Automation*
CLIPSPy	*CLIPS Python Binding*
DBMS	*Database Management System*
DSO	*Domain Slice Orchestrator*
E2E	*End-to-End*
ETSI	*European Telecommunications Standards Institute*
FQDN	*Fully Qualified Domain Name*
HLA	*High-Level Architecture*

Sigla	**Expansão**
IBN	*Intent-Based Networking*
ISPs	*Infrastructure Service Providers*
KPI	*Key Performance Indicator*
MANO	*Management and Orchestration*
ML	*Machine Learning*
MEC	*Multi-access Edge Computing*
MAPE-K	*Monitor-Analyze-Plan-Execute-Knowledge*
NBI	*Northbound Interface*
NFV	*Network Function Virtualization*
NMS	*Network Management System*
NS	*Network Slicing*
OPNFV	*Open Platform for NFV*
ONF	*Open Networking Foundation*
PII	*Policy Intent Inference*
PoC	*Policy as Code*
PBN	*Policy-Based Networking*
RAN	*Radio Access Network*
RSA	*Resilient Slice Architecture*
SCL	*Security Closed-Loop*
SCLA	*Security Closed-Loop Automation*
SD	*Security Decision*
SDN	*Software-Defined Networking*
SDO	*Standard Development Organizations*
SLA	*Service Level Agreement*
SLO	*Service Level Objective*
SSLA	*Security Service Level Agreement*
SVM	*Support Vector Machine*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
VNF	*Virtual Network Function*
YANG	*Yet Another Next Generation* (data modeling language)
ZSM	*Zero-touch Network Service Management*

Sigla	**Expansão**
ZTP	*Zero Touch Provisioning*

Introduction

1.1 Background and Motivation

The swift advancement of mobile network technologies from 4G to 5G and the impending 6G has revolutionised the telecommunications landscape. These advancements have brought unprecedented enhancements in data transmission speeds, reduced latency, and the capacity to connect a myriad of services. Technologies such as Software Defined Networking (SDN), Network Function Virtualisation (NFV), Network Slicing, Multi-access Edge Computing (MEC), and Zero-touch Network and Service Management (ZSM) have emerged as key enablers in this transformation, offering modularity, scalability, and flexibility in network design, management, and provisioning.

However, this technological progression has also introduced a new array of complex security challenges. The increasing sophistication of network architectures, coupled with the proliferation of connected devices and services, has expanded the attack surface for malicious actors. Traditional security mechanisms are often inadequate to address the dynamic and multifaceted threats targeting modern networks. Therefore, there is an imperative need for innovative security solutions that can adapt to

the evolving landscape.

This thesis is developed within the context of an ongoing European Union-funded project, called OPENSEC, aimed at enhancing network security in 5G and beyond networks. The project focuses on developing a Security Closed-Loop Automation (SCLA) system that leverages advanced technologies to provide robust, real-time security solutions across various network domains, including the Cloud, Radio Access Network (RAN), and Transport networks.

1.2 Problem Statement

As networks become more complex and dynamic, ensuring security compliance and threat mitigation in real-time has become a significant challenge. Traditional reactive security measures are insufficient for the rapid detection and response required in modern networks. Additionally, the integration of multiple technologies and domains necessitates a unified approach to security management that can operate effectively across different layers and services.

Specific challenges addressed in this thesis include:

- **Real-time Threat Detection and Mitigation:** The need for systems capable of detecting and responding to security threats, such as Distributed Denial of Service (DDoS) attacks, in real-time to prevent service disruption.
- **Provider-level Performance Monitoring:** Ensuring that service providers comply with defined Service Level Agreements (SLAs) and Service Level Objectives (SLOs), and effectively managing non-compliance to maintain service quality.
- **Handling high-stress network scenarios:** Ensuring that the SCLA is able to maintain policy compliance and acceptable performance, while handling high data throughput within the simulated scenario.

1.3 Objectives

The primary objective of this thesis is to design, implement, and evaluate a Security Closed-Loop Automation (SCLA) system that enhances network security by addressing the aforementioned challenges. The specific objectives are:

1.3.1 Develop a System for Real-time Threat Detection and Mitigation:

- Implement the Security Decision (SD) component capable of detecting network anomalies and security threats using forward chaining inference.
- Integrate with Security Data Collection (SDC) and Security Data Analytics (SDA) components to process network data efficiently.

1.3.2 Ensure Provider-level Performance Monitoring and Compliance:

- Design an End-to-End (E2E) SCLA that monitors service provider performance metrics.
- Implement mechanisms to enforce SLAs/SLOs and penalise or switch providers based on performance.

1.3.3 Ensure Performance and Compliance in a stress test scenario:

- Incorporate efficient threading and batching for the processing of high data throughput.
- Take advantage of CLIPSPy's inference performance along with efficient data processing to mitigate threats at acceptable speeds in a DDoS scenario.

1.4 Methodology

To achieve these objectives, in practical and conceptual terms, the thesis follows a structured approach:

- **Literature Review:** An extensive review of existing technologies and frameworks, including SDN, NFV, Network Slicing, MEC, ZSM, and Closed-Loop Automation, to establish a foundational understanding and identify gaps in current solutions.
- **Proposed Model Development:** Designing the SCLA system architecture based on the Zero-touch Network and Service Management principles, outlining the interactions between components such as the SD, SDA, SDC, and the Security and Privacy Data Service (SPDS).
- **Implementation:** Developing the SD component using Python-based tools, implementing forward and backward chaining mechanisms using CLIPSPy, and integrating with other components using Kafka brokers and APIs for data communication.
- **Testing and Evaluation:** Conducting targeted tests to evaluate the system's effectiveness in real-time threat detection, provider-level compliance monitoring, and the integration of ML and rule-based inference.

1.5 Contributions

The contributions of this thesis can be described as follows:

- **Innovative SCLA System:** Introduces a novel SCLA System that leverages advanced ML integration, real-time policy enforcement, and closed-loop feedback mechanisms to enhance network security.

- **Enhanced SD Component:** Develops a sophisticated SD component capable of both forward and backward chaining inference, that can be integrated with ML algorithms for improved threat detection, and is also capable of managing SCLA system configuration and deployment at the Domain/E2E level.
- **E2E SCLA Implementation:** Implements an E2E SCLA for comprehensive provider-level performance monitoring, ensuring compliance with SLAs/SLOs across multiple domains.
- **Practical Testing Framework:** Establishes a testing framework aligned with key research questions, providing a basis for evaluating the system's effectiveness and guiding future developments.

1.6 Thesis Structure

The remainder of this thesis is organised in the following manner:

- **Chapter 2: Literature Review** – Provides an in-depth analysis of the key technologies and concepts that underpin the proposed system, including SDN, NFV, Network Slicing, MEC, ZSM, and Closed-Loop Automation.
- **Chapter 3: Proposed Model** – Describes the architectural design of the SCLA system, detailing the functionalities of each component, particularly the SD component, and illustrating the workflows for threat detection and response.
- **Chapter 4: Implementation** – Discusses the practical aspects of developing and deploying the proposed model, including the tools and technologies used, the reasoning behind design choices, and the integration of components within the SCLA.
- **Chapter 5: Testing and Results** – Presents the testing framework, procedures, and results, analysing the system's performance in relation to the key research questions.

- **Chapter 6: Conclusion and Future Work** – Summarises the findings, discusses the implications of the research, and outlines potential areas for future development, considering the ongoing nature of the project.

1.7 Scope and Limitations

While the thesis aims to provide a comprehensive solution to network security challenges in modern telecommunications, certain limitations exist due to the ongoing nature of the project:

- **Incomplete Feature Integration:** Some features and real-world testing scenarios are still under development and are not fully integrated into the current system.
- **Focus on Specific Threats:** The testing primarily focusses on DDoS attacks, and while the system is designed to handle a range of threats, further testing is required for other attack vectors.
- **Simulation Environment:** Due to constraints, some tests are conducted in simulated environments, which may not capture all real-world complexities.

These limitations provide opportunities for future work to further enhance and validate the system.

1.8 Summary

In summary, this thesis addresses critical security challenges in modern mobile networks by developing a robust and adaptive SCLA system. By integrating advanced technologies and adhering to ZSM principles, the proposed system offers a scalable and flexible solution capable of real-time threat detection and response. The research contributes valuable information on the practical application of closed-loop

automation in network security, laying the foundation for future advancements in the field.

2 Literature Review

This section of the document will focus on reviewing the technologies that serve or can serve as support for systems such as the one envisioned for implementation, the SCL. This review will be based on existing literature. Some of it will be with a higher focus on the technologies themselves and their inner workings, while the remaining will be focused on literature that demonstrates or infers about possible applications of these technologies that may serve our system in the next generation of mobile network technologies. As the section progresses, the scope of what is being discussed shall continuously narrow, until reaching literature that is as closely related as possible to the model proposed in the next section, after being given the conceptual context of subjects or technologies upon which it is built upon.

For better visualization of the subjects being discussed in this section refer to the following Fig. 2.1:

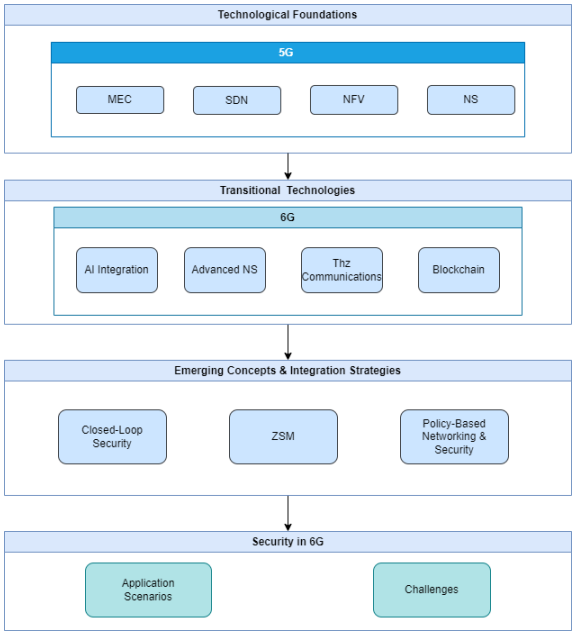

Figure 2.1 – Chapter 2 guideline

2.1 Evolution of and up to 5G

As the basis generation of telecommunications for most of the technologies used to achieve what shall be proposed with this work, the fifth generation of mobile networks (5G) was and is a significant leap from the previous generation, offering increased speeds, capacity, and the ability to connect more devices than before. It enables the enhancement of use cases such as cloud storage, augmented reality, AI, and advances IoT capabilities through reduced latency and edge computing [Sufyan et al., 2023]. It is known through the progression of these technologies that in cellular networks, the concept of frequency reuse is very common, because as the networks evolve, so does the demand for them in terms of the number of users and applications that utilize these networks. As well as frequency reuse, it is very common to find that the key enablers that serve as a foundation for the previous mobile network generation are often redefined, built upon, or in some cases even reused with enhanced capabilities in the following one. With that in mind and to start this chapter with a broad concept that touches upon many of the

subsequent subjects, the key enablers and technological foundations that separate this generation of mobile technology from the previous ones shall now be iterated over before beginning to disseminate what is expected and what will be leveraged to achieve what is believed to be the vision of the subsequent generation (6G).

2.2 Software-Defined Networking

Software-Defined Networking (SDN) is an innovative network paradigm that decouples the network control and its forwarding functions, thus enabling network administrators to directly program network control decisions by using a centralised and programmable SDN controller or controllers. This separation of the control and data planes allows for a more flexible network management, optimises resource utilisation and reutilisation, as well as a faster deployment of new services than in traditional networks, where it would be necessary to configure most of the network devices manually [Kreutz et al., 2015, Xia et al., 2015]. The architecture of SDN networks is made up of three main layers:

- Application Layer: contains the network applications or services that communicate with the SDN controller via northbound APIs[Tuncer et al., 2018].
- Control Layer: hosts the SDN controller, which acts as the brain of the network, providing a centralised view of the entire network and making decisions about traffic flow within it.
- Infrastructure Layer: consists of network devices (such as switches and routers) that execute the traffic forwarding decisions made by the SDN controller, communicating via southbound APIs like OpenFlow[Allied Telesis, 2022].

SDN shares many of the principles of other technologies that support the fifth generation of mobile networks and will most likely transition to the next, by allowing dynamic, programmable networking, which in turn facilitates network automation and orchestration while enhancing network security through flexible control mechanisms.

It is particularly beneficial in environments such as complex, scalable cloud-based networks and next-generation data centres where the traditional network architectures may not be able to fulfill these network's requirements. The work presented by Rafique et al. [2020] support this definition of the concept and demonstrates how it can be combined with edge computing to complement IoT services by enhancing network service provisioning and management. Since this kind of technology has a characteristic network architecture as has been previously described, the Fig. 2.2 should help visualise the layers which comprise the architecture and how they are interconnected as well as what types of components, be it hardware or software, present within each layer.

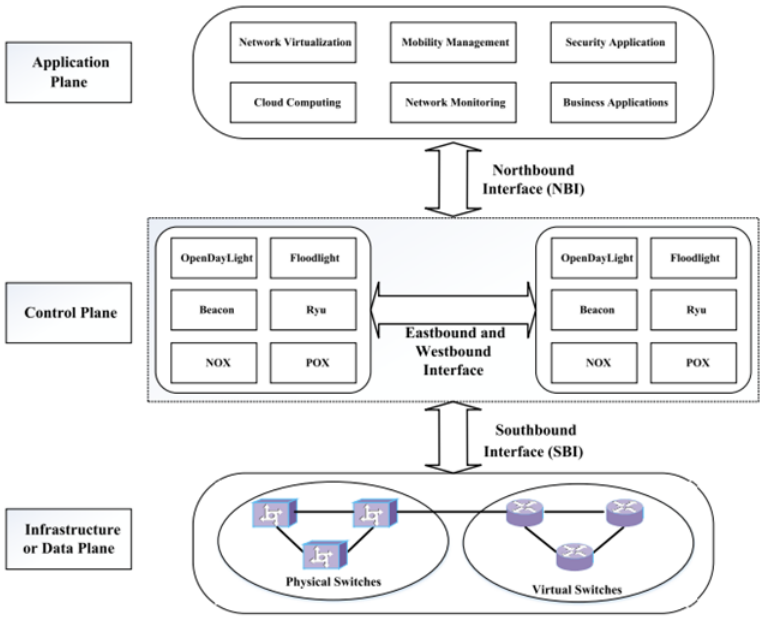

Figure 2.2 – General SDN architecture [Xie et al., 2019]

For comparison the Table 2.1 displays a few of the different characteristics between the SDN architecture and classical network architectures:

Despite the changes that SDN enables, through a more modular network infrastructure and decoupling of functionalities through different layers, it is always important to analyse the issues that may arise from the implementation of new technologies. And since the focus of this work will be security, the article by Dabbagh et al. [2015]

Characteristics	SDN architecture	Classical architecture
Programmability	✓	
Centralized control	✓	
Error-prone configuration		✓
Complex network control		✓
Network flexibility	✓	
Improved performance	✓	
Easy implementation	✓	
Efficient configuration	✓	
Enhanced management	✓	

Table 2.1 – Comparison of SDN and Classical Architecture Characteristics [Benzekki et al., 2016]

provides an extensive analysis of the security implications associated with SDN. The key points to take from this previously cited work are that in terms of security advantages, SDN provides the following:

- Centralized Control: By centralising control, SDN can provide a comprehensive overview of the network, which enhances monitoring, management, and security enforcement.
- Improved Response to Threats: The centralized nature allows for quicker response and adaptation to security threats across the network.
- Network Behavior Flexibility: SDN's ability to reprogram network traffic dynamically at runtime is highlighted as a critical asset for implementing security policies and network segmentation.

However, as it is common, no technology is flawless, therefore it was disseminated in this work that SDN also brings forth some new challenges in terms of security:

- New Attack Surfaces: The centralisation of control also introduces new risks, such as the potential for single points of failure if the SDN controller is compromised.

- Scalability and Security: Concerns are raised about the scalability of security solutions as networks grow larger and more complex.

- Interoperability with Legacy Systems: The article by Dabbagh et al. [2015] addresses the challenges of integrating SDN with existing network infrastructures without compromising security standards.

2.3 Network Function Virtualisation

Network Function Virtualisation (NFV) is an emerging technology paradigm poised to revolutionise the way telecommunication services are provisioned and managed. By decoupling network functions from the physical hardware on which they traditionally operate, NFV allows these functions to be hosted on virtual machines or containers across various locations, including data centres, network nodes, or even end-user premises. This abstraction from physical devices leads to more flexible and efficient deployment [Mijumbi et al., 2016] and scaling of network services, which can be dynamically adjusted to meet fluctuating demands.
To internalise some of the differences between older conventional networks and the new generation of network architecture, Figure 2.3 showcases a clear comparison between conventional, SDN and NFV based architectures:

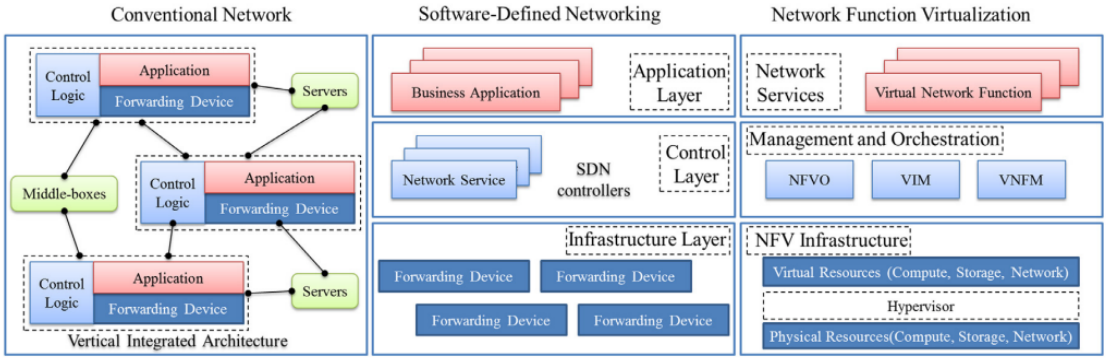

Figure 2.3 – Comparison between conventional, SDN and NFV networks [Mijumbi et al., 2016]

Notice the pattern of modularity in both the SDN and NFV network architecture, which follows accordingly to the decoupling of functions mentioned before.
The core idea of NFV is to utilise standard IT virtualisation technology to consolidate many types of network equipment onto high-volume servers, switches, and storage platforms. This not only reduces equipment costs and power consumption, but also provides greater agility in deploying and scaling network services. NFV encompasses a variety of functions, such as firewalls, load balancers, and intrusion detection systems, which can be rapidly deployed, managed, and decommissioned as virtual instances. This flexibility significantly shortens the service innovation cycle, allowing telecom providers to offer new services faster and with reduced capital expenditure[Mijumbi et al., 2016]. Furthermore, NFV is closely associated with the management and orchestration (MANO) of network services, which is vital for maintaining the performance, reliability, and security of virtual network functions (VNFs). MANO frameworks help in the deployment, configuration, and management of VNFs and their interactions with the physical network infrastructure. By providing a standardised approach to VNF management, NFV facilitates a more cohesive and unified management environment, supporting complex multi-vendor network architectures and promoting interoperability and open standards. This strategic shift towards a virtualised networking environment promises significant operational efficiencies and paves the way for more adaptive and innovative network management practices. Further elaboration on the topic is present in Yi et al. [2018a] which discusses the standardised NFV architecture. This architecture is vital for understanding how different components interact within the NFV framework to support the efficient deployment and operation of network services. In addition, it tackles some of the challenges present in NFV, particularly in integrating NFV with other cutting-edge technologies such as IoT and 5G. The survey also explores the algorithms associated with VNFs, such as placement, scheduling, migration, chaining, and multicast, which are all operations required to be executed effectively for efficient VNF deployment within a network. Lastly, in this work is also discussed the role of standardisation and open source initiatives provided by standard development organizations (SDOs) like ETSI, ONF, and OPNFV which are crucial for ensuring interoperability and support a wide adoption of NFV solutions across

different vendors and platforms. The challenge of VNF placement is also disseminated in the work of Laghrissi and Taleb [2019] and the placement is categorised into dynamic and static placement. The latter is simpler and a more stable placement of VNFs, but lacks the flexibility of dynamic VNF placement, which can adapt to changing network conditions and workload demands. It all comes down to efficient resource allocation and optimisation of virtual resources to meet the performance requirements of different network services, and this article discusses the possibility of using slice-specific service levels, which combines network slicing and NFV in a placement strategy that can efficiently scale up or down resources or move the VNFs across different slices dynamically, without impacting the performance across the network infrastructure. In this work the need for secure virtualisation platforms is also emphasised, since VNFs rely heavily on the underlying virtualisation infrastructure, effective isolation between different VNFs, preventing issues in one VNF from affecting others, and the potential vulnerabilities within the management and orchestration layers.

2.4 Network Slicing

Exploring the advancements in 5G and beyond technologies, network slicing (NS) emerges inevitably as one of the core innovations. It was proposed as a solution to overcome limitations such as previous network architectures and the cost of hardware upgrades which were required to advance in terms of higher data rates, improved reliability, and reduced latency. It is a pivotal technology in 5G, designed to address the increasing complexity and variety of requirements in modern telecommunications. At its core, network slicing allows the partitioning of a single physical network into multiple virtual networks, each tailored to meet specific service requirements, such as low latency or high throughput. This flexibility is essential as 5G networks support a diverse range of applications from massive IoT deployments to critical communications and high-speed mobile broadband [Zhang, 2019].
In our previous study, we proposed a comprehensive framework that leverages these technologies to proactively detect and mitigate security threats within network slices

[Cunha et al., 2024]. This technology leverages advances in SDN and NFV to dynamically manage and orchestrate network resources. By, as has been common with these networking technologies, decoupling the network's control and user planes, SDN and NFV enable the efficient creation and management of virtualised network functions, allowing each slice to behave as an independent network. This is a critical capability in a context where network demands are not only about higher data rates but also about providing customised connectivity solutions that can adapt to varying performance needs [Subedi et al., 2021].

In practical terms, network slicing supports numerous applications, enhancing the network's ability to meet specific requirements for different use cases such as virtual reality, autonomous driving [Khan et al., 2018], smart cities [Zhou et al., 2020], and remote healthcare [Kapassa et al., 2019]. Each application can benefit from a network slice that provides the appropriate connectivity, data throughput, and security level. For instance, a slice allocated for IoT might prioritize energy efficiency and wide coverage, while a slice for autonomous vehicles would focus on ultra-reliable and low-latency communications [Subedi et al., 2021].

The integration of network slicing with technologies such as NFV is noted as essential in transitioning from traditional hardware-based network functions to more adaptable, cloud-computing based virtual environments [Afolabi et al., 2018]. This combination of technologies is very common in most literature discussing NS, as is in the work proposed by Taleb et al. [2020], where an illustration of multi-domain network slicing can be found depicting how both SDN and NFV are employed in network slicing as seen in Figure 2.4 below:

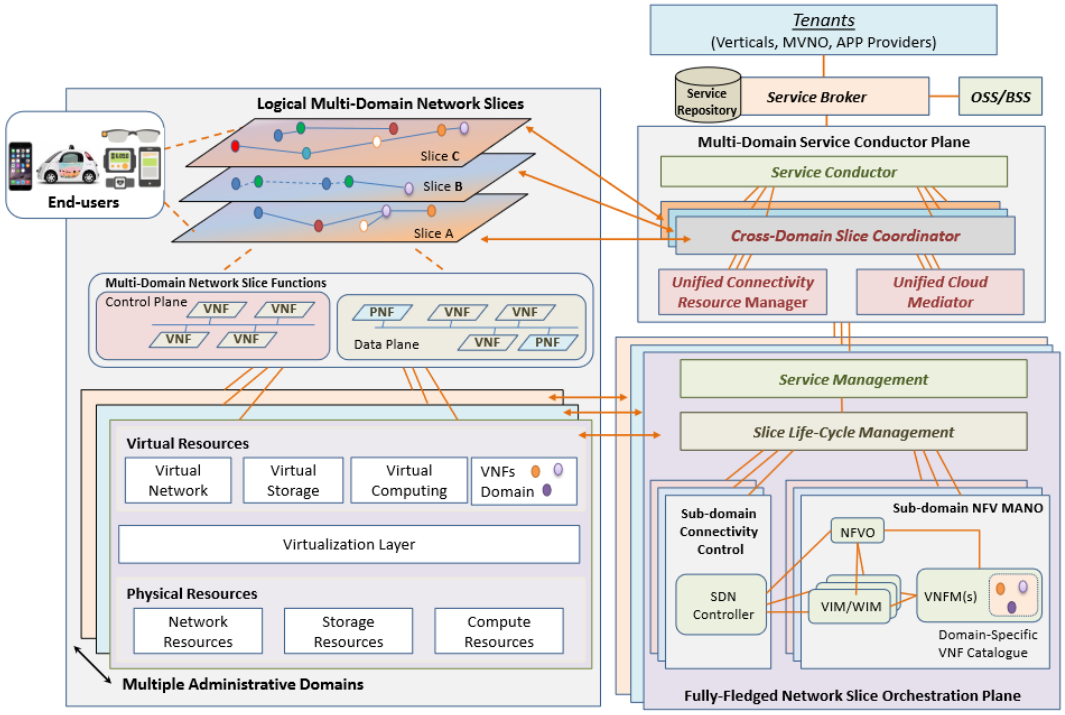

Figure 2.4 – Multi-domain slicing architecture [Taleb et al., 2020]

Since the depicted architecture employs not only network slicing but also technologies such as SDN, NFV, and MANO, it helps visualise how what has been discussed up until now in this chapter is connected. For better contextualisation, some of the components present in Figure 2.4 will now be described as presented. We observe that most of the architecture is divided into stratums for a more modular framework:

- **Multi-domain Service Conductor Stratum** (middle right column): This stratum is responsible for mapping all service requirements of the different multi-domain requests to their respective administrative domains.
- **Domain-specific fully-fledged Stratum** (bottom right column): This stratum is responsible for allocating internal domain resources for the establishment of a federated network slice instance (NSI) and providing the corresponding Life-Cycle Management (LCM) using the service management.
- **Service Broker Stratum** (top right column): Used as a service broker in

the functional plane to handle all incoming service requests from application providers, MVNOs (Mobile Virtual Network Operators) and other different verticals. It is responsible for managing NSI revenue, which involves charging and billing of slice owners, and also performs network slicing admission control and negotiation.

- **Sub-domain Infrastructure Stratum** (bottom left column): Consists of both the physical and virtual infrastructure (VNFs, virtual resources, virtualisation layer, and physical infrastructure).

In relation to the Life Cycle Management presented in Figure 2.4, the work of Zhang [2019] shows how this process happens, as depicted in the following Figure 2.5:

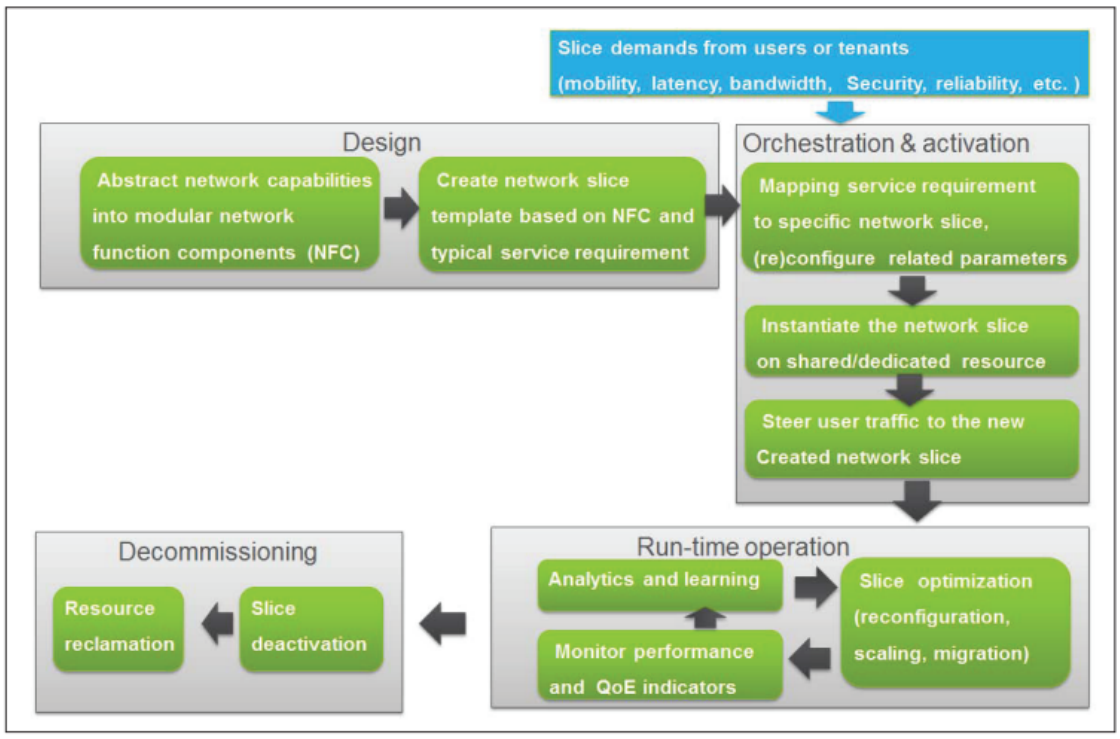

Figure 2.5 – Lifecycle management of network slices [Zhang, 2019]

As can be seen, the lifecycle of a network slice is divided into four phases:

1. **Design**: This initial phase involves the creation of a catalog of Network Function Components (NFCs), which serve as the building blocks for network slices. As new service requirements emerge and network capabilities evolve, this catalog is updated. Network slice templates, which include the necessary configurations, are created from these NFCs. A forwarding graph, which details how the NFCs are related and interact, is also generated during this phase.

2. **Orchestration and Activation**: During this phase, specific network slice templates are selected based on the service requirements from users or tenants. This selection process involves mapping these requirements to a slice template that supports the necessary performance, such as capacity and Quality of Service (QoS). Network Function components are instantiated within cloud infrastructures, and virtual connections are established as per the forwarding graph to ensure the components are properly linked.

3. **Run-Time Assurance**: In this phase, the performance indicators of the network slice are continually monitored. These indicators include metrics such as network function load, resource utilisation, and QoS parameters. Tools such as big data analytics and machine learning are employed to analyse the performance data. The insights gained from this analysis are used to make adjustments to the NSI, to maintain or improve service quality, ensuring compliance with the service-level agreement.

4. **Decommissioning**: The final phase involves the deactivation of an active NSI when it is no longer needed, based on changes in service requirements from tenants or shifts in the business strategy of operators. The resources that were allocated to the slice are then released and can be repurposed for other network slices or services.

After a thorough analysis on the challenges that can arise from implementing network slicing, a pattern similar to both SDN and NFV technologies was discovered. In the article of Barakabitze et al. [2020] this pattern is present and documented as the following challenges:

- **Integration Complexity**: As network slicing involves integrating SDN, NFV, and other emerging technologies, there is a complexity associated with achieving seamless integration. This includes technical challenges related to interoperability between different network functions and services that need to coexist on the same physical infrastructure.

- **Security and Isolation**: Ensuring robust security and effective isolation between slices is a major challenge. Each slice may have different security requirements and threat models, which necessitates dynamic security mechanisms that are adaptable to the specific needs of each slice.

- **Management and Orchestration**: Efficient management and orchestration of network slices are crucial, particularly as networks increase in complexity with the addition of new services and technologies. This involves automating the lifecycle management of slices to ensure they meet specific service levels and criteria without human intervention.

- **Scalability and Flexibility**: As the demand for more customised and service-specific network slices increases, the network architecture must be scalable and flexible enough to accommodate these changes. This includes the ability to dynamically modify slices in real-time based on changing service demands and network conditions.

- **Standardization**: There is a need for standardised frameworks and protocols to manage network slicing across different vendors and operators. This would ensure compatibility and ease the integration of diverse systems and components involved in the slicing process.

- **Economic and Business Models**: Developing viable business models that can capitalise on the capabilities of network slicing is a challenge. This includes pricing models that can reflect the value provided by different slices and determining how resources are allocated and billed.

2.5 Multi-access Edge Computing

Multi-access Edge Computing (MEC) was first introduced as Mobile Edge Computing in 2014 by the European Telecommunications Standards Institute (ETSI). The concept was presented as a means to provide cloud computing capabilities and an IT service environment at the edge of the network. This approach was particularly pertinent to the field of Internet of Things (IoT), which relies mostly on applications that demand minimal latency and high bandwidth at the edge of networks. The MEC improves the performance of wireless networks by optimising the delivery of network-based services and reducing traffic loads on the network core. By processing data closer to the source of information, MEC supports a variety of real-time applications and services across diverse domains, including smart cities, industrial automation, and healthcare, which are increasingly dependent on rapid data processing capabilities [Abbas et al., 2020b].

The integration of MEC with IoT devices offers significant operational benefits, specifically in enhancing the efficiency of data processing and the responsiveness of IoT systems. In smart cities, for instance, MEC can facilitate complex applications such as traffic management systems and real-time public safety monitoring by allowing data to be processed locally at the edge of the network. This not only reduces the latency involved in sending data back and forth to a centralised cloud, but also optimises the network's bandwidth usage. Moreover, MEC provides an effective solution for IoT scenarios characterised by a vast amount of data generation, necessitating quick decision-making based on real-time data analysis [Porambage et al., 2018].

However, the deployment of MEC within such expansive networks introduces substantial security and privacy challenges. The decentralisation of data processing to numerous edge nodes can complicate the security architecture of networks, making each node a potential target for security breaches. Thus, implementing robust security measures that ensure data integrity and protection against breaches is crucial [Mansouri and Babar, 2020].

Furthermore, the privacy of user data processed at these edge nodes must be guaranteed to build trust and ensure compliance with global data protection regulations. Addressing these challenges requires a comprehensive approach to security that encompasses the entire network, from the edge device to the core processing centres.

2.6 Zero-touch Network and Service Management

Currently, within the european telecommunications ecosystem, there's an incentive to develop networks according to the Zero-touch Network Service Management (ZSM) standard, defined by ETSI. This type of network architecture follows an extensive and comprehensive framework for the automation of operations in 5G and beyond networks. The key concept introduced by this framework is the minimisation of human intervention within the network's management. This is achieved by leveraging advanced technologies that are in constant development, such as ML and Deep Learning (DL). Taking this approach in the current and upcoming mobile network environments is crucial, since these are becoming increasingly complex and dynamic, requiring rapid, scalable, and efficient management solutions.

In the 5G environment, ZSM introduces a service-based architecture, where different network functions are modular, and interact through well-defined interfaces (supported by NFVs) as can be seen in Fig. 2.5. These interactions can be executed between different communication domains, and to avoid misinterpretation of the architecture's depiction, it is important to note that the integration fabric provides the following cross-domain services:

- **Registration/de-registration of management services;**

- **Discovery of the registered management services and the means to access them;**

- **Means of supporting the invocation of management services;**

- **Means of supporting synchronous and asynchronous communication;**

Beyond supporting scalability, flexibility, and the rapid deployment of services facilitated by the use of NFV, ZSM also emphasises the importance of closed-loop automation. Automation by itself does increase the response speed, not only in terms of deployment but also in configuration and re-configuration of services. However, without a means to regulate and evaluate the automated tasks, there is no way to guarantee a reliable service management. This is where concepts such as closed-loop automation are key, and will be discussed further along this section for better contextualisation of the research conducted.

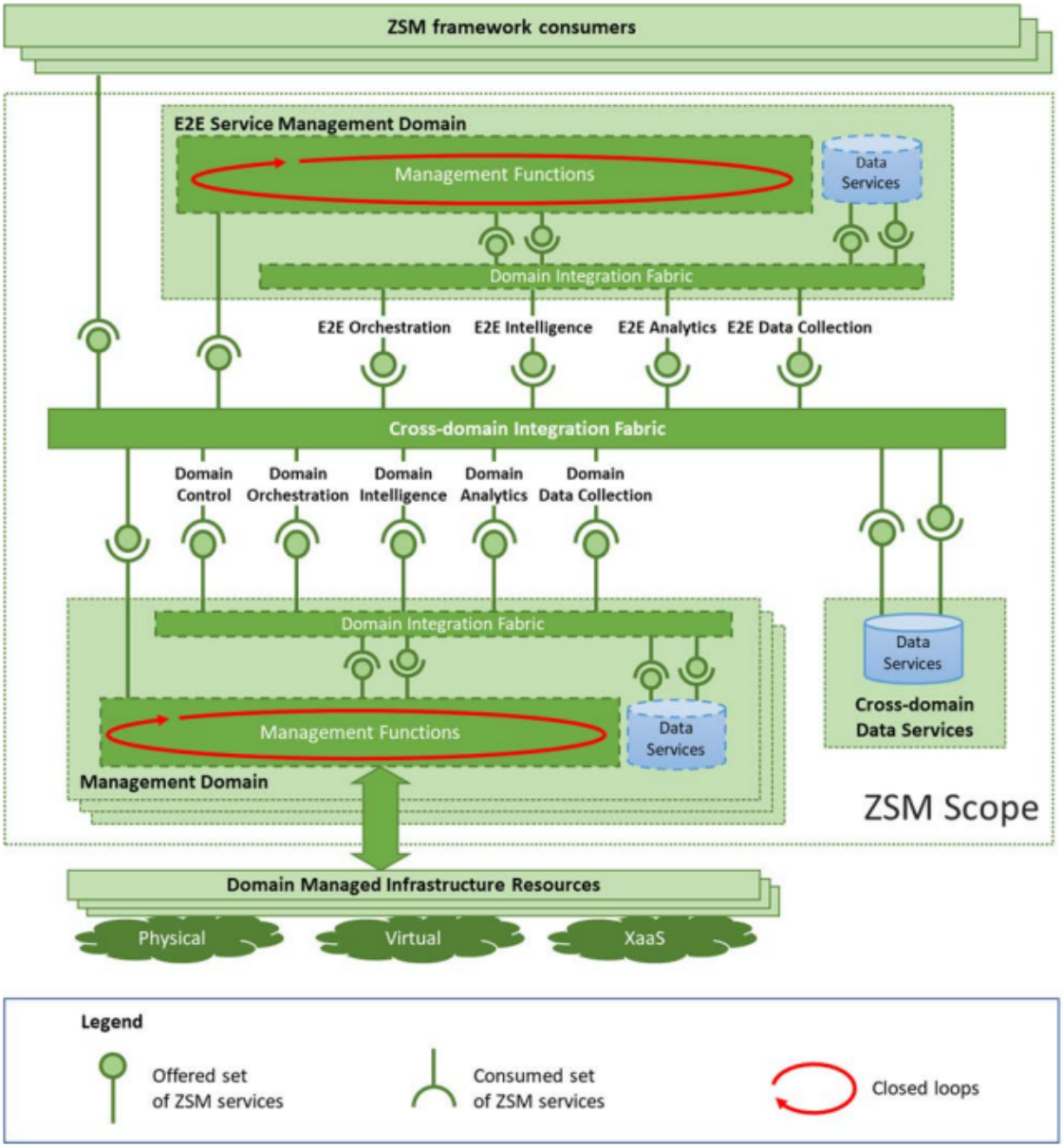

Figure 2.6 – ZSM architecture as presented in [European Telecommunications Standards Institute, 2019]

Despite all its inherent advantages, none of the technologies studied is faultless, and ZSM is no exception. The deployment of ZSM architectures requires the combination of both SDN, NFV and even ML/DL technologies to be integrated into the network for it to fulfill its automation requirements. A study conducted by Benzaid and

Taleb [2020], analyses the threat surface of the ZSM's framework, and their analysis is crucial for a better understanding of the risks associated with the development of networks that follow this architectural paradigm, built on fully automated network environments.

Building on this, the paper Chollon et al. [2022a] presents a comprehensive framework driven by the ZSM paradigm, integrated within the High-Level Architecture (HLA) developed in the INSPIRE-5Gplus[1] project. This framework is designed to enhance security management in future networks by leveraging smart 5G security methods and techniques essential for achieving robust security.

The following Figure 2.6, taken from this paper, describes the reference architecture of the aforementioned framework. It contains a few key elements that facilitate employ its security functionalities, namely:

- **Security Data Collector**: Gathers data from the network infrastructure.
- **Security Analytics Engine**: Augments the collected data with additional information and correlations.
- **Decision Engine**: Determines the appropriate mitigation actions based on the event and current infrastructure state.
- **Policy and SSLA (Security Service Level Agreement) Management**: Adapts security mitigations according to predefined security policies.
- **Security Orchestrator**: Enforces the determined security actions within the network infrastructure.
- **Integration Fabric**: Facilitates communication and coordination among the various components.
- **AI/ML Components**: Enhance the decision-making process through machine learning and artificial intelligence techniques.

[1] https://www.inspire-5gplus.eu/

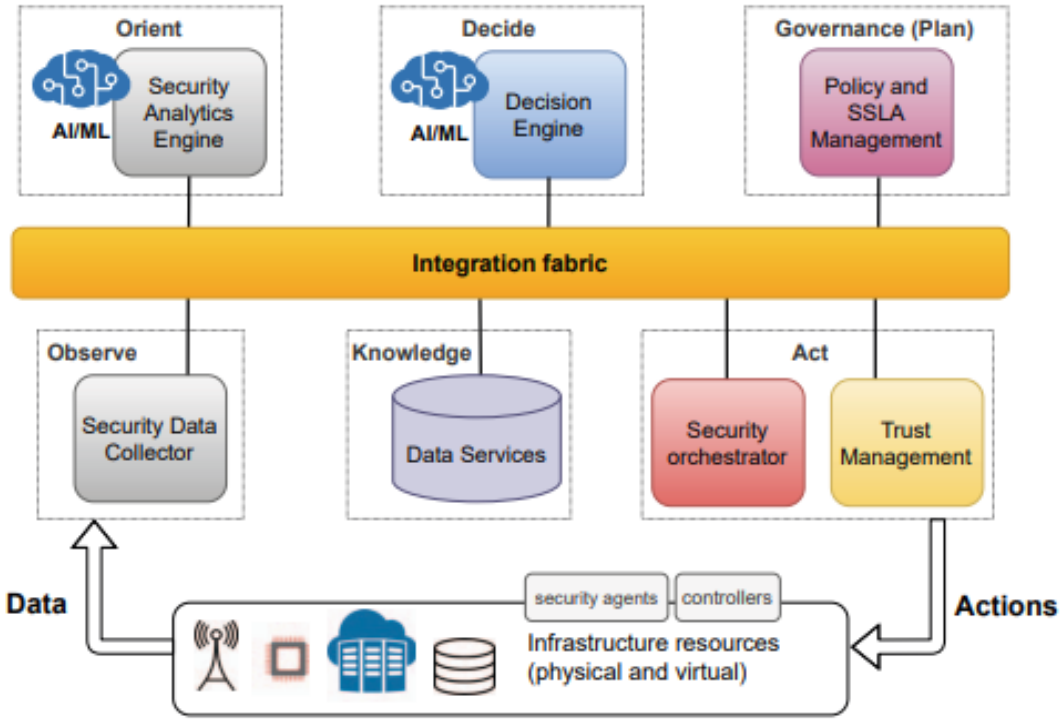

Figure 2.7 – ZSM-driven security framework [Chollon et al., 2022a]

The framework's implementation within the INSPIRE-5Gplus project demonstrates its practical applicability and effectiveness in real-world scenarios, and this study's key contributions can be summarised as follows:

- **Integration of Advanced Technologies:** The framework integrates advanced technologies such as AI, Network Function Virtualisation (NFV), Software Defined Networking (SDN), and network slicing. This integration enables intelligent, autonomous, and agile management of cellular network architectures, ensuring efficient and effective security management.
- **Closed-Loop Automation:** This significant aspect of the framework involves continuous monitoring, analysis, decision-making, and execution of security measures. The closed-loop systems follow models like the Observe-Orient-Decide-Act (OODA) and Monitor-Analyze-Plan-Execute (MAPE-K), ensuring real-time response and

adaptation to security threats.

- **End-to-End (E2E) Management:** Supports E2E management, providing a global perspective of the underlying management domains for coordinated decision-making and workflow across various network segments, enhancing the overall security posture.

- **Proactive and Reactive Security:** The framework includes mechanisms for both proactive and reactive security management. Proactive measures involve continuous evaluation of the attack surface and potential threats, while reactive measures focus on responding to detected security incidents in real-time.

- **Conflict and Policy Management:** Effective conflict and policy management are integral to the framework. Using hierarchical policy enforcement and conflict detection modules, the framework ensures that security policies are consistently applied and that any conflicts are resolved promptly.

Another relevant study in this context is the article by Jayasinghe et al. [2022], which proposes a federated learning (FL) based model for anomaly detection within the ZSM framework, addressing both privacy and communication efficiency concerns.
The use of FL allows decentralised processing, which improves privacy and communication efficiency. The models are trained on decentralised devices and aggregated, making it suitable for the ZSM architecture.
Their proposed model employs a two-stage anomaly detection mechanism integrated into the ZSM architecture. Each stage consists of FL-based detectors that filter network traffic for anomalies, improving detection accuracy.
The model incorporates a mechanism to update anomaly detectors within the closed-loop cycle, ensuring the system adapts to new and unknown security threats.

Keeping the focus on improving the security of the ZSM framework, the article by Xevgenis et al. [2023a] explores the integration of blockchain into the ZSM framework to do so. The key aspects to take from this study are the following:

- **Blockchain-Enabled Security:** The architecture integrates blockchain technology in the cross-domain integration fabric component, increasing security through decentralisation and immutability.

- **Private, Permissioned Blockchain Network:** Ensures controlled participation, minimising the possibility of malicious activities by controlling network access and participation actions.

- **Smart Contracts (SCs):** Utilised for secure, automated, and immutable execution of management functions within the ZSM framework.

- **Oracle Mechanisms:** Securely connect blockchain with other ZSM services, ensuring data integrity and validity across the network.

The integration of blockchain provides robust solutions to several security issues within ZSM, including trust relationships between management domains, security of management functions, and protection of AI/ML models against tampering and attacks. This approach presented showcases the potential of combining blockchain with ZSM to create a highly secure and efficient network management framework of next-generation networks.

2.7 Closed-Loop Automation

The concept of Closed-Loop Automation (CLA) is closely related to ZSM. As discussed in the documents (REF to CLA ETSI documents), CLA involves automated processes that manage the network by continuously monitoring data and adapting actions based on this feedback, significantly reducing human involvement as intended with the ZSM paradigm. All Closed-Loops have a few components/objectives in common. Namely, monitoring, analysis, decision, execution and knowledge components.

Monitoring components ingest and preprocess data collected from managed entities or external sources to the closed-loop. Analysis components derive insights from the monitored data and historical information. Decision components use the insights derived to generate workflows and action plans. Execution components implement the deided actions on managed/monitored entities. Lastly, knowledge components serve as a repository for shared data between various closed-loop stages/phases and also between cooperating closed-loops.
As discussed in the first ETSI CLA document mentioned, some enablers are required for the correct operation of these ZSM-based closed-loops. Within these enablers, we find Closed-Loop Governance, which refers to a set of capabilities that allows external entities to manage the lifecycle of closed-loops, encompassing starting and stopping loops, managing their configuration, and updating their behaviour based on operational feedback. Governance also involves the retrieval of performance and health status of the loops, which could be done via key performance indicators (KPIs) appropriate to the CL in question. The lifecycle management of the CLs details the different phases that a closed-loop goes through within the ZSM framework. Nomenclature may change according to the application of the closed-loop in question; however, to preserve consistency we will refer to the phases described in the official ETSI documents and described in the following Figure 2.8:

- **Preparation Phase**: Designing the CL based on the intended use case and specified goals.

- **Commisioning Phase**: Instantiating and configuring the CL before it goes live.

- **Operation Phase**: Running the CL, monitoring its performance, and making necessary adjustments to optimise functionality.

- **Decommisioning Phase**: Properly shutting down a CL when it's no longer needed, ensuring that all resources are cleaned up and documented.

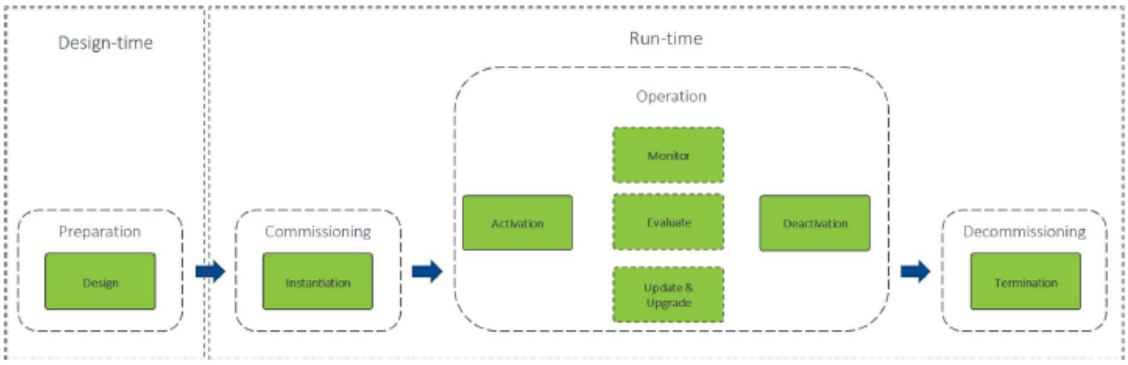

Figure 2.8 – Closed-Loop lifecycle phases and activities CITE!! []

Another enabler described in the first ETSI CLA document is Closed-Loop Coordination. This refers to a set of capabilities that allows multiple CLs running within the ZSM framework to be coordinated, with the main objective of improving their performance and fulfilling their goal(s).

CL coordination can be classified into two types:

- **Inter-Loop Coordination:** Manages how different CLs interact with teach other, ensuring that actions taken by one loop do not conflict with or duplicate the efforts of another. This is crucial in complex environments where multiple loops operate concurrently.

- **Hierarchical and Peer Coordination:** Involves coordinating actions between hierarchical loops (loops with a parent-child relationship such as E2E CL and a domain specific CL) and peer loops (loops at the same level), which is essential for maintaining system harmony and operational efficiency.

Within the second ETSI document on CLA [ETSI, 2022], some use case solutions are provided for a better understanding of how a CL can be applied, in terms of use

cases. Five different use cases scenarios and their solutions are described and these can be summarised as follows:

1. **Management of Physical Resources**
    - **Inclusion of New Physical Resources**
        - **Scenario:** A new physical resource, such as a radio unit, is added to a management domain (e.g., a Radio Access Network domain), impacting the network's service capabilities.
        - **Closed-Loop Solution:**
            * The management domain automatically updates its inventory and capabilities upon the addition of the new resource.
            * Other management domains receive notifications about these changes, allowing them to adjust their operations accordingly.
2. **Service Healing and Resilience**
    - **Automated Service Healing**
        - **Scenario:** A failure occurs in one of the management domains hosting a part of the E2E service, necessitating automatic recovery to maintain service continuity.
        - **Closed-Loop Solution:**
            * The local domain first attempts to self-heal the service.
            * If local recovery is insufficient, the issue is escalated to the E2E management domain, which coordinates across multiple domains to ensure the service is restored.
3. **Analytics-Driven Adaptation**
    - **Dynamic Configurability of E2E Service Monitoring**
        - **Scenario:** An E2E service deployed across various management domains requires dynamic adjustments in monitoring based on evolving service conditions and analytics insights.

- **Closed-Loop Solution:**
  - The E2E management domain can request changes in monitoring parameters from individual management domains based on real-time analytics.
  - This dynamic reconfiguration allows the system to adapt to changing network conditions and maintain optimal performance and service quality.

4. **Cross-Domain Coordination**

- **Coordination Between Multi-Domain Closed Loops**
  - **Scenario:** Multiple management domains must coordinate their operations to handle changes or events that affect more than one domain.
  - **Closed-Loop Solution:**
    - Information about operational changes in one domain is shared with other relevant domains to maintain a coherent and unified response across the network.
    - This coordination is crucial during large-scale events or disruptions, ensuring that all network segments operate harmoniously.

5. **Governance and Configuration**

- **Enabling Pause Points in Closed Loops**
  - **Scenario:** There is a need to temporarily halt the automated processes at certain points in the closed loop to allow for manual checks or adjustments.
  - **Closed-Loop Solution:**
    - Pause points can be configured within a closed loop, allowing operators to freeze the loop's operation at critical stages.
    - This feature is essential for scenarios where human oversight is required before proceeding with automated decisions, enhancing control and safety.

More advanced capabilities of CLA are discussed in the third document [ETSI, 2023]. These capabilities include cognitive features of CLs, which are enhanced by ML to become more adaptive and self-learning, and intent-based operations, wherein the CL's management systems can translate high-level business intents into actionable configurations and operations automatically. Beyond this, and unlike the first two documents which are mainly about establishing a foundation and providing solutions, this document pushes towards future standardisation efforts. It does this by laying out a pathway for incorporating advanced ML techniques and cognitive capabilities into standard network operations, which can be pivotal for organisations looking to adopt these technologies.

Implementations of CLA can be found throughout the available literature; however, we will keep our focus on network and security related implementations, such as the one described in Swetha et al.. In this study we are presented practical implementations of CLA into network deployment, operation, and maintenance. This is achieved through the integration of CLA with existing Network Management Systems (NMSs), such as Zero Touch Provisioning (ZTP), YANG data modelling, and configuration management protocols such as NetConf as can be seen in Figure. 2.9. This exemplifies how its possible to achieve the Ready-Made Closed Loops (RM-CL), through these YANG templates, as described in the conceptual ETSI document, wherein are also discussed Made-to-Order Closed-Loops (M2O-CL) which are tailored to the specific service needs.

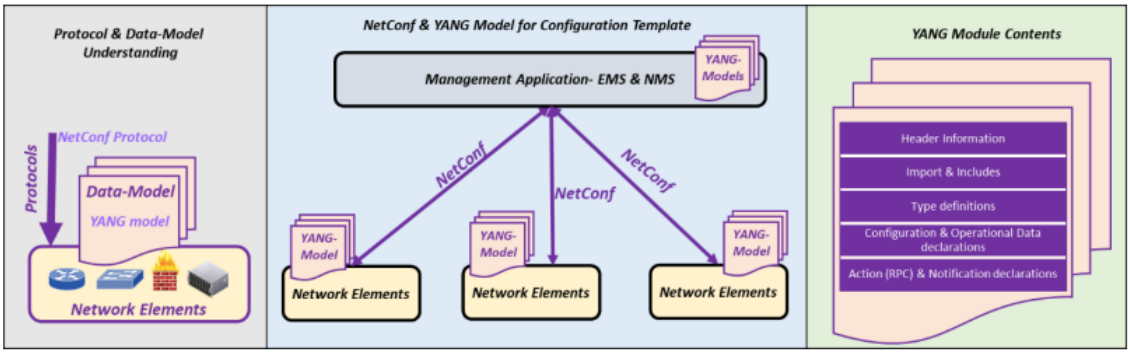

Figure 2.9 – Integration of YANG models for network configuration templates [Swetha et al.]

In terms of network security, we can find studies like the one by Henriques et al. [2022], wherein through the use of Decision Trees (DTs) derived from machine learning models, the proposed framework automatically identifies and classifies anomalies present in network data. The proposed model also demonstrates how capabilities such as high-level policy translation may be applied, since once the anomalies are detected, the identified issues are translated into security policies using a Policy as Code (PoC) approach. This means that security policies are not only documented but also scripted and encoded so that they can be automatically implemented. These policies are designed to be both human-readable and machine-executable to meet clear documentation and efficient enforcement requirements.
Futher extensing the discussion on the network security applications of closed-loop automation, a notable advancement is illustrated in the study by Benlloch-Caballero et al. [2023], which demonstrates a duañ-layer autonomous system specifically designed to enhance the self-protection capabilities of 5G/6G IoT networks against DDoS attacks. This study complements the cognitive and intent-based operations discussed earlier by showcasing a practical deployment where ML-enhanced closed loops operate autonomously across different network layers and stakeholders without the need for inter-loop communication.

The dual-layer architecture proposed in this study addresses the increasing complexity and scale of managing IoT device security by distributing the detection and mitigation processes across two layers - Digital Service Providers (DSPs) and Infrastructure Service Providers (ISPs). Each layer independently deploys closed loops that collaboratively improve the overall resilience of the network. This setup not only automates the response to DDoS threats but also optimises the performance through collaborative yet autonomous ations of multiple stakeholders. Remarkably, the empirical results from this study underscore a significant enhancement in threat mitigation effectiveness - a 78.12% effectiveness rate compared to a mere 4.73% in standalone systems. Additionally, the response time was improved by 316%, with the system responding in just 18 seconds compared to 57 seconds in conventional setups [Benlloch-Caballero et al., 2023].

This novel approach highlights the role of advanced ML techniques and cognitive capabilities in autonomously managing network security at scale, thus aligning with the future standardisation efforts discussed in ETSI's CLA documents. The ability of this system to operate without direct communication between closed loops yet achieving a coordinated defence posture illutstrates an advanced implementation of intent-based operations, where high-level security intents are translated into effective multi-layered defensive actions automatically.

In addition to the ML-based anomaly detection and network security enhancements discussed earlier, the CLARA framework, as proposed by Sousa and Rothenberg [2021], represents a significant leap in operationalising the cognitive and intent-based features of CLA within a multi-domain network management context. This framework utilises policy-based closed control loops (CCLs) to automate the management and orchestration of network services across multiple domains, from Cloud and Transport to Core and RAN. It also exemplifies the translation of high-level business intents into actionable configurations, facilitating a robust mechanism for network automation that reduces complexity and human intervention in the process. The ability of CLARA to manage services across different administrative domains using autonomous CCLs showcases a practical implementation of intent-based operations within a complex network infrastructure. This feature is particularly relevant in the management of 5G and beyond technologies where network dynamics are increasingly complex.

By automating fault management and leveraging AI-based methods from initial user requirements to the execution of control loops across various network domains (as illustrated in Figure 2.10) CLARA provides a comprehensive model for future zero-touch network environments. This model is particularly useful when it comes to adopting sophisticated network operations that are fully automated, highly resilient, and adaptive to changing network conditions across multiple domains.

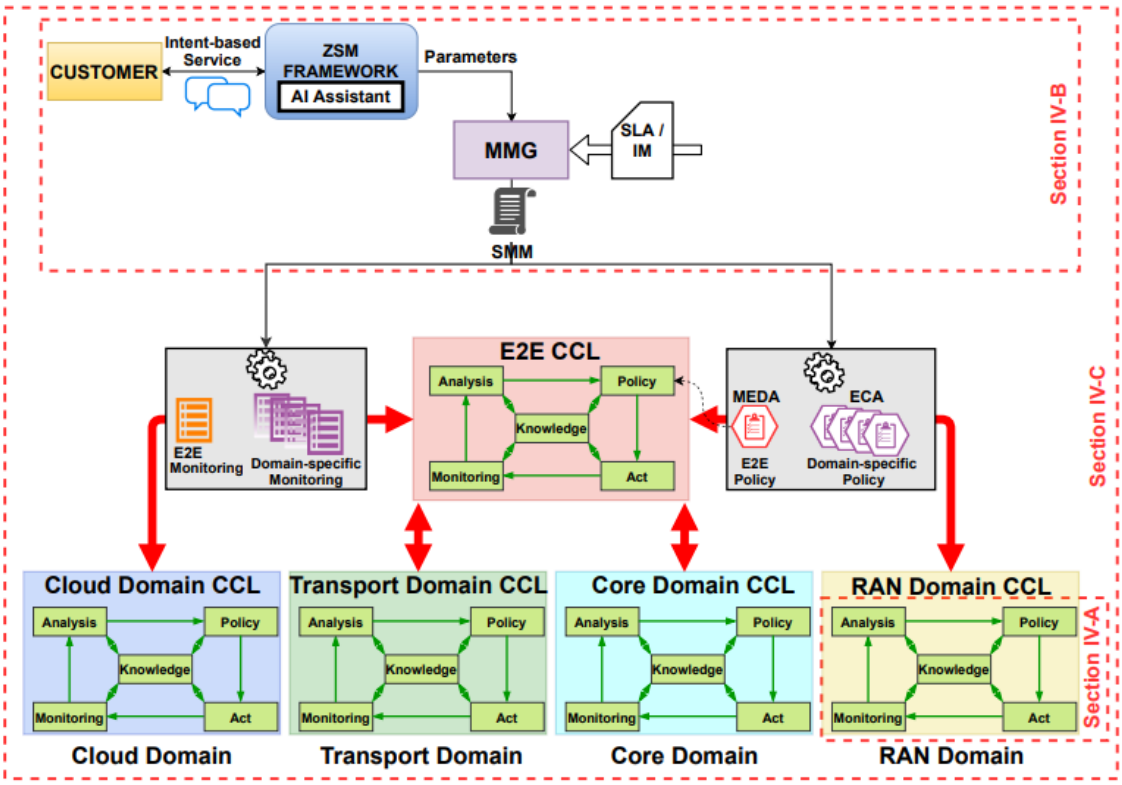

Figure 2.10 – Workflow from user requirement definition to multi-domain management [Sousa and Rothenberg, 2021]

2.8 Intent/Policy-based Networking

2.8.1 Intent-Based Networking (IBN)

In this era of automated networks, it is inevitable to mention concepts such as Intent-based Networking (IBN) and Policy-based Networking (PBN), which even though are not recent concepts, in the case of PBN, have become increasingly prominent in many studies on the matter of network management and security.

IBN simplifies network management by allowing operators to define what the network should do, rather than how it should do it. This shifts focus from manual configuration to automated outcomes [Ouyang et al., 2021]. Its design is user-centric,

since it is made to be accessible even to those without deep technical knowledge. Users state their needs in natural language or though simple interfaces, and the system interprets and executes these intents.

How IBN Works:

- **Automated Translation and Adaptation**: IBN systems use advanced algorithms, typically powered by machine learning, to translate high-level business intents into real-time network configurations. This allows the network to dynamically adapt to changes in network traffic, security threats, and other operational conditions without human intervention.
- **Continuous Learning**: Through continuous monitoring and adaptation, IBN systems learn and optimise their performance and response strategies, ensuring they align with the evolving business and technical environments[Wei et al., 2020a].

Components of IBN

- **Intent Refinement**: This process involves translating a user's declarative commands into a format that the network can understand and act upon. It typically involves natural language processing techniques to interpret and convert user inputs into actionable data.
- **Intent Translation and Policy Mapping**: After refinement, the intents are translated into specific network policies and configurations. This involves ensuring that the intent is feasible given the current network state and resources, checking for conflicts with existing policies, and ultimately deploying these configurations across the network.

- **Feedback and Adjustment**: Post-deployment, the system continually monitors the execution of intents and the network's performance, making adjustments as needed to maintain or optimise outcomes[Zeydan and Turk, 2020].

2.8.2 Policy-Based Networking (PBN)

Policy-based networking involves setting up high-level policies that govern the behavior of the network. These policies dictate how the network should handle various types of traffic and operations based on predefined rules and business objectives. PBN frameworks are designed to dynamically respond to the network's state and external inputs, adjusting traffic routes, prioritising certain types of data, and enforcing security measures automatically[Clemm et al., 2022]. By integrating PBN, networks are granted a more efficient and flexible scaling by automatically adjusting to new devices, traffic patterns, or security requirements based on predefined policies. Furthermore, by automating network responses based on policies, networks can quickly adapt to security threats or compliance requirements, reducing the risk of human error and enhancing overall network security.

Policy Management and Enforcement

- **Centralised Management**: Policies are managed centrally, which simplifies the administration of complex network operations and ensures consistency across the entire network.

- **Automated Enforcement**: Policy engines automatically enforce these policies, making real-time decisions based on network conditions and policy stipulations. This reduces the need for manual interventions and allows for quicker responses to network events[Low, 2020].

Implementations

IBN and PBN complement each other, with IBN focusing on understanding and executing user intents, and PBN providing the mechanisms to enforce these intents through detailed, rule-based controls. The integration of both approaches allows for a network that is both intelligent in understanding user needs and robust in enforcing the necessary rules to meet these needs effectively.
The following is a discussion on studies from both IBN and PBN implementations, some combining them and others using them singularly to improve network management and security performance.

In the complex landscapes of Information and Communication Technology (ICT) supply chains, managing, orchestrating, and enforcing policies becomes particularly challenging due to the heterogeneous nature of the networks involved. The article by Bensalem et al. [2022] introduces an innovative approach to leveraging ML solutions within IBN frameworks, specifically focusing on optimising the computational performance of these systems across varied hardware platforms.
The implementation of ML in IBN, as discussed in the article, involves benchmarking ML techniques to predict their performance across different hardware setups. This is critical to ensuring that the intents are executed efficiently, taking into account the computational constraints and capabilities of the underlying hardware.
A specific technique that was used named Collaborative Filtering, helps in predicting the computational performance of various ML techniques based on sparse data from previous benchmarks. This approach allows the system to efficiently select the best-suited ML model for a given intent and hardware combination, optimising both performance and resource utilisation.

A practical case study was highlighted in this article, which involved the application of these ML benchmarking techniques in an ICT supply chain network, as illustrated in Figure 2.11 wherein we can view the workflow of this process within the reference architecture of the case study in question. This allowed to demonstrate how intents can be dynamically managed and executed with high efficiency, enhancing the overall system's responsiveness and capability. However, as shown in Figure 2.12, if more than 50% of benchmark data is missing, the results show that improvements can still be made to this approach.

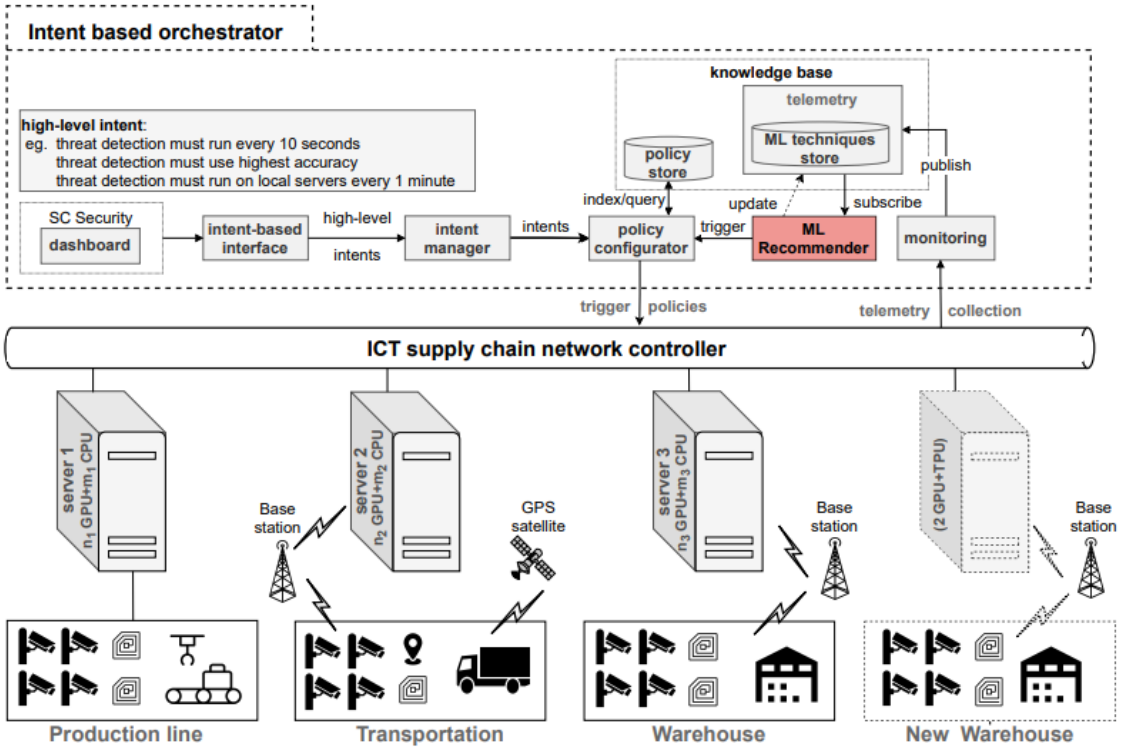

Figure 2.11 – Reference architecture of an intelligent intent based management system within an ICT supply chain [Bensalem et al., 2022]

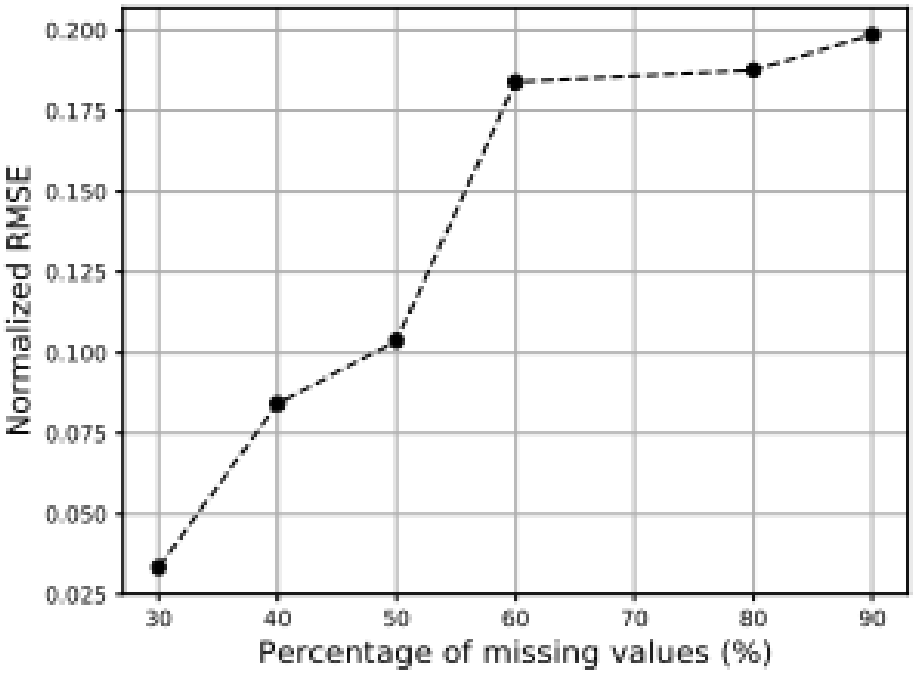

Figure 2.12 – Summary of benchmark prediction results using Normalised RMSE as presented in [Bensalem et al., 2022]

Continuing with the theme of IBN applied to complex network environments, the study by Davoli et al. [2019] proposes a reference architecture that uses an intent-based northbound interface (NBI) for end-to-end service management across multiple technological domains. The architecture proposed emphasises the critical role of a vendor-agnostic, open, and interoperable NBI that abstracts domain-specific details, allowing for unified management and orchestration of services spanning different network technologies.

The intent-based NBI is designed to manage composite services derived from chaining various network functions, often deployed across heterogeneous environments such as IoT infrastructures, cloud services, and transport networks. This aligns with the principles of IBN where high-level operational goals and service outcomes are defined rather than the intricate details of the networking mechanisms.

The practical application of this architecture was demonstrated through a complex service provisioning scenario that includes an IoT deployment, a cloud-based data processing platform, and a transport network. This example illustrates the dynamic differentiation of QoS for different data streams, which are managed effectively through the integration of computing and network resource management.
Also within the proposed scenario, data produced within the IoT domain are processed and consumed in a cloud domain, where different data streams traverse various service function chains. The orchestration is handled through an intent-based approach which dynamically adjusts the network configuration to meet the varying QoS requirements of these streams.
One of the most notable challenges addressed is the need for unified service management across multiple administrative and technological domains. The proposed solution leverages an intent-based interface to declare service chains and network functions without requiring detailed knowledge of the underlying technologies, which simplifies the management of complex multi-domain environments.
The efficacy of this architecture is validated within a heterogeneous testbed encompassing OpenFlow/IoT SDN domains. The tesbed scenario demonstrates the architecture's capability to handle real-world service provisioning tasks, confirming the practical viability of applying IBN principles across diverse technological landscapes. Among the metrics used to evaluate the architecture's performance are:

- **Responsiveness:** Measured by the time taken to detect changes in intent and reconfigure the network accordingly.
- **Resource Efficiency:** Assessed by monitoring resource utilisation rates before and after intent changes.
- **Reliability and Fault tolerance:** Evaluated during the fault tolerance tests by the system's capacity to successfully reroute traffic and maintaining service availability during simulated failures.

In the case of legacy network environments, issues are raised in the sense that network policy management in these environments is not inherently designed for

intent-based systems as raised by the work of Mercian et al. [2021]. In this work the Policy Intent Inference (PII) system is proposed, with the objective of bridging the semantic gap in these environments. And it does so by extracting policies from network devices, abstracting them into a structured data model, and then using clustering and information retrieval techniques to infer high-level policy intents. This process not only simplifies policy management but significantly aids in troubleshooting and conflict resolution, reducing the overall time and effort required in policy compilation and enforcement.

The devised system leverages semantic metadata - such as labels and descriptions embedded within device configuration - that traditionally remain underutilised in policy management. By analysing this metadata, the PII system enhances the inference of intents that remain consistent even as network entities evolve.

This study describes the application of advanced Natural Language Processing (NLP) and clustering techniques to analyse and infer policy intents from the disparate and fragmented configurations typical in legacy systems. This bottom-up approach allows for dynamic and automatic reconstruction of policy intents across a heterogeneous network landscape.

The PII system's ability to abstract and infer intents across multiple devices (using a data model schema for the extracted data as shown in Figure 2.13) and domains addresses a significant challenge in legacy networks characterised by manual configurations and diverse hardware. The results showcased in this work highlight the system's efficacy in automating the extraction and inference of policy intents, thereby facilitating a more streamlined and error-resistant approach to network management.

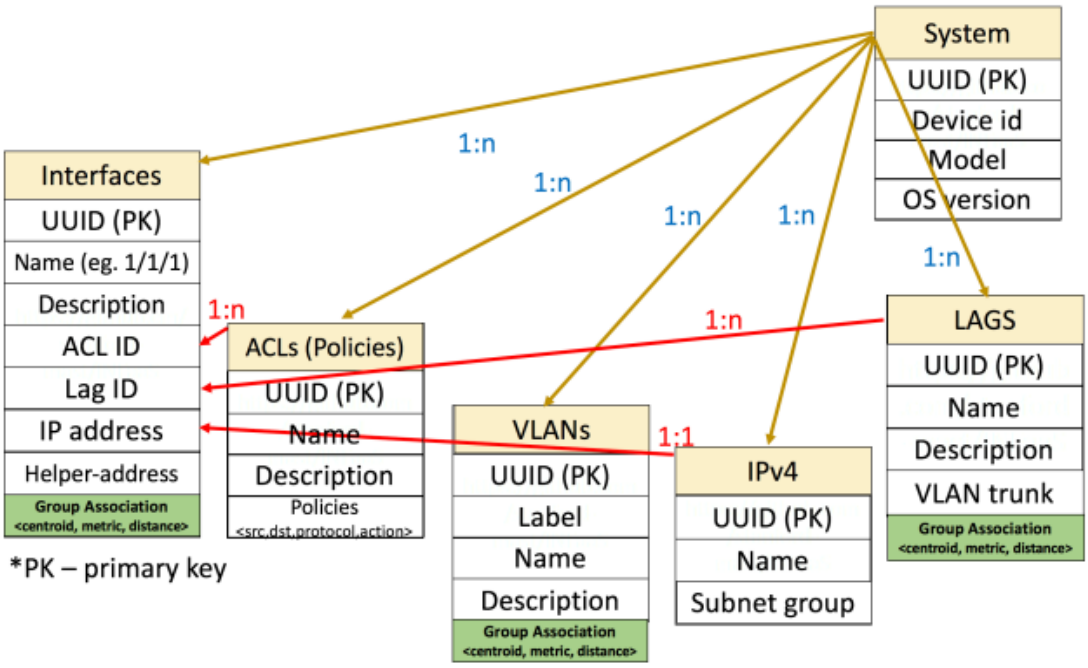

Figure 2.13 – Data Model Schema extracted per device in a Network [Mercian et al., 2021]

The implementation of the PII system, which is depicted below in Figure 2.14 through a diagram of its architecture/workflow, was tested across a heterogeneous tested, demonstrating its capacity to manage and infer policy intents effectively. The system proved capable of handling complex policy scenarios and dynamically adapting to changes in network configurations, showcasing a substancial reduction of time required in policy conflict resolution, when policies were inferred (through the PII system) as opposed to non-inferred policies. The inference time itself was measured, which demonstrated that through the interpretation of network metadata, the PII system was able to efficiently infer network policies in real-time, thus allowing rapid policy configuration and adjustments throughout the network's deployment. By comparing these inferred policies to actual policy implementations and intents documented by network operators, the PII system also demonstrated very high accuracy. Beyond this, the system proposed in this work was also able to handle large datasets, being able to process extensive network configurations without significant delays, solidifying its usefulness in network scalability.

Overall, this system showcases a scalable, accurate and efficient way to automate network configuration and re-configuration, by significantly enhancing policy management in networks.

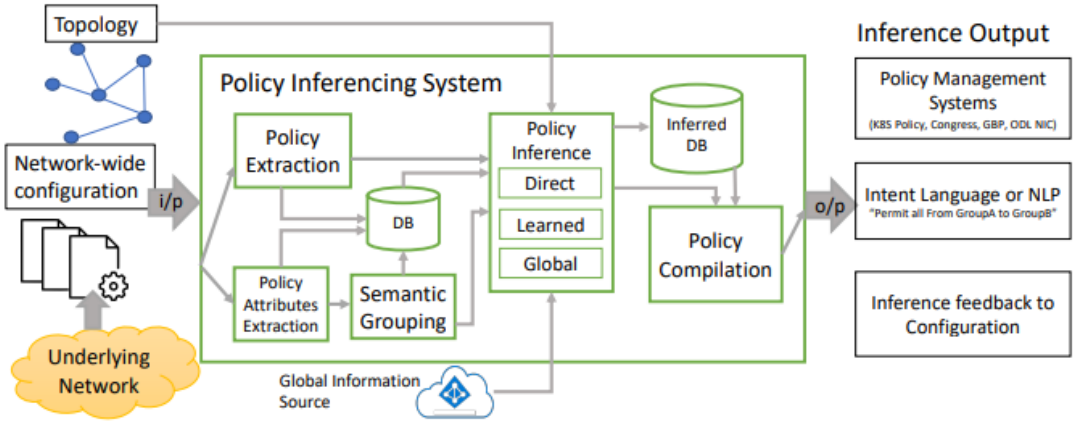

Figure 2.14 – PII System Architecture/Workflow [Mercian et al., 2021]

2.9 Summary

This chapter provides an overview of the foundational technologies relevant to the proposed system, particularly within the context of mobile network advancements. It begins with an exploration of 5G's impact on connectivity, establishing a baseline for next-generation applications in fields like IoT, augmented reality, and AI. Following this, key technological paradigms, including SDN and NFV, are reviewed. SDN introduces a flexible network architecture that separates the control and data planes, enabling centralised network management, while NFV enables virtual deployment of network functions, reducing reliance on hardware and promoting scalability.

The chapter then delves into NS, a 5G innovation that partitions networks into virtual segments tailored for diverse applications. This technology relies on SDN and NFV to allocate resources dynamically, meeting specific performance needs for applications ranging from smart cities to autonomous driving.
The concept of MEC is examined, emphasising its role in bringing computation closer to data sources, which is essential for latency-sensitive applications.

Finally, ZSM and CLA are introduced as frameworks designed to automate network management, minimising human intervention. These frameworks use machine learning and cognitive automation to respond dynamically to network events. Advanced methodologies like IBN and PBN are discussed, focussing on how they allow networks to interpret high-level user intents and enforce policies automatically, further enabling a resilient, adaptive network infrastructure.

3 Proposed Model

3.1 Introduction

The evolution of mobile networks from 4G to 5G and beyond has brought about significant advancements in network capabilities that, as previously discussed, enhanced transmission speeds, reduced latency, and increased connectivity to a broader spectrum of devices.

However, as has been shown time and again, with new developments come new challenges, and the next generation of mobile network technologies is no different in that regard. New and complex security challenges have been introduced due to these advancements, and they require innovative and robust solutions. One such solution is the implementation of closed-loop automation (CLA) systems within network security frameworks. This chapter presents a comprehensive model for such a system, with a particular focus on the Security Decision (SD) component, which will be extensively discussed, as it plays a critical role in ensuring real-time threat mitigation.

The proposed model is part of a larger European R&D project named 6GOPENSEC-SECURITY[1] aimed at developing a robust security system that leverages CLA to enhance network security across various domains, such as EDGE, cloud, and RAN. CLA provides a framework for continuous monitoring, analysis, and adaptive response to security incidents, thereby reducing the need for manual intervention and increasing the system's resilience and efficiency.

Firstly, we will begin by outlining the general architecture of the proposed model, highlighting the integration of key components such as the Security Decision component, Security Data Analytics, Security Data Collection, and Security and Privacy Data Service components. Then we'll delve into the specific functionalities and interactions of the SD component, explaining how it processes data to make informed security decisions. Detailed workflows will be presented to demonstrate how the system responds to security incidents, showcasing the practical implementations of CLA.

By examining both the architectural design and the operational workflows, this chapter aims to provide a thorough understanding of the proposed model's capabilities and its potential impact on network security. The insights gained from these model shall hopefully not only contribute to the academic field, but also offer practical applications for future mobile network technologies, ensuring that they are secure, reliable, and capable of meeting the diverse needs of modern telecommunications.

3.2 Background and Motivation

In the context of modern telecommunications, the transition from 4G to 5G and the upcoming 6G technologies have introduced unprecedented improvements in the

[1]https://www.cttc.cat/project/secure-network-slice-manager-for-open-and-disaggregated-6g-networks/

field. As we have mentioned before, these advancements have opened many possibilities for improvement, and also many challenges. The challenges that have arisen in terms of network security are particularly complex and require a continuous research and development effort, to assure data, and thus user safety.

3.2.1 Evolution of Network Security Challenges

As mobile networks evolve, so do the threats they face. Traditional security mechanisms, which were sufficient for earlier technological generations, are often inadequate to address the sophisticated attacks targeting contemporary networks. The proliferation of devices, increased data traffic, and the critical nature of many new applications (such as autonomous vehicles, remote healthcare, and smart cities) demand a robust and adaptive security framework [Geller and Nair, 2018, 5G Americas, 2019, Shree et al., 2019].

One of the key challenges is the dynamic and heterogeneous nature of modern network environments. With components spread across various domains, including EDGE, cloud, and RAN, ensuring consistent and effective security measures becomes increasingly difficult. Additionally, the rapid pace of technology adoption means that security systems must be highly flexible and capable of evolving in response to new threats and vulnerabilities.

3.2.2 Closed-Loop Automation as a Solution

CLA presents a promising solution to these challenges. By integrating continuous monitoring, real-time data analysis, and automated response mechanisms, closed-loop systems can dynamically adapt to changing security conditions with minimal human intervention. This not only enhances the efficiency and reliability of security

operations but also assures network operators that the network can respond swiftly to potential threats.

This concept of CLA has several key components that are relevant to this project and network security in general:

- **Monitoring:** Continuous observation of network activities to detect anomalies and potential threats.
- **Analysis:** Processing and interpreting collected data to identify security incidents and assess their impact.
- **Decision-Making:** Determining the appropriate response to identified threats based on predefined rules and real-time data.
- **Execution:** Implementing the necessary actions to mitigate threats and restore a regular operational state.
- **Feedback:** As is implied in any closed-loop system, it uses the outcomes of executed actions to refine and improve the system's future responses.

Within this framework, the SD component plays a critical role. It acts as the brain of the closed-loop system, where data from various sources is analysed, and decisions are made regarding the appropriate security measures. The SD component must be capable of handling vast amounts of data, applying advanced algorithms to detect threats, and coordinating with other components to implement effective responses.

The reasoning behind the focus on the SD component stems from its central role in the overall security architecture, since not only does it play its mitigation role, as it has inherited the management role from the Closed-loop Management component, which will be shown and briefly discussed further along this chapter.
An effective SD component can significantly enhance the network's ability to defend against attacks, maintain service integrity, ensure privacy and security of user data, as well as adapt to changing network conditions throughout service deployment, from instantiation or comissioning, until termination or decomissioning.

3.2.3 Impact on Future Network Technologies

The integration of CLA and a robust SD component is not just a theoretical exercise, but has practical implications for the future of network security. As we move towards 6G and beyond, the ability to automatically detect, analyse, and respond to security incidents in real-time will be crucial. This approach aligns with the broader trends in network management, where automation, artificial intelligence, and machine learning (ML) are increasingly being employed to handle the complexities of modern networks[Futuriom, 2020, Browne et al., 2021].
By developing and implementing a closed-loop automation system, this project has the goal of contributing to the creation of more secure, reliable, and resilient network infrastructures. This will not only enhance the current state of network security, but also lay the groundwork for future innovations in telecommunications.

3.3 System Architecture

3.3.1 General Framework

The proposed model for the CLA automation system is built upon the ETSI's ZSM principles, which enable zero-touch automation of network service management. These principles are essential for creating a dynamic, flexible, and secure network environment. The whole framework into which the CLA is integrated, can seen in Figure 3.1, and incorporates three main principles: **multilevel domains**, an**integration fabric**, and **closed-loop mechanisms**. Note that, in the following depiction, ZSM's reference architecture, presented in the previous chapter, is placed in the background of the security framework in a way that showcases the similarities between them.

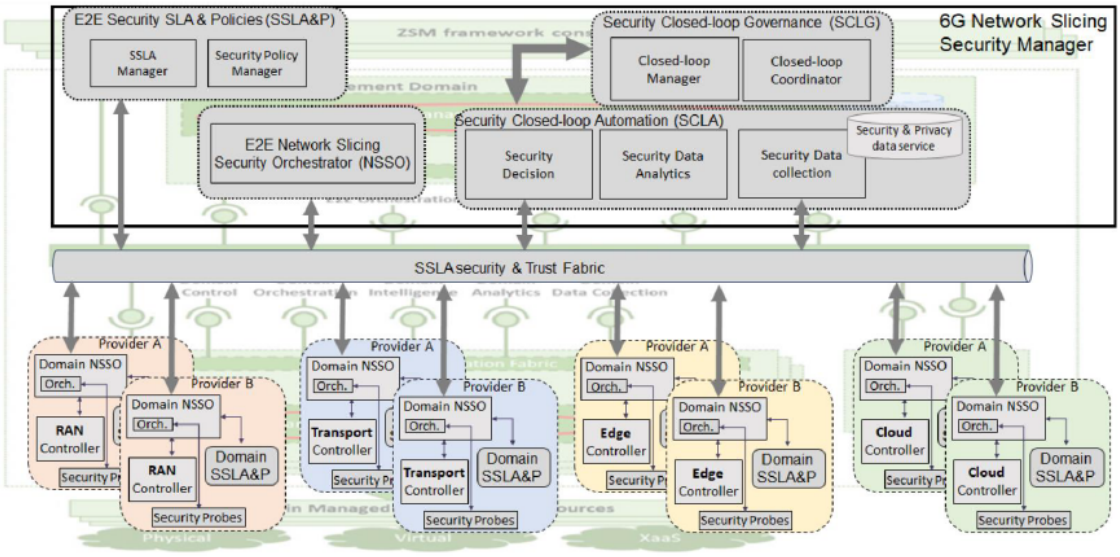

Figure 3.1 – 6G Network Slice Security Manager architecture as presented in the official documents of the 6GOPENSEC-SECURITY project

- **Multilevel Domain:** The multilevel domain approach defines the scope and granularity of the automation processes. This principle allows for different levels of abstraction and decomposition of network and service functions, ranging from end-to-end to domain-specific levels. It ensures that each level can operate independently while still contributing to the overall network security and efficiency

- **Integration Fabric:** The integration fabric connects the different domains and provides a unified view of network and service resources. This layer is crucial for enabling interoperability, data exchange, and orchestration across various network segments. By following the modern principles of service-based architecture (SBA) used in 5G and the ETSI ZSM approach, the integration fabric ensures seamless communication between components.

- **Closed-Loop Mechanisms:** Closed-loop mechanisms are integral to the continuous monitoring, analysis, and optimisation of network and service performance. These mechanisms enable proactive and reactive actions to ensure

service quality and efficiency. The closed-loop processes involve data collection, analysis, decision-making, and execution, creating a robust framework for managing security incidents.

3.3.2 System Components

Since it is best to fully contextualise the entire framework, we shall now discuss all of the parties/components involved in a high-level of abstraction. This system's architecture is composed of several key components that work together to maintain network security. Each component has specific roles and responsibilities within this network slice security management framework.

- **Security Service Level Agreements &Policies (SSLAP):** This component processes the Operator's Security SLAs and translates them into high-level security policies. These policies are expressed in a High-level Security Policy Language (HSPL) or intent policies, which are then delivered for application across the network.

- **Network Slicing Security Orchestrator (NSSO):** The NSSO is the core of the service, responsible for interacting with all elements and managing the state of processes and workflows within the Security Manager. At the E2E, and domain level, it coordinates with other components to identify, select, and manage the network domains involved in the security slice. At the domain level, the NSSO translates high-level policies into specific configurations that the domain network controller enforces.

- **Security Decision:** The SD component is central to the Security Closed-Loop Automation (SCLA) system. It evaluates threats based on data analysis and predefined policies, makes decisions on the appropriate security actions, and updates the service policies dynamically. The SD component interacts with other components, such as the Security Data Analytics (SDA) and Security

Data Collection (SDC), to implement decisions and ensure continuous security monitoring.

- **Security Data Analytics (SDA):** SDA processes and analyses security data to identify potential threats. It generates insights through reports and alerts the SD component when a threat is detected. The SDA component ensures that the network remains protected by continuously analysing data and providing actionable intelligence.

- **Security Data Collection:** SDC is responsible for collecting security-related data from various probes and sources. It forwards the collected data to other components within the SCLA for analysis and action. SDC ensures that the necessary data is available for making informed security decisions.

- **Security & Privacy Data Service:** SPDS manages the storage and protection of security data, ensuring data integrity and privacy. It supports the overall security framework by providing a reliable repository for security information.

- **Security Closed-Loop Governance:** This subsystem is composed by the **Closed-Loop Manager**(CLM) and **Closed-Loop Coordinator**(CLC). The CLM oversees the configuration and activation of closed-loop processes. It ensures that the security conditions and policies are enforced across the network domain and coordinates with other components to manage security incidents. The CLC is responsible for resolving any conflicts between deployed SCLAs, since it must assure that no SCLA is being instantiated for the same service as one already deployed. **Note:** Both the CLM and CLC's functions were inherited by the SD component during the development of this project, due to scope limitations.

3.3.3 Security Closed-Loop Automation (SCLA) system architecture

The CLA system architecture comprises several key components that interact to form a cohesive and responsive security system. These components are depicted in Figure 3.2 and can be described as follows:

- **Monitoring Component (SDC):** The Monitoring Component is responsible for collecting data from various sources within the network. This could include sensors, probes, and other data collection mechanisms that gather information on network activities.
- **Analysis Component (SDA):** The Analysis Component processes the collected data to identify patterns, anomalies and potential security threats. It uses advanced algorithms, including machine learning and statistical analysis, to derive insights and assess the severity of detected incidents.
- **Decision Component (SD):** The Decision Component, or Security Decision (SD) in this case, plays the role of Decision Engine by receiving inputs from the Analysis Component (SDA) and makes informed decisions on the appropriate actions to take. The SD component leverages predefined rules, policies, and real-time data to determine the best response to security threats.
- **Knowledge Component (SPDS):** The Knowledge Component serves as a repository for data insights generated by the CLA system. It stores historical data, incident logs, and performance metrics, providing a knowledge base that can be used to refine and improve the system's effectiveness over time.
- **Governance Component (SD):** Even though the previous Figure 3.1 was an initial draft of the whole Security Management system, wherein there was a decouple subsystem called Security Closed-Loop Governance (SCLG), which was responsible for managing the SCLA subsystem. As afforementioned, some of the key functionalities of the SCLG, namely the ones attributed to the CLM and CLC, were inherited by the SD component of the SCLA. These

functionalities include the management of the closed-loop's processes lifecycle, overseeing the configuration, activation, monitoring, and deactivation of the CLA system, ensuring that it operates efficiently and effectively.

Note: Within this Security Management system, there is a component that plays the role of Execution Component, as is common in Closed-Loop Architecture. However, this component is not within the closed-loop itself (CLA). It is the NSSO component that plays this role, with whom the SCLA system interacts with, after making an informed decision on how to act facing a security threat, and it is the NSSO that has the power to carry out these actions and enforce the policies required by the service in question.

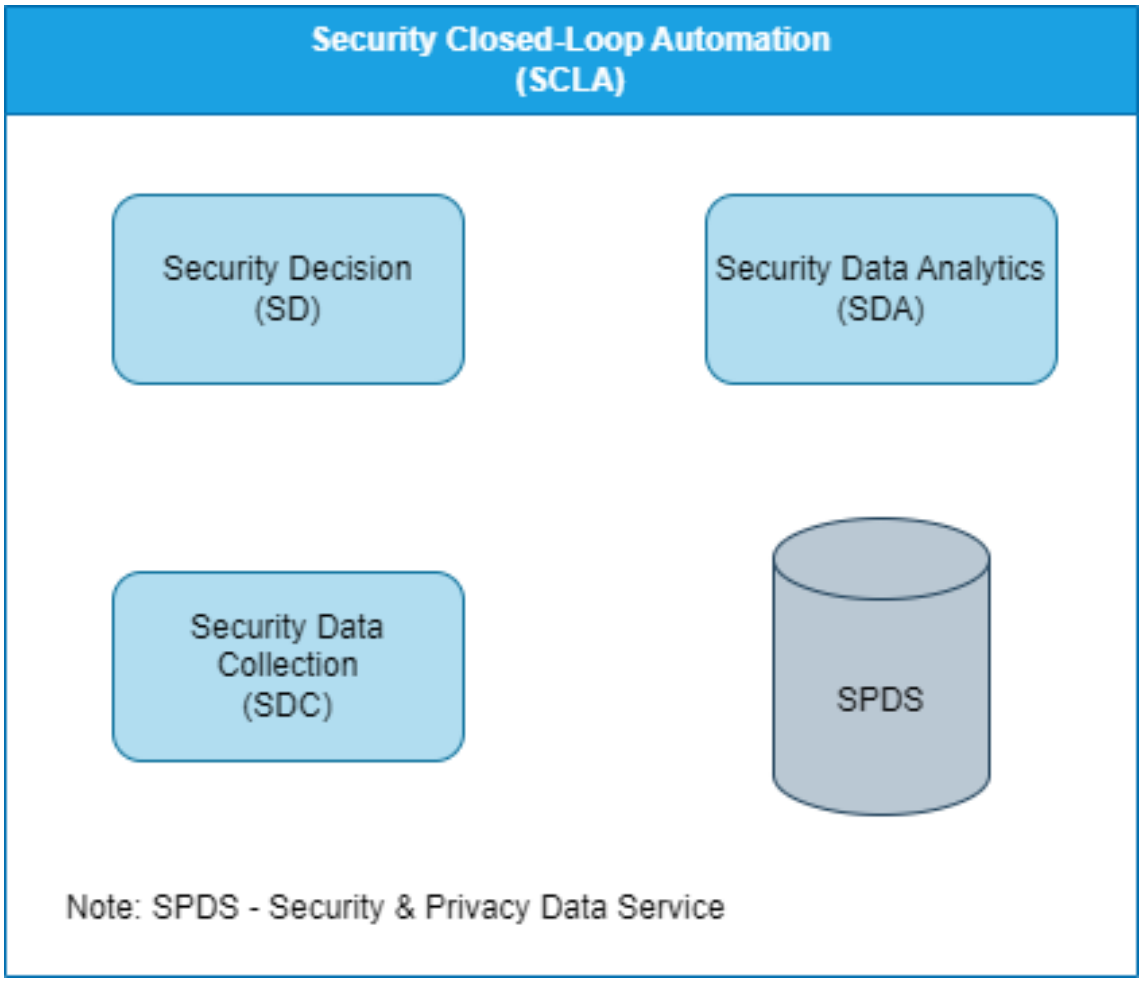

Figure 3.2 – Simplified architecture of the SCLA including its internal components

3.4 Security Decision Component

The Security Decision (SD) component is integral to the SCLA system, providing real-time threat assessment, immediate response determination, and ensuring policy compliance. The other key functionalities of the SD were inherited from the components in the SCLG and include the management of the SCLA's lifecycle, from its deployment, to its termination. This includes the configuration and/or reconfiguration of the SCLA, if necessary, through automation tools. We will now delve into these functionalities, decision-making algorithms, automatic configuration tools, and rule-based systems, as well as key interactions with other SCLA components, and real-world applications of SD.

3.4.1 Security Decision Functionalities

- **Threat Assessment:**
  - The SD component continuously evaluates potential threats based on real-time data and predefined security policies.
  - Utilises a combination of expert system rules to detect anomalies and predict the best course of action to assure service security.
- **Response Determination:**
  - Upon identifying a threat, the SD component determines the most appropriate immediate response.
  - This includes actions such as isolating affected segments, adjusting network configurations, and alerting relevant authorities or operators.
- **Policy Compliance:**
  - Ensuring that all actions and configurations comply with the established security policies is a primary function.

  - The SD component constantly reviews and updates policies to reflect current security needs and regulatory requirements.

- **Closed-Loop Management:**
  - Upon being request a security service, the SD component must be able to interpret the requirements of such service, and deploy the appropriate, tailored SCLA.
  - This commissioning of services must be automated and any adjustments required during the deployment of the security service, must also be handled by the SD component.
  - In the case of the service deployed no longer being required, or the event of an attack that requires the service to be terminated prematurely, the SD component, particularly the one in the E2E domain, which we will discuss further on, must be able to do so swiftly.
  - In the event of a service already deployed, being requested, the SD component must be able to assess and resolve whichever conflicts may arise from this deployment. This is to avoid wasting resources by deploying more than one SCLA for the same service.

3.4.2 Decision-Making Algorithms and Rules

The SD component's decision-making process relies on a sophisticated algorithm that integrates multiple data sources and inference mechanisms:

- **Expert System with Rule-based Inference:** The core of the SD component is an expert system, which mimics human decision-making by applying a

set of predefined rules to the data it receives. These rules are derived from security policies and operational guidelines, which can be updated as necessary to adapt to new threats, policies or service requirements.

- **Forward and Backward Chaining:** The decision engine employs both forward and backward chaining methods. Forward-chaining is data-driven, starting with available real-time data from the network and using rules to infer conclusions. Backward chaining is goal-driven, starting with potential threats that have been detected by the SDA and reported to the SD, and working backwards by analysing available real-time data to confirm or deny their presence based on this data.

- **Integration with Data Analytics:** The SD component works closely to the SDA component. The SDA provides real-time threat reports, based of what its ML algorithms have detected. These threat reports are then used by the SD component for its inference process, particularly in the backward chaining method. This collaboration ensures that the SD component can quickly respond to new threats and evolving attack patterns.

3.4.3 Integration/Interaction with other SCLA components

For SCLA to operate to its full operational capabilities, its components must collaborate seamlessly. The SD component is no exception, and it needs not only to fulfill its role correctly, as well as for the other components to do the same to form a cohesive security framework:

- **Security Data Collection (SDC):** This component is responsible for gathering data from various network points or probes, with which the entire SCLA depends upon. With the SD component in particular, the SDC is responsible for providing it real-time data which it can use during its continuous inference processes, while analysing potential threats.

- **Security and Privacy Data Service (SPDS):** This component stores and manages collected data, ensuring its integrity and availability for posterior analysis and reporting. Beyond collected network data, the SPDS also stores data generated by the other components during their service security-related processes. For the SD component, this means mitigation action data as well as data used upon the inference of such actions. The SPDS provides a safe place for it to be stored and audited later on This component is responsible for analysing collected data to identify potential threats, via complex ML algorithms, and generate threat reports. These threat reports are what bind the SDA and SD together, since the inference process is highly dependent on the content of these threat reports in conjunction with real-time data.

The following table, Table 3.1 contains the internal modules of the SD component, which are required for it to fulfil its security and management functionalities. These modules will be further detailed in the following chapter, when discussing the implementation of this model. They represent a lower level of abstraction which is more common when discussing the implementation itself, however, an understanding of these inner workings of the SD component, will help us understand how the SCLA works in general. This modularity is also one of the principles we have tried to uphold throughout the development of this project. Since there are quite a few components that act as central points, in a way to counter single points of failure, we have devised modules that even though are within the same software component, have different roles and responsibilities, assuring efficient use of resources.

Module	Capabilities	Functionalities	Interfaces
Closed-Loop Deployment (CLD)	Deploying and re-configuring the closed-loop based on security requirements	SCLA automated deployment, Dynamic SCLA re-configuration	**Input:** Security service descriptor provided by the slice orchestrator **Output:** Deployment configurations
Decision Engine (DE)	Determining the appropriate action, based on the assessment and metrics on threat reports.	Action Mapping, Escalation Logic, Policy Enforcing	**Input:** Policies obtained, predefined action sets or playbooks. **Output:** Chosen response or action, escalation alerts (if required).
Communication Interface	Connecting different components within the SCLA to the SD, and the SD to external components.	Broker, Producer/Consumer, APIs, Data formatting	**Input:** SCLA configuration parameters/Threat alerts/ Policy related metrics/Security Policies. **Output:** Feedback or status updates for other system components.
Policy Compliance (PC)	Maintaining the security policies that guide the decision-making process.	Policy Storage & Retrieval, Policy Compliance Revision	**Input:** Requirements for policy implementations, updates to required set of policies. **Output:** Set of compliance revised policies or newly updated policies to implement.
Incident Logging & Reporting (ILR)	Logging decisions and incidents to create an audit trail and generating reports for further analysis and compliance purposes.	Mitigation Logging, Report Generation	**Input:** Incident data. **Output:** Mitigation logs, analytical reports.

Table 3.1 – Security Decision Modules

3.5 SCLA's automation process

3.5.1 Configuration and Activation

Configuring the SCLA involves several key steps to ensure the system is prepared to monitor, detect, and respond to security threats to the service it was requested to protect, effectively.

In Figure 3.3, we can see how the whole process is executed step by step:

- **Policy/Service Requirements Loading - Step 1 (steps 1 through 4 within the SD) :** The process begins with the Slice Orchestrator, whose role is played by the NSSO, sending the data containing the service configurations and policies required for the SCLA to enforce and maintain throughout the service to the SD component. The SD component, uses its communication interface module to handle and interpret this received data and send it to the CLD and PC modules. These modules then process these configuration parameters, and policies to, in the case of the CLD, deploy the SCLA in the requested domain, with tailored specifications, and in the case of the PC, to enable or disable rules in the Knowledge Base (KB) of the inference engine (DE) of the SD.
- **Integration with Data Sources and ML models - Steps 2 through 5:** The SD component is then responsible for configuring and deploying the SDA component of the SCLA with the appropriate ML algorithm, since each domain and service will have a more appropriate algorithm to detect threats in its domain/environment. It is also responsible to configure the SDC, in order for it to establish a connection with the appropriate data sources, such as network traffic monitors and security probes within the domain of the service provider in question. All of this data, from the configuration parameters to the policies to uphold is stored in the SPDS with the purpose of auditability.
- **Component Synchronisation:** All the SCLA components must then be

synchronised to ensure they are correctly integrated and able to communicate seamlessly via their communication channels. This includes all four components, the SD, SDA, SDC and SPDS.

- **Deployment of Security Functions - step 6:** After all the previous steps are completed and the SCLA is ready to fulfil its purpose in the requested service, it is deployed with the appropriate real-time monitoring and response capabilities. To finalise the proccess a confirmation message is sent to the NSSO, informing it of the SCLA's deployment for the requested service.

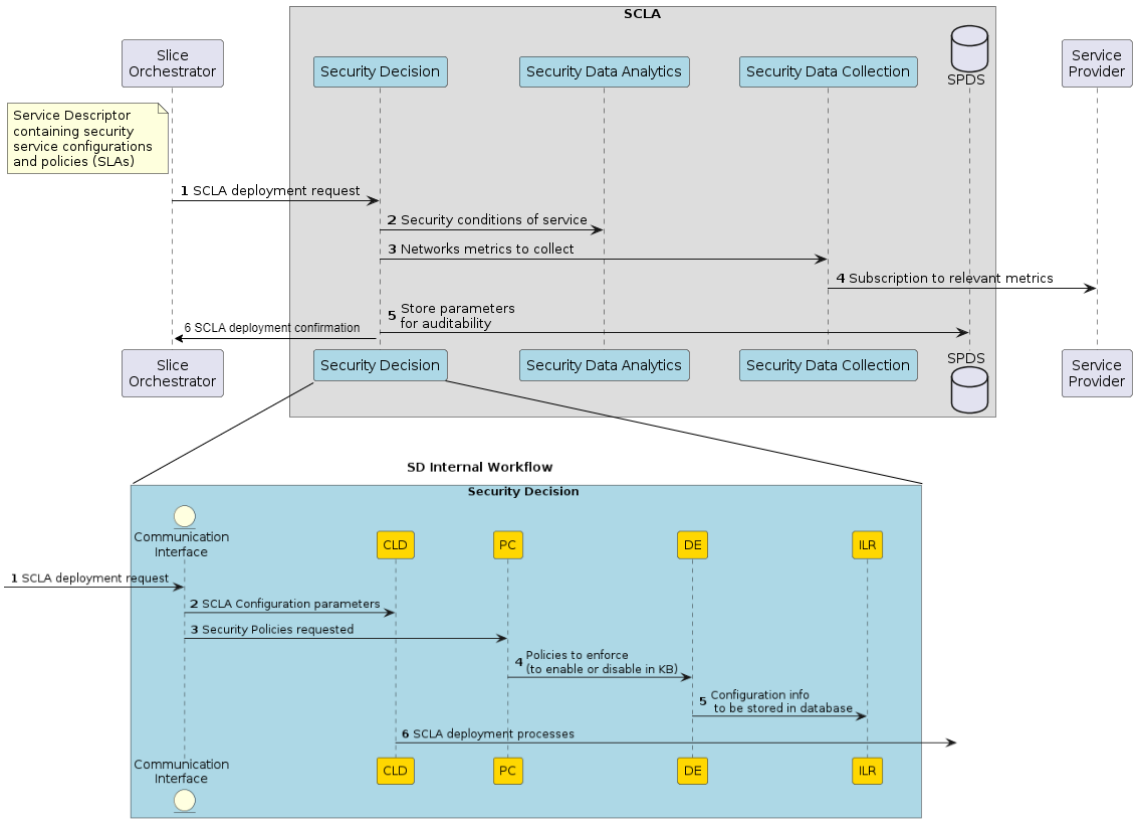

Figure 3.3 – Workflow of the SCLA's configuration process (including SD's internal subprocesses)

3.5.2 Incident Response

The following incident response workflow, displayed in Figure 3.4, showcases how the SCLA system takes action, when responding to any potential threat detected, via a procedural and effective workflow. The following are detailed descriptions of the steps in this procedure, to explain the inner workings of such process.

- **Detection and Analysis - Steps 1 through 4:** The attack on the service is initiated via an external party, which we may call the attacker. The process of continuous monitoring of data, as showcase by the SDC and SDA in steps 2 and 4, is what makes it possible to detect such an attack. By collecting (SDC) and afterwards analysing network data (SDA), the SDA is able to detect any suspicious network activity via its refined ML models.

- **Threat Assessment and Decision Making - Steps 5 through 7:** After identifying a potential threat, the SDA component generates a threat report which it sends to the SD component. The SD component evaluates the severity of the detected threat, using predefined rules, and this threat is then categorised based on its potential impact and likelihood. The threat is then validated, usually through the backward chaining method, by analysing key performance indicators (KPIs), related to the policies required by the security service. After validation, the course of action is determined via the mitigation actions associated with the violated policies in the attack.

- **Response Implementation - Step 8:** After determining the proper course of action to assure service security and policy compliance, the SD component then sends this data to the NSSO, which has the capability of enforcing them in the network. This may involve updates to the SCLA itself, since some attacks may require reconfiguration of the SCLA to better protect the service requested.

- **Post-Incident Review:** After the course of action is determined and enforced, the SCLA must be able to review the effectiveness of its actions. This

is where the data stored in the SPDS on the attack becomes crucial, since it is used to cross-reference current real-time data, after the attack, to validate that the course of action taken was indeed effective. Through this self-evaluation, the SCLA displays self-healing capabilities, since it is able to determine whether or not, it is appropriately configured to secure the requested service.

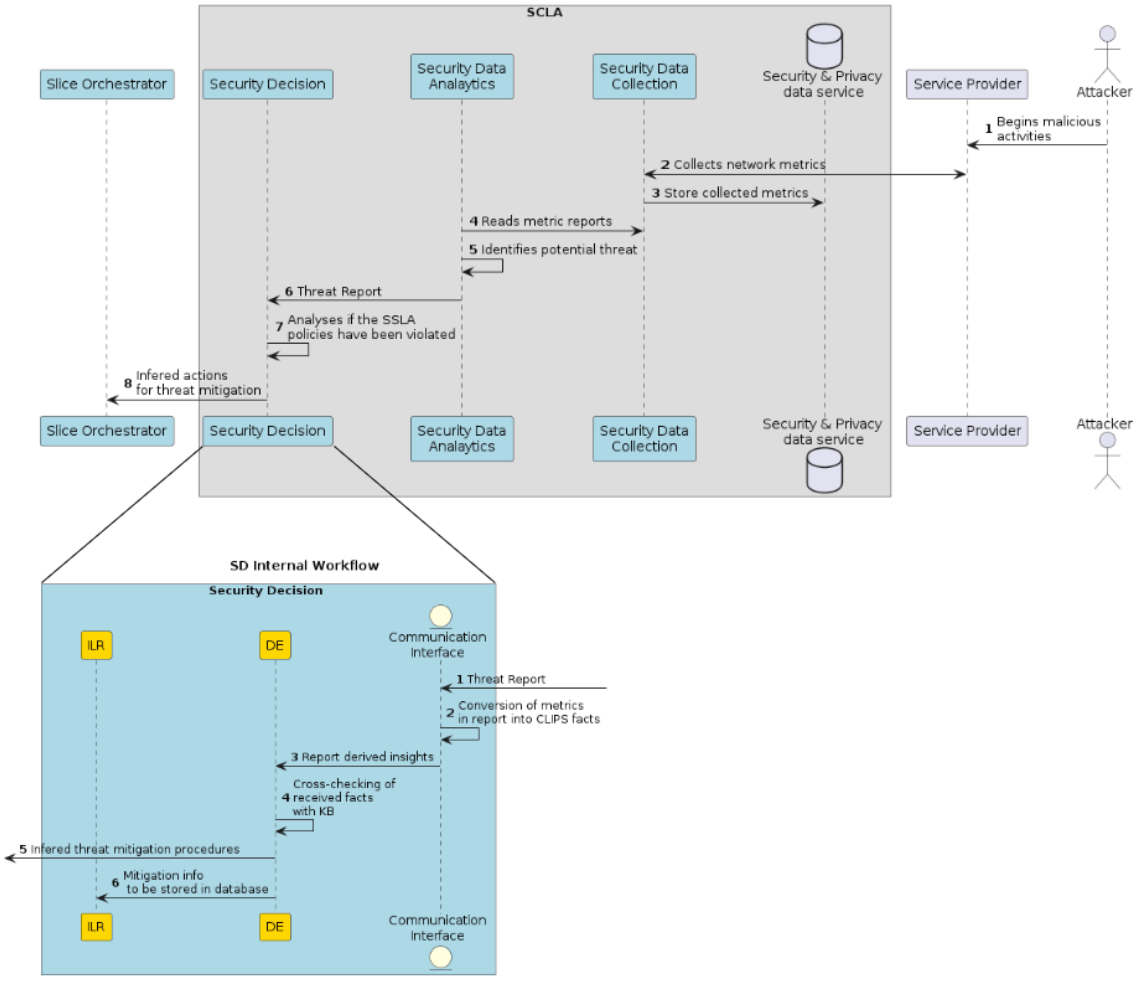

Figure 3.4 – Workflow of the SCLA's incident response process (including SD's internal subprocesses)

3.5.3 Feedback Mechanisms

As with any Closed-Loop implementation, the feedback mechanisms are both crucial for its operation, and also what distinguishes them from other automated systems, by allowing them to self-regulate and minimise the human aspect of these processes. The following is a brief summary of how these mechanisms work and are implemented into the SCLA:

- **Continuous Monitoring and Data Collection:** The SCLA's components continuously collect real-time data from the network, providing ongoing feedback on network conditions and effectiveness of implemented security measures. Beyond this, detailed reports on detected anomalies and incidents are generated, providing insights into the types and frequencies of threats faced by the network.

- **Analysis and Kearning:** Key performance metrics are tracked, such as response times, false positive rates, and mitigation success rates. These metrics are analysed to identify areas of improvement. Based on the feedback and performance metrics, machine learning models and rule-based algorithms within the SCLA (and SD in particular) are refined. This ensures the system adapts to evolving threats and improves its detection and response capabilities.

- **Policy Configuration Updates:** Feedback on the effectiveness of current policies is used to update and refine security policies. The SD component ensures that these updated policies are enforced within the network. Continuous feedback allows for dynamic adjustments to the system configuration. The SD component is responsible for overseeing these adjustments and ensuring that the SCLA system remains responsive and effective.

3.6 Innovative Aspects of the Proposed SCLA

The proposed SCLA system, particularly the Security Decision (SD) component, introduces several novel features and improvements over existing security solutions. This section highlights these aspects and provides context into how they come into play within the SCLA and its components, across various network domains and services.

3.6.1 Advanced Machine Learning Integration

The SD component incorporates cutting-edge machine learning algorithms to enhance threat detection and system responsiveness.

Adaptive Threat Detection

- **Continuous Learning:** Integrates advanced machine learning algorithms that continuously learn from new data, allowing for the adaptive detection of emerging threats.
- **Behavioral Analysis:** Moves beyond traditional signature-based systems by using behavioral analysis to identify anomalies, reducing false positives and enhancing detection capabilities.

3.6.2 Real-Time Policy Enforcement

- **Dynamic Policy Adjustment:** The SD component dynamically adjusts security policies in real-time, adapting to the current threat landscape without manual intervention.
- **Immediate Response Coordination:** Enhances threat mitigation effectiveness by coordinating immediate responses across different network domains.

3.6.3 Closed-Loop Feedback Mechanism

- **Continuous Improvement:** Utilizes a closed-loop feedback mechanism to continuously improve decision-making algorithms based on past incident analysis.
- **Automated Incident Review:** Regularly collects and analyses incident data, facilitating ongoing system enhancements and adaptations.

3.6.4 Comprehensive Policy Compliance

- **Centralized Policy Management:** Manages security policies centrally, ensuring uniform enforcement across all network domains and simplifying updates.
- **Regulatory Compliance:** Maintains compliance with various regulatory standards by automatically adjusting policies to new regulations.

3.6.5 Scalability and Flexibility

- **Modular Architecture:** Features a modular design that allows for scalability across different network sizes and types, enhancing flexibility for technology upgrades.
- **Cross-Domain Integration:** Ensures comprehensive security coverage across multiple network domains such as EDGE, cloud, and RAN through effective integration strategies.

3.7 Use Case Scenarios

In this section, we will explore a few practical use-case scenarios, focussing on the SCLA system and the SD component, particularly in scenarios involving Distributed

Denial of Service (DDoS) attacks and similar threats across different network domains. This type of attack was chosen as the main focus of attack prevention for the project; therefore, the SCLA is particularly optimised to deal with this type of attack.

- **DDoS Attack on Cloud Services**
  - **Scenario**: A multinational corporation's cloud services are targeted by a DDoS attack.
  - **SCLA Implementation**:
    - **Detection**: Gather real-time data and analyze for abnormal patterns.
    - **Threat Assessment**: Assess as high-severity.
    - **Response Determination**: Implement traffic filtering and rate limiting.
    - **Execution**: Execute protective measures.
    - **Feedback**: Adjust strategies based on outcomes.
  - **Expected Outcomes and Benefits**:
    - **Service Continuity**: Minimise downtime.
    - **Resource Optimisation**: Prevent resource exhaustion.
    - **Enhanced Detection**: Improve future threat handling.
- **DDoS Attack on Smart City's IoT**
  - **Scenario**: A smart city's IoT network experiences a coordinated DDoS attack.
  - **SCLA Implementation**:
    - **Detection**: Collect data from IoT devices.
    - **Threat Assessment**: Prioritise critical services.

    - **Response Determination**: Isolate devices and adjust communications.
    - **Execution**: Enforce actions and restore operations.
    - **Feedback**: Update security measures.
  - **Expected Outcomes and Benefits**:
    - **Service Reliability**: Ensure continuous operation.
    - **IoT Security**: Enhance protection protocols.
    - **Resilience**: Strengthen infrastructure.

- **Cross-Domain DDoS on Telecom**
  - **Scenario**: A telecom provider faces a multi-domain DDoS attack.
  - **SCLA Implementation**:
    - **Unified Monitoring**: Collect comprehensive data.
    - **Integrated Analysis**: Identify the attack scope.
    - **Coordinated Response**: Implement cross-domain strategies.
    - **Execution**: Apply protective measures uniformly.
    - **Feedback**: Refine security policies.
  - **Expected Outcomes and Benefits**:
    - **Holistic Security**: Protect all network domains.
    - **Consistent Policy**: Uniform security application.
    - **Enhanced Management**: Better handle complex threats.

3.8 Summary and Transition to Implementation

This chapter details the development of the SCLA system, a model to improve security within the next generation of mobile networks. Building on the challenges introduced by emerging 5G and 6G technologies, the chapter outlines a security framework featuring key components such as SD, SDA, and SDC. It explains how

each component works within the closed-loop system to monitor, analyse, and respond to security threats with minimal human intervention.

In particular, the SD component functions as the system's decision engine, responsible for threat assessment, policy compliance, and coordination across network domains. The integration of machine learning, closed-loop feedback, and real-time policy enforcement sets this model apart, aiming for scalability and resilience across complex network architectures.

The chapter transitions to the next by discussing potential implementation scenarios, notably focussing on mitigating DDoS attacks. This sets the stage for Chapter 4, where the practical implementation of the SCLA system and its internal components, particularly the SD, will be elaborated. This upcoming section will provide concrete steps on deploying the SCLA to achieve the secure, responsive, and adaptable network infrastructure envisioned.

4 Implementation

4.1 System Overview

This chapter aims to demonstrate how the proposed model was developed and deployed. Going through the tools used and required for this process, as well as the reasoning behind choosing each of them. We will provide diagrams similar to the ones shown previously, but with a higher focus on the data flow and processes that occur within the SCLA, particularly within the Security Decision module.

Throughout the development of this project, it was decided that using the SCLA only for domain level (e.g. Cloud, RAN, Transport, etc) security was not enough. Therefore we were tasked with the development of an E2E SCLA system, which focuses on the monitoring of domain level SCLAs that themselves monitor the service at hand. This E2E SCLA is developed toward assuring that the service provider with which the domain SCLA is tasked with monitoring a particular service of, is compliant with higher level policies defined upon the deployment of said service. In other words, the E2E SCLA monitors service provider related data, and the domain SCLA monitors service related data. This E2E SCLA monitors multiple instances of domain SCLAs, and is able to penalise the providers with which these

domain SCLAs are associated with, based on their performance. Instead of possessing multiple modules as in its domain counterpart, it is represented by an instance of a containerised application and an instance of a database prepared for E2E data instead of domain related data. Also unlike its domain counterpart, instead of modules for each component it uses services to represent these components, such as the Security Decision and Security Data Collection service, since no data analytics are required to monitor service provider data, which is mostly performance based and facilitates a more straightforward approach to any mitigation actions required in the event of an issue/attack.

The following Figure 4.1 illustrates the differences between the E2E SCLA and the Domain SCLA:

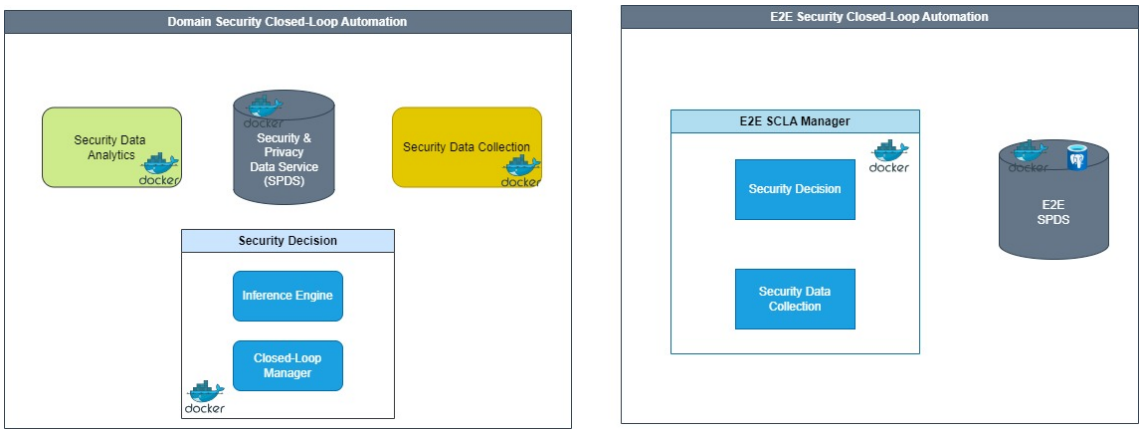

Figure 4.1 – Simplified E2E and Domain SCLA architectures

Beyond what is shown in the architecture diagrams, both the Domain and E2E versions of the SCLA possess an Integration Manager, which is responsible for handling any external communication (outside the closed-loop system), with the exception of not handling network probe communication. Both these integration components are, simply put, a set of APIs that facilitate communication with exterior components, namely the Slice Orchestrator, both at Domain and E2E level.

4.2 Security Decision Module/Component

Implementing the SD component involved finding tools that would both be feasible to use in a containerised environment and that went according to the project requirements, such as the limitation to Python-based tools. One of the key functionalities of this component was the granular deployment of an SCLA that followed the instructions present in the configuration data object received from the Slice Orchestrator as can be seen in Figure 4.2.

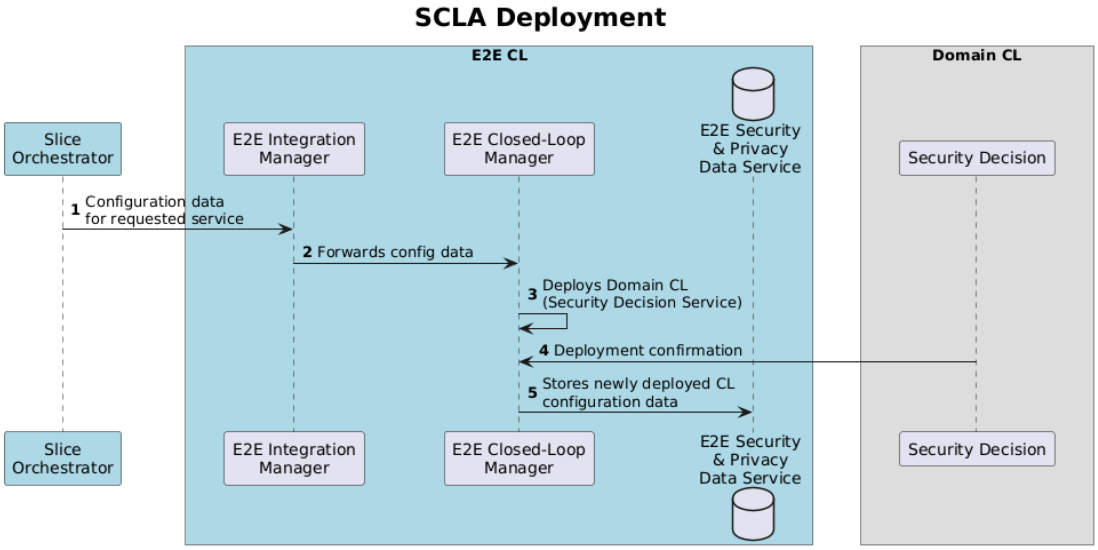

Figure 4.2 – SCLA Deployment

The initial deployment of the Domain SCLA is done by the E2E Closed-Loop Manager, more specifically by the SD Service, as illustrated in the previous figure, using Ansible, which provided the control and granularity required for a customised SCLA deployment based on the configuration data.

Post-deployment, any updates to the configuration must come from the Slice Orchestrator at the Domain level, and these updates must be reported to the E2E Closed-Loop Manager by the SD module for storage in the E2E database.

Operations during runtime of the SCLA may involve changing the source of network data. Security probes, or network probes act as the source of that data for the SCLA and the Domain SCLA has the same role in relation to the E2E SCLA, since it provides it with reports on the service provider's performance.

This allows for the separation of concern in terms of service related metrics and provider related performance metrics, and adds a compensating security control, in the event that the Domain SCLA fails to identify possible issues in the service or provider, the E2E can review the data and identify possible threats.

The Figure 4.3 displays another instance where Ansible is used to reconfigure an already deployed SCLA according to updated configuration data from the Slice Orchestrator.

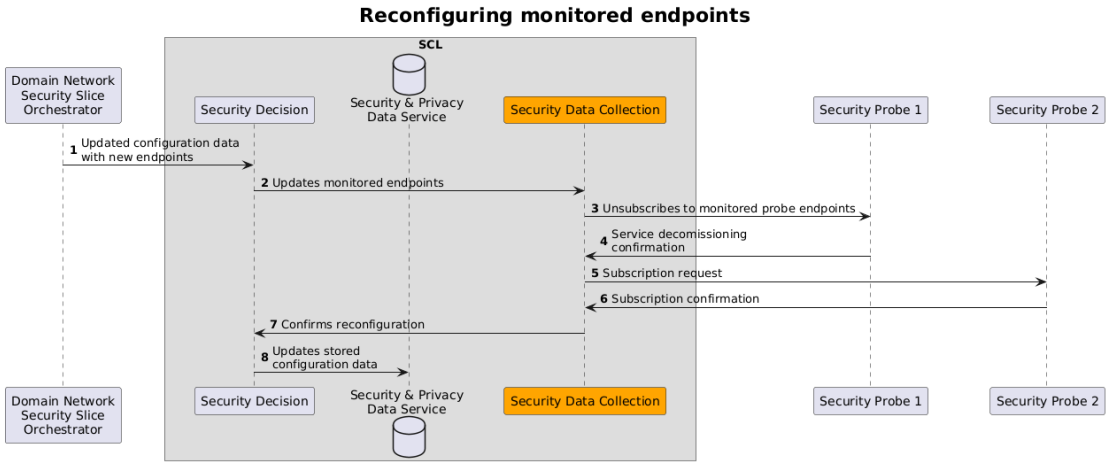

Figure 4.3 – Endpoint Reconfiguration

The decomissioning of an SCLA is also performed using Ansible within the SD module; this could be necessary in the event that the service is no longer required, or the provider is no longer deemed secure by the E2E Closed-Loop Manager, and a new instance of the Domain SCLA is then deployed for a different service provider.

The following Figure 4.4 displays the workflow of such a process, and afterwards we will provide a snippet of how the Ansible playbooks were developed in order to execute these operations in Figure 4.5.

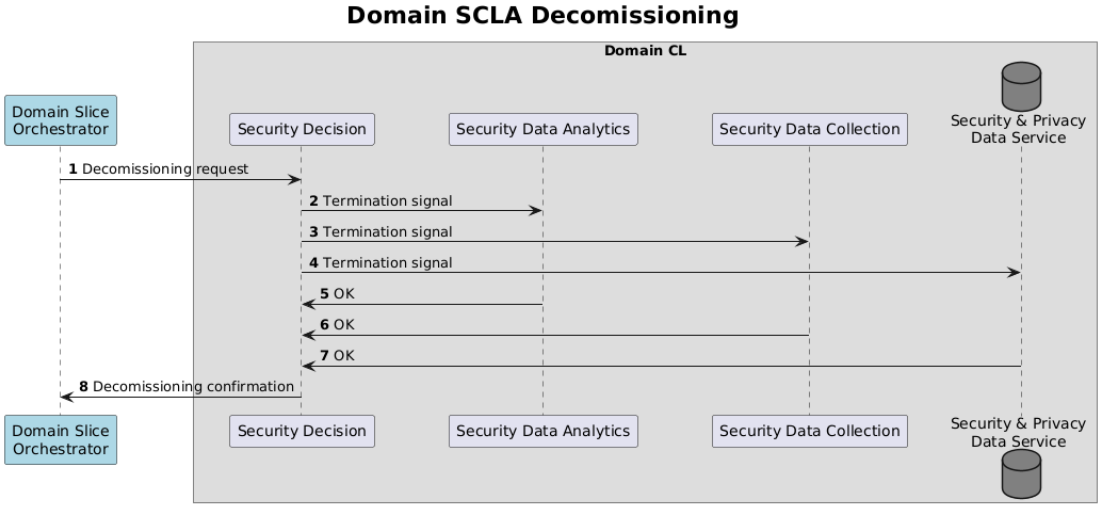

Figure 4.4 – Closed-Loop Decomissioning

All these operations were automated due to the capabilities of Ansible, in conjunction with APIs and proper data handling, which we will discuss further in this chapter. As mentioned above, the following is a snippet of one of the first Ansible playbooks developed that was used to deploy the SPDS container.

```yaml
- name: Deploy SPDS API and Kafka with Docker Compose
  collections:

  vars:
    ansible_become: yes
    ansible_become_method: sudo
    ansible_become_password: [redacted]
  tasks:
    - name: Set ANSIBLE_CONFIG to use default settings
      ansible.builtin.set_fact:
        ansible_env:
          ANSIBLE_CONFIG: /dev/null
    - name: Allow ansible user to have passwordless sudo
      ansible.builtin.lineinfile:
        path: /etc/sudoers
        line: 'root ALL=(ALL) NOPASSWD: ALL'
        validate: 'visudo -cf %s'
      become: yes

    - name: Ensure docker, docker-compose, and dependencies are installed
      ansible.builtin.apt:
        name:
          - docker.io
          - docker-compose
          - git
          - netcat-traditional
          - dos2unix
          - iputils-ping

        state: present
        update_cache: yes
      become: yes

    - name: Ensure Docker images are pulled
      community.docker.docker_image:
        name: "{{ item }}"
        source: pull
      loop:
        - pvieira23/spds-api:latest
        - wurstmeister/zookeeper
        - wurstmeister/kafka
      become: yes
```

Figure 4.5 – Database deployment snippet

The playbook, a set of instructions, acquires the required authentication to fetch the appropriate docker images and dependencies to deploy the container or containers in question and personalise them as it sees fit, according to the configuration data.

4.3 SCLA functionalities

After the system is deployed the data flow within the SCLA is done using two different tools, depending on the data traffic orientation. If the data is coming initially from an outside source and into the Decision/Analytics components, it will be relayed through a Kafka broker. We chose Kafka due to its scaling capabilities and built-in tools that not only aid in the load balancing of data transmission but also provide visual tools that help in testing and validating these transmissions. In Fiure.4.6 we are able to see an example of this data flow, particularly in the case of the forward chaining mechanism of the SD's inference engine.

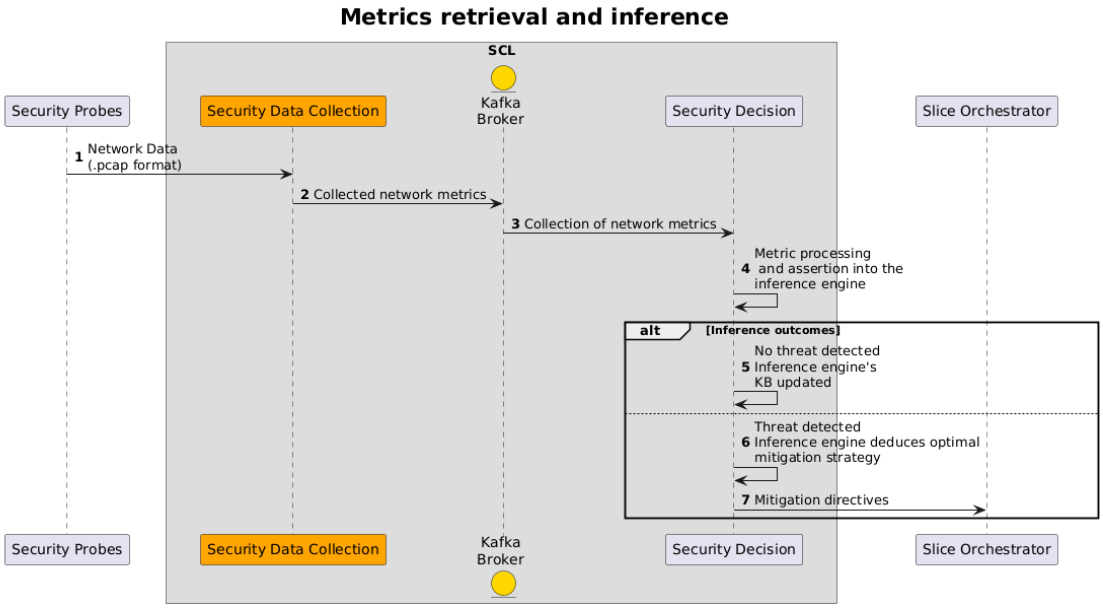

Figure 4.6 – Forward chaining inference data flow

As can be seen in the previous workflow, the network data enters the SCLA in the pcap format, the SDC module has more than one method of data handling for this pcap data, but for this workflow in particular, it simply extracts the pertinent network metrics that have been defined upon configuration of the SCLA, and sends it via the kafka broker, to the appropriate topic consumed by the SD. The SD then reads this data, in json format, and converts it before asserting it into the inference

engine's Knowledge Base (KB). The format used at the domain level can be seen in Figure 4.7 and can be updated according to policy specifications.

```
(deftemplate traffic-data
  (slot IP-Address (type STRING))
  (slot TTL-Value (type INTEGER))
  (slot Fragmentation-Status (type SYMBOL))  ; Use SYMBOL for yes/no or true/false values
  (slot Connection-Count (type INTEGER))
  (slot Port (type INTEGER))
  (slot TCP-Window-Size (type INTEGER))
  (slot Port-Traffic-Count (type INTEGER))
  (multislot Previous-TCP-Window-Size (type INTEGER))  ; List of previous window sizes
  (multislot SYN-to-ACK-Ratio (type INTEGER))  ; Two values: SYN count and ACK count
)
```

Figure 4.7 – Data template for assertion in the Inference Engine

After assertion, the Inference engine checks it against its rules file. These rules as well as their outcome if triggered, are based off of the configuration data received upon the deployment of the SCLA. Its possible that more than one rule is triggered upon analysis of the network traffic data, therefore features such as *salience* were important to define which rule is prevalent over adjacent rules. The Inference Engine used was achieved using CLIPSPy, which is a Python binding for the CLIPS (C Language Integrated Production System) expert system tool. We chose this tool due to the necessity of an efficient inference engine that could be used in a Python environment, with high scalability and adaptability to have its rules updated according to changing network requirements. The CLIPSPy also has the feature of being able to reach a conclusion in both forward and backward chaining methodologies. We have showcased how the forward chaining process works; it is data driven, the inference engine continuously evaluates network data, and based on that data it reaches a conclusion on whether the network is compliant with the defined rules, or not, and it defines the appropriate action to uphold this compliance. We will get into the backwards chaining method next, and this involves the collaboration with another of the SCLA's components. But before that we must divulge how this component was integrated into the SCLA and in a brief manner, discuss how it operates in order to collaborate with the SD component. To do so, the following Figure 4.8 can serve as an introduction for this process in the SDA component.

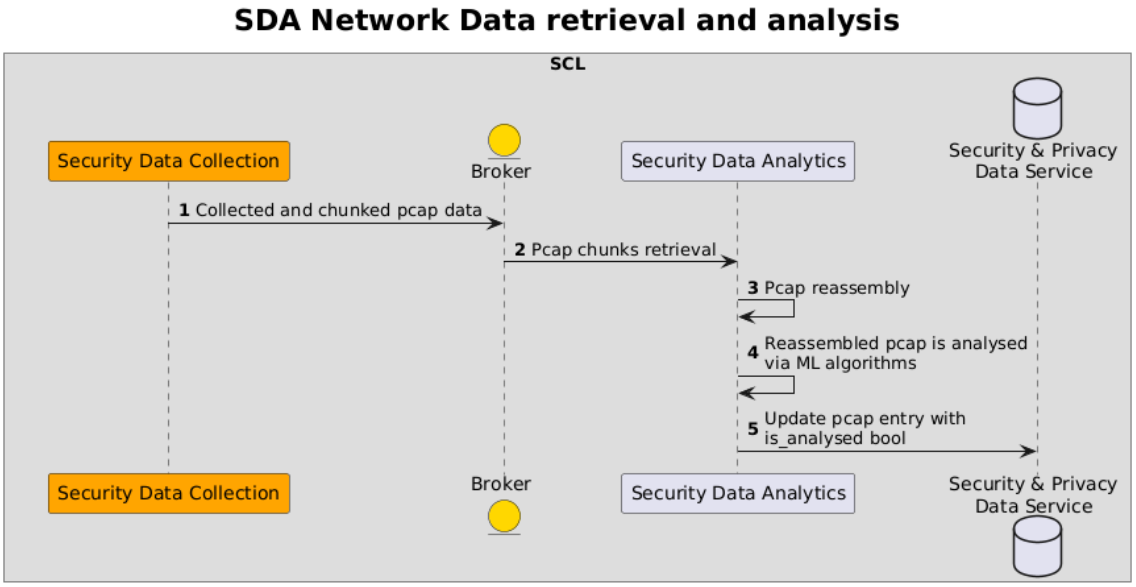

Figure 4.8 – SDA data handling

As demonstrated before, the SDC component collects network data from Security/Network Probes via APIs, and as we mentioned before it has more than one method to handle this data. What we displayed before in the forward chaining mechanism of the inference engine was how the SDC extracted relevant network metrics from the pcap data and sent it to the SD component to be infered upon. However in this scenario, the SDC module needs to relay this data to the SDA module, and the SDA module possesses algorithms that can only read entire pcap files for analysis. Since all internal communications in the SCLA are done via Kafka broker, and the broker has a message size limit of 1Mb per message, we needed to chunk the pcap file, tag each chunk for identification, and send the chunks via broker to the SDA module. Here in the SDA module, using the same tools used to chunk the pcap file, we reassembled the pcap file by identifying which chunks belonged to each pcap and how many chunks were created in order to assure no data loss between both modules. After this process the data was prepared to be fed into the ML algorithm for analysis and possible threat detection. The following Figure 4.9 depicts the next steps in this process and after showcasing it, we can finally delve into the details of the backwards chaining method in the Inference Engine.

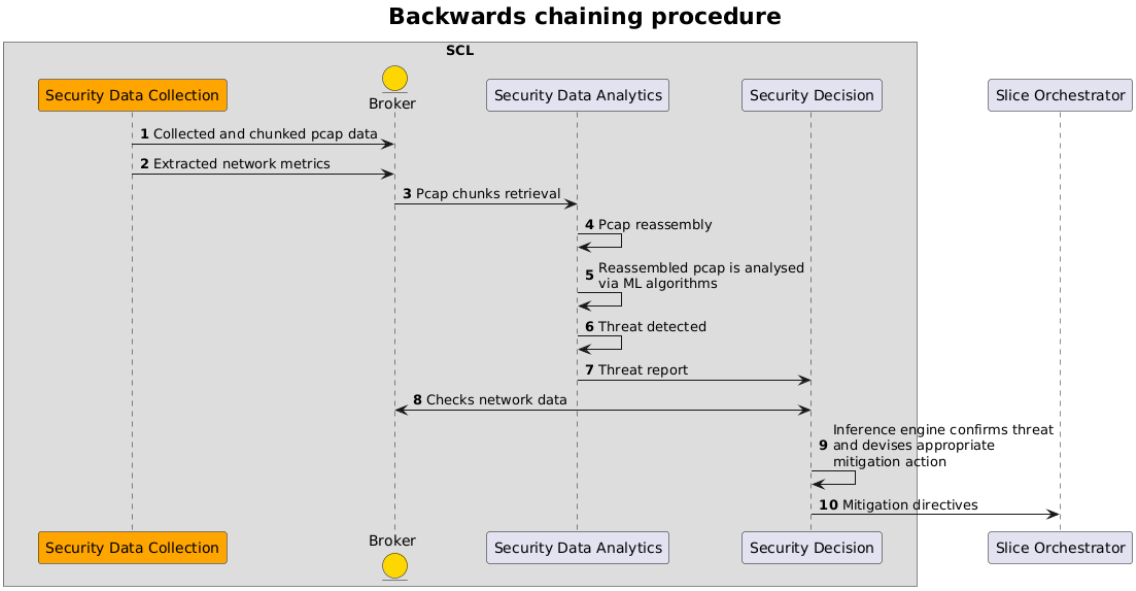

Figure 4.9 – Backwards chaining method

The process of backwards chaining, unlike the previously shown forwards chaining, is not data-driven. In this scenario, the SDA component presents a threat report with an identified threat, and the goal for the inference engine will be to validate the threat based on network metrics collected by the SDC. The threat report contains data that facilitates the process of searching for this data like suspicious IP addresses, thus accelerating the inference process, and assuring a higher degree of accuracy since both the inference engine and ML algorithm detected and validated the threat.

In the event that a threat is identified by the SDA module, but cannot be validated by the inference engine, the SD stores the report in the event that another report follows with the same pattern. If that is the case the SD module escalates the issue to the E2E SCLA Manager, sending both the threat report from the SDA and the usually scheduled service report containing metrics related to the performance of the service provider. If the E2E SCLA Manager is unable to find and mitigate the threat, the following procedure is to verify the accuracy of the ML algorithm in the SDA with a prepared set of data for this test. If the outcome is above a pre-determined threshold and the accuracy is reliable, the issue is then escalated to the E2E Slice Orchestrator. If otherwise, the accuracy is below the pre-determined

threshold, the SD module begins the SDA re-training procedure using Ansible to automate the process. This occurrence can be visualised in Figure 4.10.

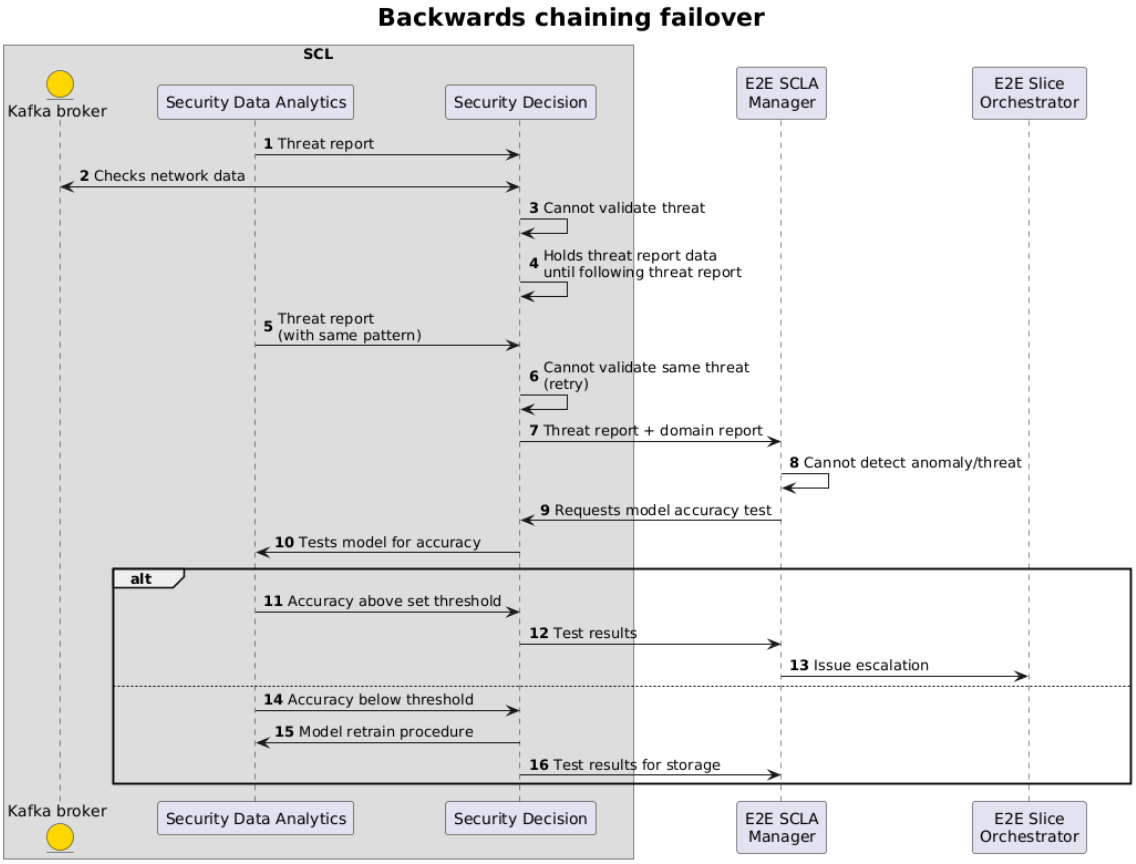

Figure 4.10 – Backwards chaining method failover

4.4 Testing Framework

This section provides an overview of the tests conducted to evaluate the SCLA system's effectiveness in real-time threat detection, provider compliance monitoring, and handling high-stress scenarios. The revised research questions addressed are as follows:

- **How effectively can the system detect and mitigate network security threats in real-time?**

- **How reliably does the system ensure provider-level performance monitoring and compliance with SLAs/SLOs?**
- **How does the system handle high-stress scenarios to maintain compliance and performance within the Security Decision component?**

4.4.1 Test 1: Real-Time Threat Detection and Mitigation (Domain Level)

Objective:

Evaluate the effectiveness of the SD component in detecting and mitigating network security threats in real-time at the domain level using forward chaining (as described in Fig.4.6).

Setup:

- **Synthetic Traffic Preparation:** Use pcap files containing various protocols (e.g., TCP, UDP, IP) to simulate realistic network conditions.
- **Data Feeding:** Feed traffic data to the SDC component through established APIs between SDC and Security Probes.
- **Configuration:** Configure the SD component with appropriate rules and policies defined based on the received SCLA configuration data.

Procedure:

- Deploy Security Probes containing pcap files.
- SDC collects and pre-processes the traffic data.
- The SD component processes the received data asynchronously (via Kafka broker), using forward chaining to detect policy non-compliance.

- Upon detecting an anomaly, the SD component sends mitigation instructions to the Domain Slice Orchestrator (DSO).
- The DSO sends confirmation messages to the SD component to verify mitigation application.

Metrics:

- **Detection Time:** Time taken to detect a threat after it occurs.
- **Mitigation Effectiveness:** Ability of the system to neutralise identified threats by enforcing predefined policies.

4.4.2 Test 2: Provider-Level Compliance Monitoring (E2E Level)

Objective:

Assess the reliability of the E2E SCLA in monitoring service provider performance and ensuring compliance with SLAs/SLOs across different domain-level SCLA instances.

Setup:

- **Performance Degradation Simulation:** Introduce scenarios where providers exceed packet rate thresholds or fail to meet other performance metrics.
- **Reporting Mechanism:** Ensure that domain SD components report performance data to the E2E SCLA.

Procedure:

- Adjust network parameters to simulate provider underperformance.

- Domain SD components aggregate and send performance reports to the E2E SCLA.
- E2E SCLA evaluates performance data against SLA/SLO compliance criteria.
- If non-compliance is detected, the E2E SCLA penalises the provider or initiates a provider switch.
- Monitor and record service performance changes following the corrective action.

Metrics:

- **Detection Accuracy:** Rate of correctly identified non-compliant providers.
- **Response Time:** Time taken from non-compliance detection to action.
- **Service Continuity:** Evaluate service impact during provider switch or penalty enforcement.

4.4.3 Test 3: Stress Test on Security Decision Component (SD) under High-Load Conditions

Objective:

Evaluate the SD component's inference performance and compliance maintenance capability under high-stress scenarios with elevated processing rates.

Setup:

- **Traffic Rate Simulation:** Configure the system to handle high processing rates (e.g., 100, 1000, 5000, 10000 reports per second).
- **Inference Capacity:** Use CLIPS-based inference engine to process data and monitor average inference time across different packet rates.

Procedure:

- Simulate high-volume traffic feeding into the SD component.
- Measure inference times and observe if the system maintains compliance under high load.
- Log inference performance and compliance status across various processing rates.

Metrics:

- **Average Inference Time:** Measure how inference time scales with increased processing rates.
- **System Stability:** Evaluate system's ability to maintain performance without exceeding acceptable inference times.
- **Policy Compliance Rate:** Measure the frequency of successful policy enforcement under high load.

4.5 Summary

In Chapter 4, the focus is on implementing the SCLA model introduced in the previous chapter, detailing the tools and methods used to deploy and manage both domain-level and E2E SCLA systems. The E2E SCLA is designed to oversee domain-level SCLAs, assessing service provider compliance with high-level policies and penalising non-compliant providers when necessary. The E2E SCLA is streamlined compared to its domain counterpart, using containerised services for functions like data collection and decision-making.

The SD component, a core element of the SCLA, is implemented using Ansible to automate SCLA deployment, configuration updates, and decommissioning. The

SD module can handle complex operations such as endpoint reconfiguration and performance monitoring, integrating seamlessly with security probes and external components via APIs.

Data flow within the SCLA is facilitated by Kafka, which allows for efficient data transmission between components and supports the forward and backward chaining inference mechanisms of the SD's engine. The SD uses CLIPSPy to manage these inference processes, allowing it to detect threats based on real-time network data (forward chaining) or validate alerts raised by the SDA module (backward chaining). To ensure accuracy, a failover mechanism is in place: if the SDA's threat detection results are inconclusive, the SD initiates retraining of the SDA's ML model or escalates to the E2E orchestrator.

Finally, a testing framework is presented to validate the SCLA's effectiveness under various scenarios:

- **Real-Time Threat Detection:** Assessing the SD's ability to detect and mitigate threats at the domain level.
- **Provider-Level Compliance Monitoring:**Evaluating E2E SCLA's performance in monitoring provider compliance with SLAs.
- **Stress Test:** Measuring the SD's inference speed and policy compliance under high-load conditions.

This chapter lays the groundwork for practical application by demonstrating the SCLA's operational resilience, adaptability, and compliance management capabilities, setting up for the evaluation and refinement stages in subsequent chapters.

5 Testing and Results

5.1 Introduction

In this chapter, we present the testing methodologies and results obtained from evaluating the SCLA system, with a focus on the SD component both in its Domain and E2E capabilities. The tests aim to assess the system's effectiveness in real-time threat detection and mitigation, provider-level compliance monitoring, and performance under high-stress conditions. The results are analysed to determine how well the system meets the objectives outlined in chapter 1.

The chapter is organised as follows:

- **Section 5.2:** Discusses the results of Test 1, focussing on real-time threat detection and mitigation at the domain level.
- **Section 5.3:** Presents the findings from Test 2, evaluating provider-level compliance monitoring at the E2E level.
- **Section 5.4:** Details the outcomes of Test 3, analysing the performance of the SD component under high-stress scenarios.

- **Section 5.5:** Provides a summary of the key findings from the tests.

5.2 Test 1: Real-Time Threat Detection and Mitigation at the Domain Level

5.2.1 Objective

The objective of Test 1 was to evaluate the effectiveness of the Sd component in detecting and mitigating network security threats in real-time at the domain level using forward chaining, as described in Fig. 4.6.

5.2.2 Methodology

Setup

- **Synthetic Traffic Preparation:** We generated pcap files containing various protocols (TCP, UDP, IP) to simulate realistic network conditions, including normal traffic and malicious packets resembling a Distributed Denial of Service (DDoS) attack
- **Data Feeding:** The synthetic traffic was fed to the SDC component through APIs connected to security probes.
- **Configuration:** The SD component was configured with a set of rules and policies derived from the SCLA configuration data, focussing on detecting anomalies such as unusually high packet rates from a single IP address.

Procedure

- **Deployment:** Security probes were deployed, and the SCLA system was initiated as seen in Fig. 5.1.

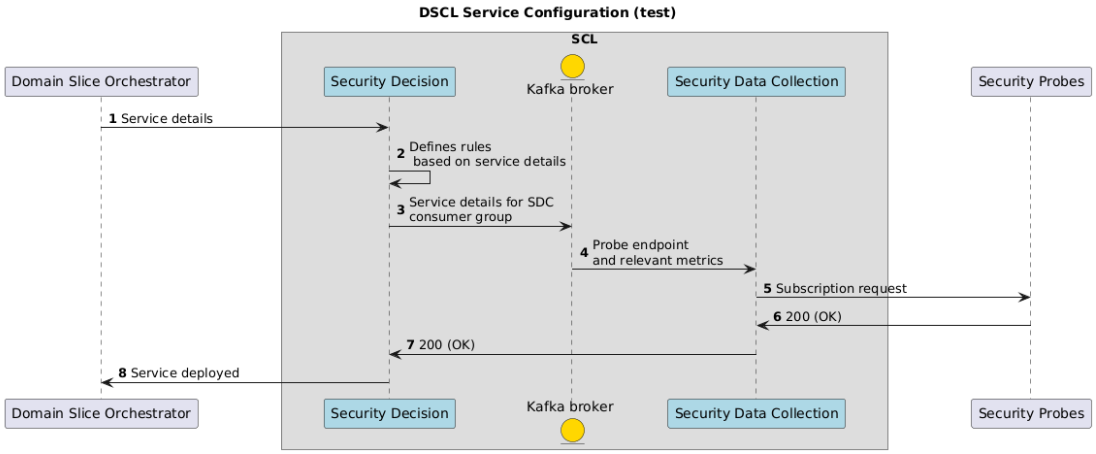

Figure 5.1 – DSCL service configuration

- **Data Collection:** The SDC component collected and pre-processed the traffic data in real-time.

- **Inference Process:** The SD component processed the received data asynchronously via the Kafka broker, utilising forward chaining to detect policy non-compliance.

- **Threat Detection:** Upon detecting an anomaly indicative of a DDoS attack (according to defined policies), the SD component generated an alert.

- **Mitigation Actions:** The SD component sent mitigation instructions to the DSO to enforce actions such as blocking the offending IP address or rate-limiting as seen in Fig. 5.2.

- **Confirmation:** The DSO sent confirmation messages to the SD component to verify the application of mitigation measures.

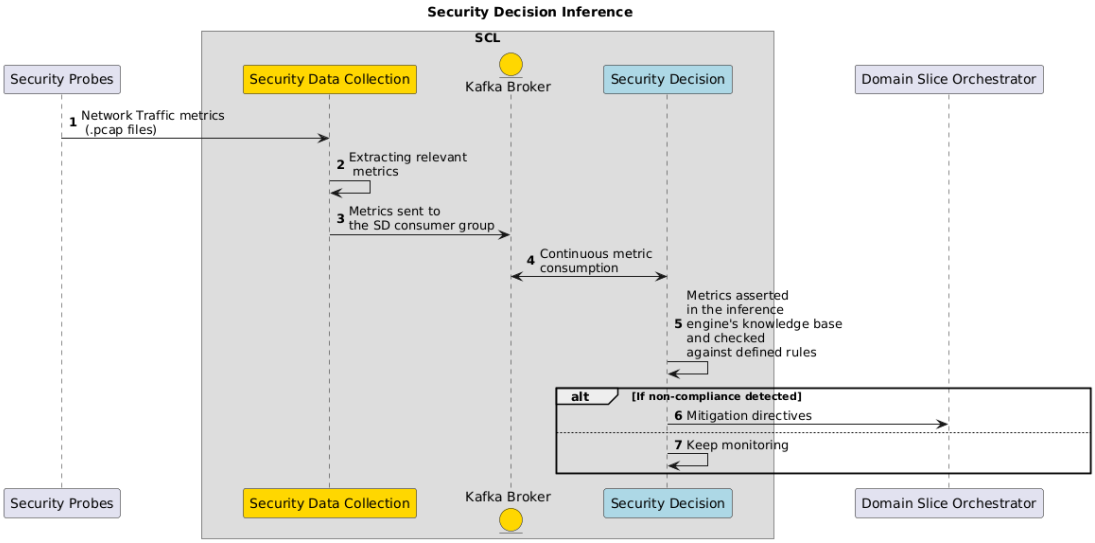

Figure 5.2 – Test 1 Workflow

5.2.3 Results

Detection Time

The system demonstrated rapid detection capabilities. The time taken to detect the threat after its initiation was consistently under 10 miliseconds, as shown in Fig. 5.3.

Mitigation Effectiveness

- **Threat Neutralisation:** The system successfully neutralised the simulated DDoS attack by promptly enforcing the predefined policies.
- **Policy Enforcement:** The SD component was able to detect policy non-compliance in a time frame that allowed the system to apply the appropriate directives for each non-compliance event, therefore ensuring no damage to the network was significant and QoS would remain at a similar level throughout the event.

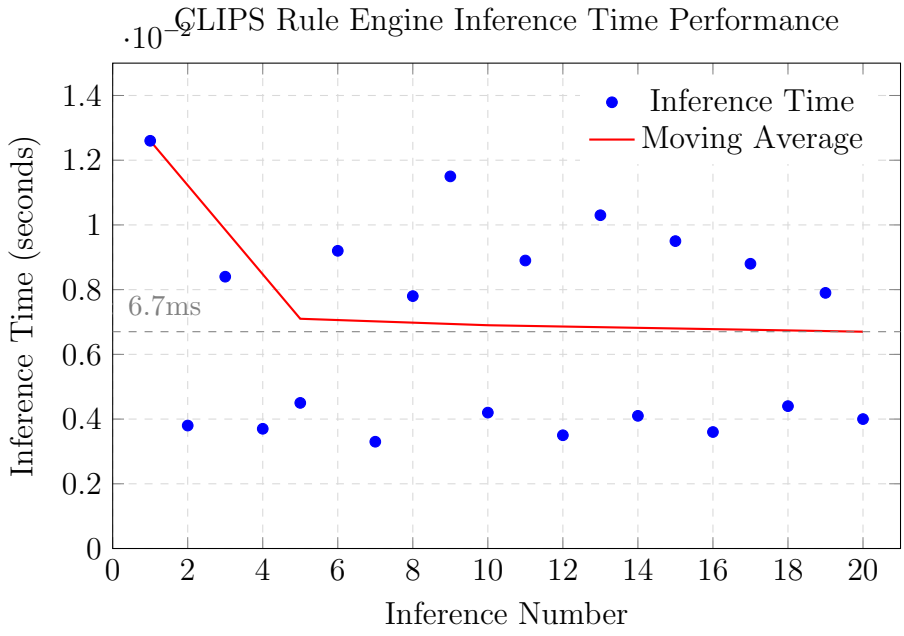

Figure 5.3 – Performance analysis of the CLIPS rule engine showing individual inference times and moving average.

5.2.4 Discussion

The results indicate that the SD component effectively detects and mitigates network security threats in real-time at the domain level. The use of forward chaining in the inference engine enabled rapid processing of incoming data, ensuring timely responses to threats. The average inference time of 6.7 miliseconds confirms the system's capability to meet real-time operational requirements.

5.3 Test 2: Provider-Level Compliance Monitoring at E2E Level

5.3.1 Objective

Test 2 aimed to assess the reliability of the E2E SCLA in monitoring service provider performance and ensuring compliance with Service Level Agreements (SLAs) and

Service Level Objectives (SLOs) across different domain-level SCLA instances.

5.3.2 Methodology

Setup

- **Performance Degradation Simulation:** Scenarios were introduced where providers exceeded packet rate thresholds or failed to meet other performance metrics defined in the SLAs.

- **Reporting Mechanism:** Domain SD components were configured to report performance data to the E2E SCLA at regular intervals.

Procedure

- **Simulating Underperformance:** Network parameters were adjusted to simulate scenarios where providers did not meet the agreed performance metrics.

- **Data Aggregation:** Domain SD components aggregated performance metrics and sent reports to the E2E SCLA as seen in Fig. 5.4.

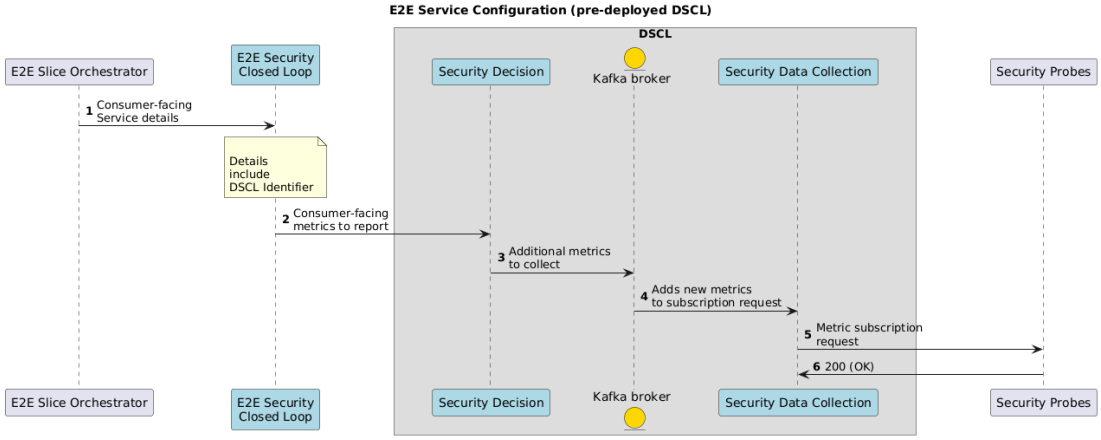

Figure 5.4 – E2E service configuration

- **Compliance Evaluation:** The E2E SCLA evaluated the performance data against SLA/SLO compliance criteria.
- **Corrective Actions:** Upon detecting non-compliance, the E2E SCLA penalised the provider by reallocating resources or initiating a provider switch as seen in Fig. 5.5.

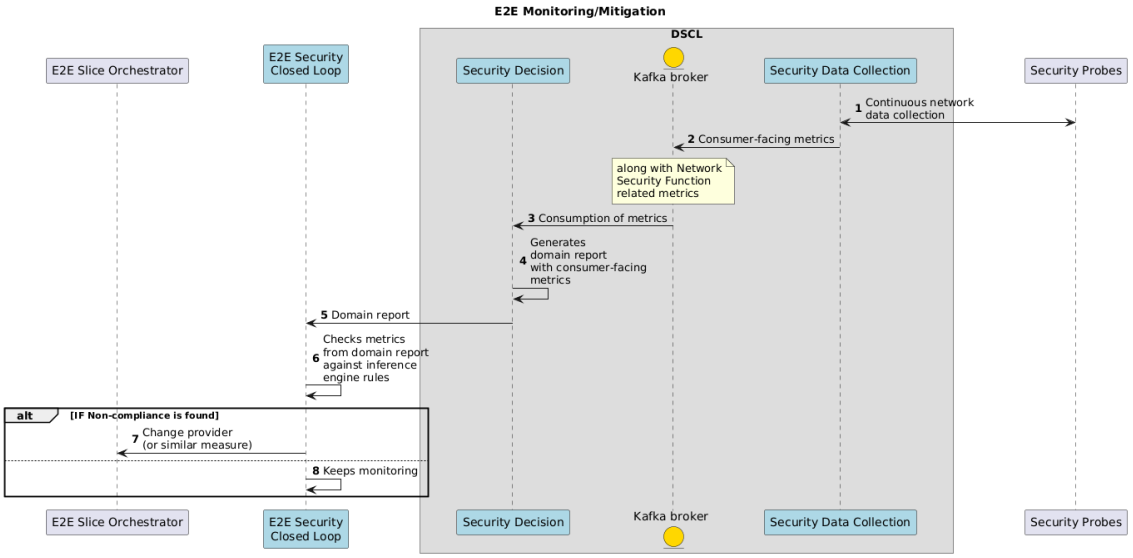

Figure 5.5 – E2E monitoring/mitigation workflow

5.3.3 Results

Detection Accuracy

The E2E SCLA was configured with rules aligned to the defined SLAs and SLOs. As a result, it successfully identified all instances of provider non-compliance with a detection accuracy of 100%, exhibiting no false positives or negatives. This high accuracy is attributable to the fact that non-compliance is only detected when a service experiences a failure, ensuring that normal operational conditions do not trigger incorrect alerts. These findings demonstrate that, when equipped with a well-defined set of security policies, the E2E SCLA effectively detects and mitigates abnormalities, provided that the system remains free from systematic failures.

Response Time

The average time taken from non-compliance detection to action implementation was 50 miliseconds, the inference time was even lower as the graph from Fig. 5.6. shows, since the domain reports do not occur as often as metric reports within the DSCL.

Service Continuity

- **Minimal Impact:** The corrective actions applied can be executed without degrading QoS, so no significant impact on service continuity is expected.

- **Improved Performance:** Due to the fact that within the system the SCLA is deployed, the provider is chosen based on previous performance, changing provider returns the performance metrics into compliance with the SLAs.

5.3.4 Discussion

The E2E SCLA demonstrated high reliability in monitoring provider performance and enforcing compliance with SLAs/SLOs. the system's prompt detection and response mechanisms ensured that any deviations were swiftly corrected with minimal disruption to services. The inference times (shown in Figures 5.6,5.7,5.8) remained within acceptable limits even as processing rates increased, indicating scalability and robustness of the system.

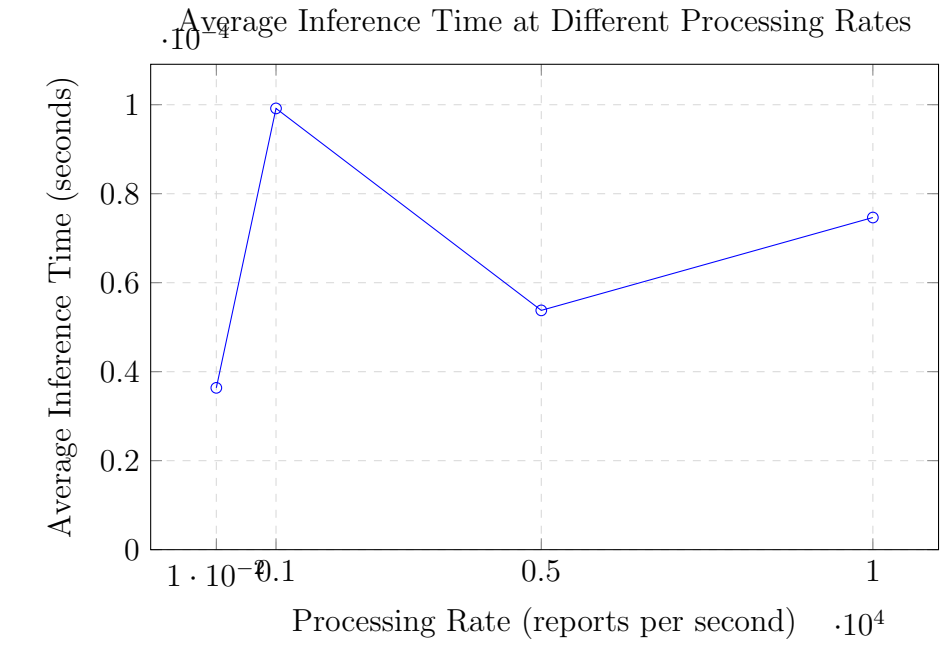

Figure 5.6 – Average Inference Time at Different Processing Rates

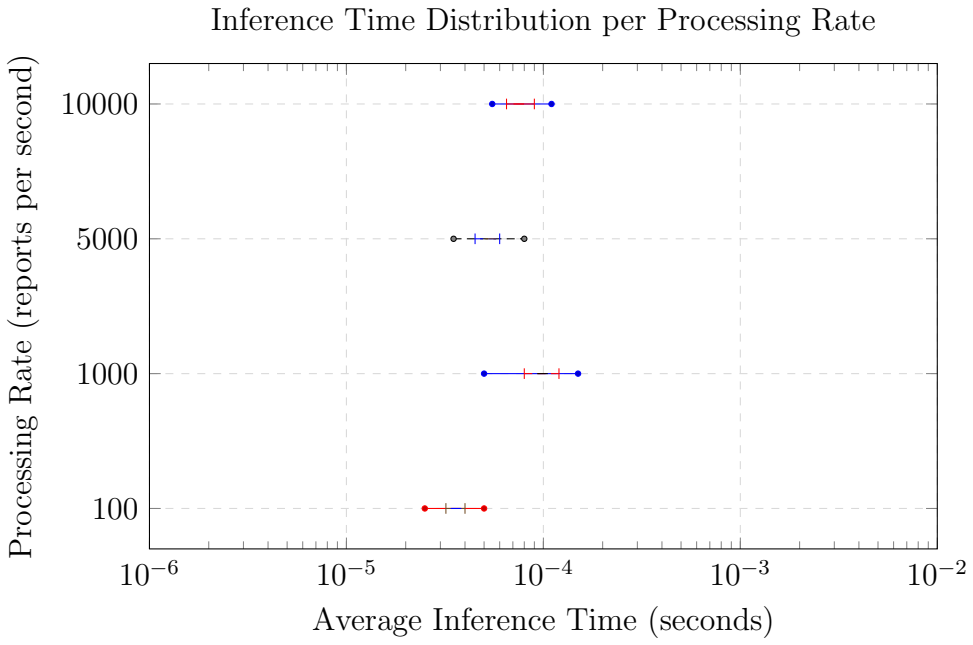

Figure 5.7 – Inference Time Distribution per Processing Rate

Figure 5.8 – Inference Time vs. Rules Fired

5.4 Test 3: Stress Test on Security Decision Component Under High-Load conditions

5.4.1 Objective

The objective of Test 3 was to evaluate the SD component's inference performance and compliance maintenance capability under high-stress scenarios with elevated processing rates.

5.4.2 Methodology

Setup

- **Traffic Rate Simulation:** The system was configured to handle high processing rates of 100, 1000, 5000, 10000, and 100000 reports per second.
- **Inference Capacity:** The CPLIPSPy-based inference engine was used to

process data, and the average inference time was monitored across different packet rates.

Procedure

- **Simulating High-volume Traffic:** High-volume synthetic traffic was fed into the SD component to simulate stress conditions.
- **Performance Measurement:** Inference times were measured, and system stability was observed at each processing rate.
- **Data Logging:** Inference performance and compliance status were logged for analysis.

5.4.3 Results

Average Inference Time

The average inference times at different processing rates are illustrated in Figures 5.9 to 5.11.

System Stability

The system was able to maintain consistent inference times across all processing rates. There was no evidence of system overload or degradation in performance, even at the highest processing rate of 100000 packets per second.

5.4.4 Discussion

The SD component demonstrated robust performance under high-stress conditions. The inference engine efficiently processed large volumes of data without significant

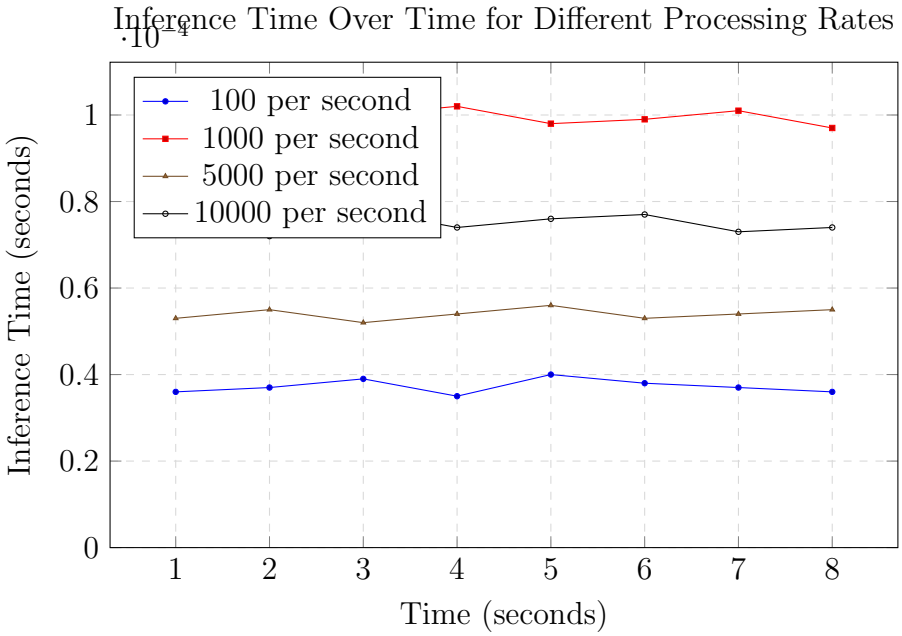

Figure 5.9 – Inference Time Over Time for Different Processing Rates

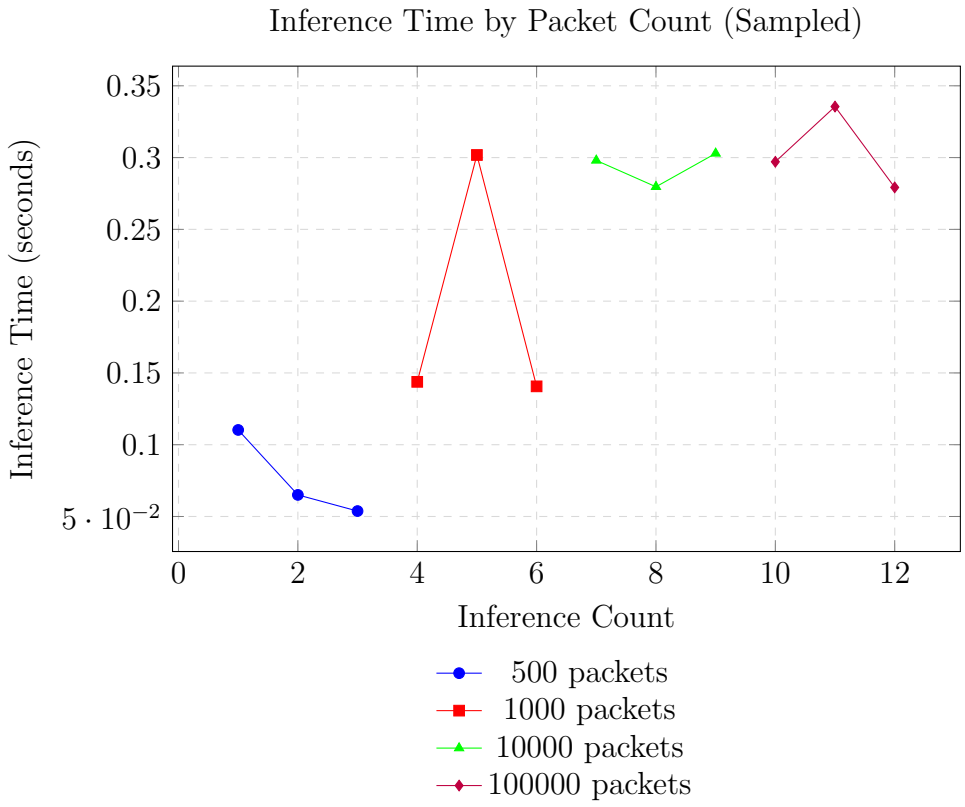

Figure 5.10 – Inference Time by Packet Count (Sampled)

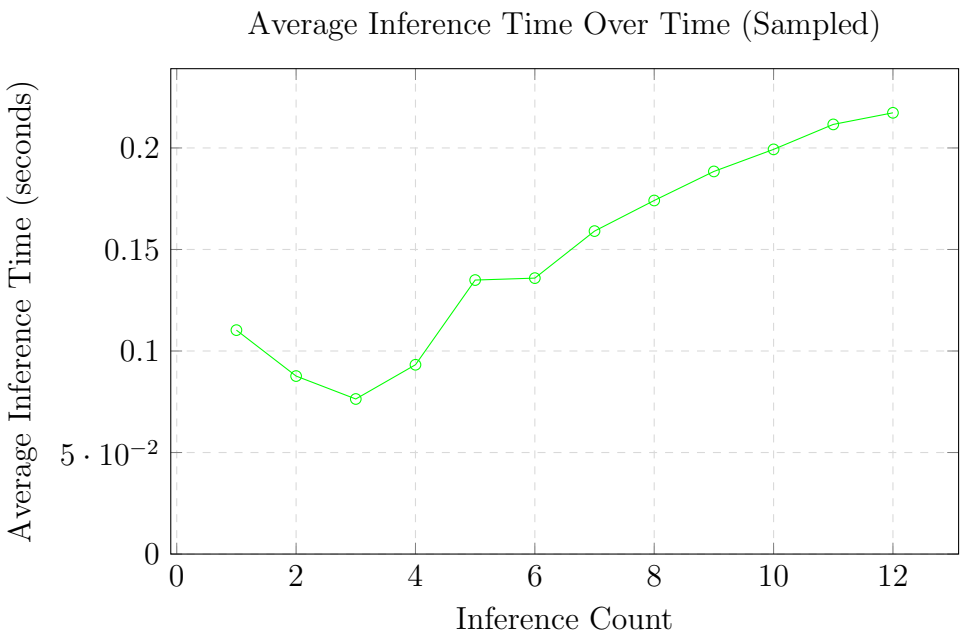

Figure 5.11 – Average Inference Time Over Time (Sampled)

increases in inference time or loss of compliance. the results suggest that the system is capable of scaling to meet the demands of high-throughput network environments.

5.5 Summary of Key Findings

The testing of the SCLA system yielded the following key findings:

- **Real-Time Threat Detection:** The SD component effectively detected and mitigated threats in real-time at the domain level, with average inference times suitable for operational requirements
- **Provider-Level Compliance Monitoring:** The E2E SCLA reliably monitored provider performance, ensuring compliance with SLAs/SLOs and swiftly implementing corrective actions when necessary.
- **High-Stress Performance:** Under high-load conditions, the SD component maintained consistent performance and policy compliance, demonstrating scalability and robustness.

6 Conclusion and Future Work

6.1 Introduction

This thesis was set out to address the growing challenges of network security in the evolving landscape of 5G and beyond technologies. by developing an SCLA system with a focus on the SD component, we aimed to enhance real-time threat detection, provider-level performance monitoring, and system scalability under high-stress conditions.

6.2 Summary of Contributions

6.2.1 Real-Time Threat detection and Mitigation

The SCLA system effectively used the capabilities of the inference engine capabilities to detect and mitigate network security threats in real-time. The integration of the CLIPSPy inference engine and Kafka for data streaming, along with an efficient use of batching and parallel processing, allowed for rapid processing of incoming data, achieving average inference times as low as 6.7 miliseconds. This performance meets

the stringent requirements for real-time operations in modern network environments.

6.2.2 Provider-Level Compliance Monitoring

At the E2E level, the SCLA system demonstrated high reliability in monitoring service provider performance against SLAs and SLOs. The system accurately detected instances of non-compliance and implemented corrective actions promptly, ensuring minimal disruption to services and maintaining overall network integrity.

6.2.3 Scalability and Performance Under High-Stress Conditions

The SD component maintained a consistent performance under high-load scenarios, processing up to 100000 reports per second without significant degradation. The system's modular architecture and efficient use of resources contributed to its scalability and robustness, indicating its suitability for deployment in large-scale, high throughput network environments.

6.2.4 Innovative Integration of Technologies

The project introduced several novel aspects, including:

- **Dual-Layer SCLA Architecture:** Implementing both domain-level and E2E SCLA instances to provide comprehensive security and performance monitoring.
- **Combination of forward and Backward Chaining:** Enhancing the threat detection process by using both data-driven and goal-driven inference methods, even though we did not include the results of the backward chaining method due to the ongoing nature of this process.

- **Automated Deployment and Configuration:** Using Ansible for automated deployment, reconfiguration, and decommissioning of SCLA components, enabling rapid adaptation to changing network conditions.

6.2.5 Constraints

- **Focus on DDoS Attacks:** The testing primarily addressed DDoS attacks, limiting the assessment of the system's effectiveness against other types of threats such as malware or insider attacks.

- **Synthethic Data Usage:** Yhe use of synthethic traffic and simulated scenarios may not capture all the complexities of real-world network environments. In a further stage of this process, which will be the validation phase, we should be able to test the system in an appropriate testbed, to better emulate a real-world network scenario.

- **Scalability Constraints:** While the system performed well up to 100000 reports/packets per second, further testing is required to assess performance at even higher processing rates and in distributed environments. Being able to access more resources would take full advantage of the scalability mechanisms provided by Kafka and Ansible, that even though provided an efficient usage of resources, were limited by the capabilities of the machine wherein the tests were performed.

- **Validity of Policies:** The policies used in the tests are general rules that are able to detect abnormalities based on network traffic. The definition of SLAs and or SLOs was outside of the scope of our project. However, being able to develop a system that can interpret High-Level Policies and turn them into low-level network rules or configurations would ensure that the service is perfectly fitted to the user's requirements. As it stands the system can be tailored to the configuration requested, however, the security aspect of such configuration is still an underdeveloped aspect.

6.3 Recommendations for Future Work

6.3.1 Expanding Threat Coverage

Future research should focus on testing the SCLA system against a broader range of security threats, including advanced persistent threats, zero-day exploits, and insider attacks. Integrating additional machine learning models specialised in detecting various threat types could enhance the system's versatility.

6.3.2 Real World Deployment

Deploying the SCLA system in live network environments would provide valuable insights into its operational effectiveness and reveal any practical challenges not apparent in simulated settings. Collaboration with industry partners could facilitate access to real-world data and infrastructure.

6.3.3 Advanced Machine Learning Integration

Exploring the integration of deep learning techniques and adaptive learning algorithms could improve the SDA component's ability to detect evolving and sophisticated threats. Implementing federated learning could also enhance data privacy and enable collaborative threat intelligence sharing across domains.

6.3.4 Scalability Enhancements

To further improve scalability, future work could investigate:

- **New Infrastructure Technologies:** Implementing the system in a Kubernetes environment, and finding a way to combine it with Ansible, would not

only increase its self-healing capabilities as well as its scalability and efficient management.

- **Resource Optimisation:** Implementing dynamic resource allocation strategies, pivoting on the containerised nature of the system, would optimise performance under varying load conditions.

6.3.5 Enhanced Policy Management

Developing more sophisticated policy compliance mechanisms, possibly incorporating artificial intelligence for the automation of policy updates and conflict resolution, could enhance the system's adaptability and reduce manual intervention.

6.4 Final Reflections

The project has demonstrated the potential of closed-loop automation to improve network security and management. The integration of advanced inference engines, machine learning algorithms, and automation tools has resulted in a system capable of responding swiftly and effectively to security threats while maintaining compliance with performance standards.

The challenges encountered, such as optimising system performance under high-stress conditions and ensuring accurate threat detection, have provided valuable learning experiences. Overcoming these challenges has strengthened the design of the system and highlighted areas for future improvement.

6.5 Summary

In conclusion, the development and testing of the SCLA system represent a significant step towards realising secure, automated network management in the era of 5G

and beyond. The findings confirm that closed-loop automation, when implemented effectively, can greatly enhance network resilience, efficiency, and security.

By addressing current limitations and pursuing the recommended future work, there is considerable potential to further advance the capabilities of the SCLA system. This work lays a solid foundation for ongoing research and development, contributing to the broader goal of securing modern telecommunications infrastructures against evolving threats [Alemany et al., 2024].

References

IEEE Xplore Full-Text PDF:, a. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9397776.

IEEE Xplore Full-Text PDF:, b. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9397776.

IEEE Xplore Full-Text PDF:, c. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9665048.

IEEE Xplore Full-Text PDF:, d. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9797849.

IEEE Xplore Full-Text PDF:, e. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9502641.

IEEE Xplore Full-Text PDF:, f. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9099866.

IEEE Xplore Full-Text PDF:, g. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8637976.

Interface to Network Security Functions (i2nsf), h. URL https://datatracker.ietf.org/wg/i2nsf/charter/.

The OpenFlow Protocol Feature Overview and Configuration Guide. i.

Network Security Concepts, Dangers, and Defense Best Practical. *Computer Engineering and Intelligent Systems*, March 2023. doi: 10.7176/CEIS/14-2-03. URL https://iiste.org/Journals/index.php/CEIS/article/view/60555.

5G Americas. 5g security white paper, 2019. Retrieved from $5G_Americas_5G_Security_White_Paper_Final.pdf$. 51

Khizar Abbas, Muhammad Afaq, Talha Ahmed Khan, Asif Mehmood, and Wang-Cheol Song. IBNSlicing: Intent-Based Network Slicing Framework for 5G Networks using Deep Learning. In *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 19–24, Daegu, Korea (South), September 2020a. IEEE. ISBN 978-89-950043-8-8. doi: 10.23919/APNOMS50412.2020.9237008. URL https://ieeexplore.ieee.org/document/9237008/.

Nabeel Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. *Journal of Network and Computer Applications*, 149:102472, 2020b. 22

Ibrahim Afolabi, Tarik Taleb, Konstantinos Samdanis, Adlen Ksentini, and Hannu Flinck. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys & Tutorials*, 20(3):2429–2453, 2018. doi: 10.1109/COMST.2018.2815638. URL https://ieeexplore.ieee.org/document/8323459. 17

Md Morshed Alam, Israt Jahan, and Weichao Wang. IoTWarden: A Deep Reinforcement Learning Based Real-time Defense System to Mitigate Trigger-action IoT Attacks, January 2024. URL http://arxiv.org/abs/2401.08141. arXiv:2401.08141 [cs].

P. Alemany, R. Muñoz, R. Vilalta, Ll. Gifre, R. Martínez, R. Casellas, M. Castro,

P. Ferreira, D. Moreira, J. García, J. Cunha, I. Núñez, G. Gómez, S. Castro, A. Pastor, and D. López. Security and trust in open and disaggregated 6g networks. In *2024 24th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, 2024. doi: 10.1109/ICTON62926.2024.10647935. 110

Pol Alemany, Anton Roman, Ricard Vilalta, Ana Pol, Jose Bonnet, Evgenia Kapassa, Marios Touloupou, Dimosthenis Kyriazis, Panagiotis Karkazis, Panagiotis Trakadas, Josep Martrat, Ramon Casellas, Ricardo Martinez, and Raul Munoz. A KPI-Enabled NFV MANO Architecture for Network Slicing with QoS. *IEEE Communications Magazine*, 59(7):44–50, July 2021. ISSN 0163-6804, 1558-1896. doi: 10.1109/MCOM.001.2001077. URL https://ieeexplore.ieee.org/document/9502641/.

Pol Alemany, Alejandro Molina, Cyril Dangerville, Rodrigo Asensio, Dhouha Ayed, Raul Muñoz, Ramon Casellas, Ricardo Martínez, Antonio Skarmeta, and Ricard Vilalta. Management and enforcement of secured E2E network slices across transport domains. *Optical Fiber Technology*, 73:103010, October 2022. ISSN 10685200. doi: 10.1016/j.yofte.2022.103010. URL https://linkinghub.elsevier.com/retrieve/pii/S1068520022001936.

Allied Telesis. *The OpenFlow Protocol Feature Overview and Configuration Guide*, 2022. URL https://www.alliedtelesis.com. C613-2208. 11

Chamitha De Alwis, Anshuman Kalla, Quoc-Viet Pham, Pardeep Kumar, Kapal Dev, Won-Joo Hwang, and Madhusanka Liyanage. Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research. *IEEE Open Journal of the Communications Society*, 2:836–886, 2021a. ISSN 2644-125X. doi: 10.1109/OJCOMS.2021.3071496. URL https://ieeexplore.ieee.org/document/9397776/.

Chamitha De Alwis, Anshuman Kalla, Quoc-Viet Pham, Pardeep Kumar, Kapal Dev, Won-Joo Hwang, and Madhusanka Liyanage. Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research. *IEEE Open Journal of the Communications Society*, 2:836–886, 2021b. ISSN 2644-125X. doi: 10.1109/OJCOMS.2021.3071496. URL https://ieeexplore.ieee.org/document/9397776/.

Esmaeil Amiri, Massoud Reza Hashemi, and Khalilollah Raeisi Lejjy. Policy-Based Routing in RIP-Hybrid Network with SDN Controller. 2018.

Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. 5g network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges. *Computer Networks*, 167:106984, 2020. doi: 10.1016/j.comnet.2019.106984. URL https://www.sciencedirect.com/science/article/pii/S1389128619304773. 20

Cataldo Basile, Fulvio Valenza, Antonio Lioy, Diego R. Lopez, and Antonio Pastor Perales. Adding Support for Automatic Enforcement of Security Policies in NFV Networks. *IEEE/ACM Transactions on Networking*, 27(2):707–720, April 2019. ISSN 1063-6692, 1558-2566. doi: 10.1109/TNET.2019.2895278. URL https://ieeexplore.ieee.org/document/8637976/.

Bruno Lopes Alcantara Batista and Marcial Porto Fernandez. PonderFlow: A New Policy Specification Language to SDN OpenFlow-based Networks. 2014.

Dallal Belabed, Bruno Vidalenc, and Alejandro Molina Zarca. WP3 / T3.3 THALES SIX GTS France. a.

Dallal Belabed, Bruno Vidalenc, and Alejandro Molina Zarca. WP3 / T3.3 THALES SIX GTS France. b.

Pablo Benlloch-Caballero, Qi Wang, and Jose M. Alcaraz Calero. Distributed dual-layer autonomous closed loops for self-protection of 5g/6g iot networks from distributed denial of service attacks. *Computer Networks*, 222:109526, 2023. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2022.109526. URL https://www.sciencedirect.com/science/article/pii/S1389128622005606. 35

Mounir Bensalem, Jasenka Dizdarević, and Admela Jukan. Benchmarking various ml solutions in complex intent-based network management systems. In *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 476–481. IEEE, 2022. xxi, 40, 41, 42

Chafika Benzaid and Tarik Taleb. ZSM Security: Threat Surface and Best Practices. *IEEE Network*, 34(3):124–133, May 2020. ISSN 1558-156X. doi: 10.1109/MNET.001.1900273. Number: 3 Conference Name: IEEE Network. 25

Kamal Benzekki, Abdelhakim El Fergougui, and Abdellah Elbelrhiti Elalaoui. Software-defined networking (sdn): A survey. *Security and Communication Networks*, 2016, 2016. doi: 10.1002/sec.1737. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1737. 13

Badre Bousalem, Vinicius F. Silva, Rami Langar, and Sylvain Cherrier. Deep Learning-based Approach for DDoS Attacks Detection and Mitigation in 5G and Beyond Mobile Networks. In *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, pages 228–230, Milan, Italy, June 2022. IEEE. ISBN 978-1-66540-694-9. doi: 10.1109/NetSoft54395.2022.9844053. URL https://ieeexplore.ieee.org/document/9844053/.

Daniele Bringhenti, Jalolliddin Yusupov, Alejandro Molina Zarca, Fulvio Valenza, Riccardo Sisto, Jorge Bernal Bernabe, and Antonio Skarmeta. Automatic, verifiable and optimized policy-based security enforcement for SDN-aware IoT networks. *Computer Networks*, 213:109123, August 2022. ISSN 13891286. doi: 10.1016/j.comnet.2022.109123. URL https://linkinghub.elsevier.com/retrieve/pii/S1389128622002468.

John Browne, Emma Collins, Krzysztof Kepka, Sunku Ranganath, Jabir Kanhira Kadavathu, Swati Sehgal, Killian Muldoon, and Michal Kobylinski. Closed loop automation - telemetry aware scheduler for service healing and platform resilience, 2021. URL https://networkbuilders.intel.com/docs/networkbuilders/closed-loop-platform-automation-service-healing-and-platform-resilience.pdf. White Paper. 53

J. Burns, A. Cheng, P. Gurung, S. Rajagopalan, P. Rao, D. Rosenbluth, A.V. Surendran, and D.M. Martin. Automatic management of network security policy. In *Proceedings DARPA Information Survivability Conference and Exposition*

II. DISCEX'01, volume 2, pages 12–26, Anaheim, CA, USA, 2001. IEEE Comput. Soc. ISBN 978-0-7695-1212-9. doi: 10.1109/DISCEX.2001.932156. URL http://ieeexplore.ieee.org/document/932156/.

A. Campanella. Intent based Network Operations. In *Optical Fiber Communication Conference (OFC) 2019*, page W4G.3, San Diego, California, 2019. OSA. ISBN 978-1-943580-53-8. doi: 10.1364/OFC.2019.W4G.3. URL https://opg.optica.org/abstract.cfm?URI=OFC-2019-W4G.3.

R. Chadha, G. Lapiotis, and S. Wright. Guest editorial - policy-based networking. *IEEE Network*, 16(2):8–9, March 2002. ISSN 0890-8044. doi: 10.1109/MNET.2002.993217. URL http://ieeexplore.ieee.org/document/993217/.

Emmanuel Sibusiso Chaki and Mthulisi Velempini. Implementing a Machine Learning based Hybrid Model to Counter Attacks in Mobile Edge Computing. *International Conference on Artificial Intelligence and its Applications*, pages 59–64, December 2023. URL https://mauricon.org/conferences/index.php/icarti/article/view/42.

Geoffrey Chollon, Dhouha Ayed, Rodrigo Asensio Garriga, Alejandro Molina Zarca, Antonio Skarmeta, Maria Christopoulou, Wissem Soussi, Gürkan Gür, and Uwe Herzog. Etsi zsm driven security management in future networks. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 334–339, 2022a. doi: 10.1109/FNWF55208.2022.00065. xxi, 26, 27

Geoffrey Chollon, Dhouha Ayed, Rodrigo Asensio Garriga, Alejandro Molina Zarca, Antonio Skarmeta, Maria Christopoulou, Wissem Soussi, Gürkan Gür, and Uwe Herzog. ETSI ZSM Driven Security Management in Future Networks. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 334–339, October 2022b. doi: 10.1109/FNWF55208.2022.00065. ISSN: 2770-7679.

Ankur Chowdhary, Abdulhakim Sabur, Neha Vadnere, and Dijiang Huang. Intent-Driven Security Policy Management for Software-Defined Systems. *IEEE Transactions on Network and Service Management*, 19(4):5208–5223, December 2022a.

ISSN 1932-4537, 2373-7379. doi: 10.1109/TNSM.2022.3183591. URL https://ieeexplore.ieee.org/document/9797849/.

Ankur Chowdhary, Abdulhakim Sabur, Neha Vadnere, and Dijiang Huang. Intent-Driven Security Policy Management for Software-Defined Systems. *IEEE Transactions on Network and Service Management*, 19(4):5208–5223, December 2022b. ISSN 1932-4537, 2373-7379. doi: 10.1109/TNSM.2022.3183591. URL https://ieeexplore.ieee.org/document/9797849/.

Mostafa Zaman Chowdhury, Md. Shahjalal, Shakil Ahmed, and Yeong Min Jang. 6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions. *IEEE Open Journal of the Communications Society*, 1:957–975, 2020. ISSN 2644-125X. doi: 10.1109/OJCOMS.2020.3010270. URL https://ieeexplore.ieee.org/document/9144301/.

Edmund Clarke. Bounded Model Checking Using Satisfiability Solving.

A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura. Rfc 9315: Intent-based networking - concepts and definitions, 2022. 39

José Cunha, Pedro Ferreira, Eva M Castro, Paula Cristina Oliveira, Maria João Nicolau, Iván Núñez, Xosé Ramon Sousa, and Carlos Serôdio. Enhancing network slicing security: Machine learning, software-defined networking, and network functions virtualization-driven strategies. *Future Internet*, 16(7):226, 2024. doi: 10.3390/fi16070226. 17

Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani, and Ammar Rayes. Software-defined networking security: pros and cons. *IEEE Communications Magazine*, 53 (6):73–79, 2015. doi: 10.1109/MCOM.2015.7120048. 12, 14

Nicodemos Damianou, Naranker Dulay, Emil Lupu, and Morris Sloman. Ponder: A Language for Specifying Security and Management Policies for Distributed Systems. January 2000.

Gianluca Davoli, Walter Cerroni, Slavica Tomovic, Chiara Buratti, Chiara Contoli,

and Franco Callegati. Intent-based service management for heterogeneous software-defined infrastructure domains. *International Journal of Network Management*, 29 (1):e2051, 2019. 42

Zeinab El-Rewini, Karthikeyan Sadatsharan, Daisy Flora Selvaraj, Siby Jose Plathottam, and Prakash Ranganathan. Cybersecurity challenges in vehicular communications. *Vehicular Communications*, 23:100214, June 2020. ISSN 22142096. doi: 10.1016/j.vehcom.2019.100214. URL https://linkinghub.elsevier.com/retrieve/pii/S221420961930261X.

ETSI. *Zero-touch Network and Service Management (ZSM); Closed-Loop Automation; Part 2: Solutions for Automation of E2E Service and Network Management Use Cases*. ETSI, Sophia Antipolis, France, June 2022. URL https://www.etsi.org/standards. DGS/ZSM-009-2_CLA_sol. 31

ETSI. *Zero-touch Network and Service Management (ZSM); Closed-Loop Automation; Part 3: Advanced Topics*. ETSI, Sophia Antipolis, France, August 2023. URL https://www.etsi.org/standards-search. DGR/ZSM-009-3_Cla_AdvTop. 34

European Telecommunications Standards Institute. Zero-touch network and service management (zsm); reference architecture. ETSI Group Specification (GS) ZSM 002 V1.1.1, ETSI, August 2019. Accessed: 2023-07-03. xxi, 25

Matthias Falkner and John Apostolopoulos. Intent-based networking for the enterprise.

Nate Foster, Michael J. Freedman, Rob Harrison, Jennifer Rexford, Matthew L. Meola, and David Walker. Frenetic: a high-level language for OpenFlow networks. In *Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow*, pages 1–6, Philadelphia Pennsylvania, November 2010. ACM. ISBN 978-1-4503-0467-2. doi: 10.1145/1921151.1921160. URL https://dl.acm.org/doi/10.1145/1921151.1921160.

Futuriom. Csp network automation trends, 2020. URL https://www.nuagenetworks.net/wp-content/uploads/2020/08/Futuriom_CSP_Network_Automation_v2_6_final_Nokia.pdf. Premium Technology Research. 53

Michael Geller and Pramod Nair. 5g security innovation with cisco white paper, 2018. URL https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/service-provider-security-solutions/5g-security-innovation-with-cisco-wp.pdf. 51

K. Giotis, Y. Kryftis, and V. Maglaris. Policy-based orchestration of NFV services in Software-Defined Networks. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pages 1–5, London, United Kingdom, April 2015. IEEE. ISBN 978-1-4799-7899-1. doi: 10.1109/NETSOFT.2015.7116145. URL http://ieeexplore.ieee.org/document/7116145/.

Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97, February 2015. ISSN 0163-6804. doi: 10.1109/MCOM.2015.7045396. URL http://ieeexplore.ieee.org/document/7045396/.

João Henriques, Filipe Caldeira, Tiago Cruz, and Paulo Simões. An automated closed-loop framework to enforce security policies from anomaly detection. *Computers Security*, 123:102949, 2022. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2022.102949. URL https://www.sciencedirect.com/science/article/pii/S0167404822003418. 35

Suwani Jayasinghe, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, and Mika Ylianttila. Federated learning based anomaly detection as an enabler for securing network and service management automation in beyond 5g networks. 06 2022. doi: 10.1109/EuCNC/6GSummit54941.2022.9815754. 28

E. Kapassa, M. Touloupou, et al. An innovative ehealth system powered by 5g network slicing. *IEEE Internet of Things Journal*, 2019, 2019. doi: 10.1109/JIOT.2019.2963489. URL https://ieeexplore.ieee.org/abstract/document/8939266/. 17

H. Khan, P. Luoto, M. Bennis, et al. On the application of network slicing for 5g-v2x. In *European Wireless 2018*. IEEE, 2018. doi: 10.1109/EW.2018.8385521. URL https://ieeexplore.ieee.org/abstract/document/8385521/. 17

Talha Ahmed Khan, Khizar Abbas, Afaq Muhammad, Adeel Rafiq, and Wang-Cheol Song. GAN and DRL Based Intent Translation and Deep Fake Configuration Generation for Optimization. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 347–352, Jeju, Korea (South), October 2020. IEEE. ISBN 978-1-72816-758-9. doi: 10.1109/ICTC49870.2020.9289564. URL https://ieeexplore.ieee.org/document/9289564/.

Kenneth Kimani, Vitalice Oduol, and Kibet Langat. Cyber security challenges for IoT-based smart grid networks. *International Journal of Critical Infrastructure Protection*, 25:36–49, June 2019. ISSN 18745482. doi: 10.1016/j.ijcip.2019.01.001. URL https://linkinghub.elsevier.com/retrieve/pii/S1874548217301622.

Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015. doi: 10.1109/JPROC.2014.2371999. 11

Abdelquoddouss Laghrissi and Tarik Taleb. A survey on the placement of virtual resources and virtual network functions. *IEEE Communications Surveys & Tutorials*, 21(2):1409–1426, 2019. doi: 10.1109/COMST.2018.2884835. 16

Adrian Lara and Byrav Ramamurthy. OpenSec: Policy-Based Security Using Software-Defined Networking. *IEEE Transactions on Network and Service Management*, 13 (1):30–42, March 2016. ISSN 1932-4537. doi: 10.1109/TNSM.2016.2517407. URL http://ieeexplore.ieee.org/document/7378982/.

Woosik Lee and Namgi Kim. Security Policy Scheme for an Efficient Security Architecture in Software-Defined Networking. *Information*, 8(2):65, June 2017. ISSN 2078-2489. doi: 10.3390/info8020065. URL http://www.mdpi.com/2078-2489/8/2/65.

Aris Leivadeas and Matthias Falkner. A Survey on Intent-Based Networking. *IEEE Communications Surveys & Tutorials*, 25(1):625–655, 2023. ISSN 1553-877X, 2373-745X. doi: 10.1109/COMST.2022.3215919. URL https://ieeexplore.ieee.org/document/9925251/.

Qianmu Li, Youhui Tian, Qiang Wu, Qi Cao, Haiyuan Shen, and Huaqiu Long. A Cloud-Fog-Edge Closed-Loop Feedback Security Risk Prediction Method. *IEEE Access*, 8:29004–29020, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2972032. URL https://ieeexplore.ieee.org/document/8985306/.

Madhusanka Liyanage, Quoc-Viet Pham, Kapal Dev, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, and Gokul Yenduri. A survey on Zero touch network and Service Management (ZSM) for 5G and beyond networks. *Journal of Network and Computer Applications*, 203:103362, July 2022. ISSN 10848045. doi: 10.1016/j.jnca.2022.103362. URL https://linkinghub.elsevier.com/retrieve/pii/S1084804522000297.

Jun Sheng Low. *Intent-based networking: policy to solutions recommendations*. PhD thesis, UTAR, 2020. 39

Leonidas Lymberopoulos, Emil Lupu, and Morris Sloman. [No title found]. *Journal of Network and Systems Management*, 11(3):277–303, 2003. ISSN 10647570. doi: 10.1023/A:1025719407427. URL http://link.springer.com/10.1023/A:1025719407427.

Hocine Mahtout, Mariam Kiran, Anu Mercian, and Bashir Mohammed. 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*, pages 27–30, Stockholm Sweden, June 2020. ACM. ISBN 978-1-4503-7980-9. doi: 10.1145/3391812.3396269. URL https://dl.acm.org/doi/10.1145/3391812.3396269.

Carlos Manso, Pol Alemany, Ricard Vilalta, Raul Muñoz, Ramon Casellas, and Ricardo Martínez. End-to-End SDN/NFV Orchestration of Multi-Domain Transport Networks and Distributed Computing Infrastructure for Beyond-5G Services. *IEICE Transactions on Communications*, E104.B(3):188–198, March 2021a. ISSN 0916-8516, 1745-1345. doi: 10.1587/transcom.2020NVI0001. URL https://www.jstage.jst.go.jp/article/transcom/E104.B/3/E104.B_2020NVI0001/_article.

Carlos Manso, Pol Alemany, Ricard Vilalta, Raul Muñoz, Ramon Casellas, and Ricardo Martínez. End-to-End SDN/NFV Orchestration of Multi-Domain Transport Networks and Distributed Computing Infrastructure for Beyond-5G Services. *IEICE Transactions on Communications*, E104.B(3):188–198, March 2021b. ISSN 0916-8516, 1745-1345. doi: 10.1587/transcom.2020NVI0001. URL https://www.jstage.jst.go.jp/article/transcom/E104.B/3/E104.B_2020NVI0001/_article.

Yashar Mansouri and M. Ali Babar. A review of edge computing: Features and resource virtualization. *Information Systems*, 92:101582, 2020. 22

Anu Mercian, Faraz Ahmed, Puneet Sharma, Shaun Wackerly, and Charles Clark. Mind the semantic gap: Policy intent inference from network metadata. In *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, pages 312–320, 2021. doi: 10.1109/NetSoft51509.2021.9492552. xxii, 44, 45, 46

Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, 18(1):236–262, 2016. doi: 10.1109/COMST.2015.2477041. xxi, 14, 15

Alejandro Molina Zarca, Miloud Bagaa, Jorge Bernal Bernabe, Tarik Taleb, and Antonio F. Skarmeta. Semantic-Aware Security Orchestration in SDN/NFV-Enabled IoT Systems. *Sensors*, 20(13):3622, June 2020. ISSN 1424-8220. doi: 10.3390/s20133622. URL https://www.mdpi.com/1424-8220/20/13/3622.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. Textual keyword extraction and summarization: State-of-the-art. *Information Processing & Management*, 56(6):102088, November 2019. ISSN 03064573. doi: 10.1016/j.ipm.2019.102088. URL https://linkinghub.elsevier.com/retrieve/pii/S0306457319300044.

Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Communications Surveys & Tutorials*, 21, April 2019. doi: 10.1109/COMST.2019.2910750.

Dinh C. Nguyen, Pubudu N. Pathirana, Ming Ding, and Aruna Seneviratne. Blockchain for 5G and beyond networks: A state of the art survey. *Journal of Network and Computer Applications*, 166:102693, September 2020. ISSN 10848045. doi: 10.1016/j.jnca.2020.102693. URL https://linkinghub.elsevier.com/retrieve/pii/S1084804520301673.

W Niewolski, TW Nowak, and M Sepczuk. Security architecture for authorized anonymous communication in 5g mec. *IEEE Access*, 2020a. URL https://ieeexplore.ieee.org/document/9127819.

W Niewolski, TW Nowak, M Sepczuk, and Z Kotulski. Security context migration in mec: Challenges and use cases. *IEEE Communications Magazine*, 2020b. URL https://ieeexplore.ieee.org/document/9364272.

Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *IEEE Communications Surveys & Tutorials*, 16(3):1617–1634, 2014. ISSN 1553-877X. doi: 10.1109/SURV.2014.012214.00180. URL http://ieeexplore.ieee.org/document/6739370/.

omputer Engineering, Jeju, South Korea, Adeel Rafiq, Asif Mehmood, and Wang-Cheol Song. Intent-Based Slicing between Containers in SDN Overlay Network. *Journal of Communications*, pages 237–244, 2020. ISSN 23744367. doi: 10.12720/jcm.15.3.237-244. URL http://www.jocm.us/show-237-1517-1.html.

Ying Ouyang, Chungang Yang, Yanbo Song, Xinru Mi, and Mohsen Guizani. A brief survey and implementation on refinement for intent-driven networking. *IEEE Network*, 35(6):75–83, 2021. doi: 10.1109/MNET.001.2100194. 37

Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, 20(4):2961–2991, 2018. 22

Adeel Rafiq, Muhammad Afaq, and Wang-Cheol Song. Intent-based networking with proactive load distribution in data center using IBN manager and Smart Path manager. *Journal of Ambient Intelligence and Humanized Computing*, 11(11):4855–4872,

November 2020a. ISSN 1868-5137, 1868-5145. doi: 10.1007/s12652-020-01753-1. URL http://link.springer.com/10.1007/s12652-020-01753-1.

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):2782, April 2020b. ISSN 2071-1050. doi: 10.3390/su12072782. URL https://www.mdpi.com/2071-1050/12/7/2782.

Wajid Rafique, Lianyong Qi, Ibrar Yaqoob, Muhammad Imran, Raihan Ur Rasool, and Wanchun Dou. Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 22(3):1761–1804, 2020. ISSN 1553-877X, 2373-745X. doi: 10.1109/COMST.2020.2997475. URL https://ieeexplore.ieee.org/document/9099866/. 12

Deepak Singh Rana, Shiv Ashish Dhondiyal, and Sushil Kumar Chamoli. Software Defined Networking (SDN) Challenges, issues and Solution. *International Journal of Computer Sciences and Engineering*, 7(1):884–889, January 2019. ISSN 23472693. doi: 10.26438/ijcse/v7i1.884889. URL http://www.ijcseonline.org/full_paper_view.php?paper_id=3602.

Pasika Ranaweera, Anca Delia Jurcut, and Madhusanka Liyanage. Realizing multi-access edge computing feasibility: Security perspective. *IEEE Conference on Standards for Communications and Networking*, 2019. URL https://ieeexplore.ieee.org/document/8931357/.

Ana Carolina Riekstin, Guilherme Carvalho Januario, Bruno Bastos Rodrigues, Viviane Tavares Nascimento, Tereza Cristina Melo De Brito Carvalho, and Catalin Meirosu. A Survey of Policy Refinement Methods as a Support for Sustainable Networks. *IEEE Communications Surveys & Tutorials*, 18(1):222–235, 2016. ISSN 1553-877X. doi: 10.1109/COMST.2015.2463811. URL http://ieeexplore.ieee.org/document/7174949/.

Sergio Rivera, James Griffioen, Zongming Fei, and Jane Huffman Hayes. Expressing and Managing Network Policies for Emerging HPC Systems. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*, pages 1–7, Chicago IL USA, July 2019. ACM. ISBN 978-1-4503-7227-5. doi: 10.1145/3332186.3333045. URL https://dl.acm.org/doi/10.1145/3332186.3333045.

Robert W. Rycroft and Don E. Kash. Self-organizing innovation networks: implications for globalization. *Technovation*, 24(3):187–197, March 2004. ISSN 01664972. doi: 10.1016/S0166-4972(03)00092-0. URL https://linkinghub.elsevier.com/retrieve/pii/S0166497203000920.

D Sabella, A Reznik, KR Nayak, D Lopez, and F Li. Mec security: Status of standards support and future evolutions. Technical report, ETSI White Paper, 2021. URL https://www.etsi.org/deliver/etsi_wp/00401_00499/00462/01.01.01_60/wp_00462v010101p.pdf.

Fady Samuel, Mosharaf Chowdhury, and Raouf Boutaba. PolyViNE: policy-based virtual network embedding across multiple domains. 2013.

Eder J. Scheid, Cristian C. Machado, Muriel F. Franco, Ricardo L. Dos Santos, Ricardo P. Pfitscher, Alberto E. Schaeffer-Filho, and Lisandro Z. Granville. INSpIRE: Integrated NFV-based Intent Refinement Environment. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 186–194, Lisbon, Portugal, May 2017. IEEE. ISBN 978-3-901882-89-0. doi: 10.23919/INM.2017.7987279. URL http://ieeexplore.ieee.org/document/7987279/.

Fred B Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1).

N Navya Shree, Keertana Kamesh, Sanchita Vishwa, Radhika Ravikumar, Rajeshwari Hegde, and Sharath Kumar. Security challenges in mobile communication networks. In *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pages 82–86, 2019. doi: 10.1109/ICISC44355.2019.9036385. 51

Riccardo Sisto, Fulvio Valenza, and Amoroso. Automated Policy Enforcement in Software Defined Networking and Network Function Virtualization Environment.

Theresa Sobb, Benjamin Turnbull, and Nour Moustafa. Supply Chain 4.0: A Survey of Cyber Security Challenges, Solutions and Future Directions. *Electronics*, 9(11):1864, November 2020. ISSN 2079-9292. doi: 10.3390/electronics9111864. URL https://www.mdpi.com/2079-9292/9/11/1864. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

Nathan Franklin Saraiva De Sousa and Christian Esteve Rothenberg. CLARA: Closed Loop-based Zero-touch Network Management Framework. In *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 110–115, Heraklion, Greece, November 2021. IEEE. ISBN 978-1-66543-983-1. doi: 10.1109/NFV-SDN53031.2021.9665048. URL https://ieeexplore.ieee.org/document/9665048/. xxi, 36, 37

Prashant Subedi, Abeer Alsadoon, P. W. C. Prasad, Sabih Rehman, Nabil Giweli, Muhammad Imran, and Samrah Arif. Network slicing: A next generation 5g perspective. *Journal of Wireless Communications and Network*, 2021:102, 2021. doi: 10.1186/s13638-021-01983-7. URL https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-021-01983-7. 17

Ali Sufyan, Khan Bahadar Khan, Osama A. Khashan, Talha Mir, and Usama Mir. From 5G to beyond 5G: A Comprehensive Survey of Wireless Network Evolution, Challenges, and Promising Technologies. *Electronics*, 12(10):2200, May 2023. ISSN 2079-9292. doi: 10.3390/electronics12102200. URL https://www.mdpi.com/2079-9292/12/10/2200. 10

Pigilam Swetha, Monicka Vijayakumar, Jahfer Abdulazeez, and Prem Kumar. Closed loop engine for network automation. xxi, 34

Tarik Taleb, Ibrahim Afolabi, Konstantinos Samdanis, and Faqir Zarrar Yousaf. On multi-domain network slicing orchestration architecture & federated resource control. *Journal of Network and Systems Management*, 28(4):882–922, 2020.

doi: 10.1007/s10922-020-09564-7. URL https://link.springer.com/article/10.1007/s10922-020-09564-7. xxi, 17, 18

Anurag Thantharate, Rahul Paropkari, Vijay Walunj, Cory Beard, and Poonam Kankariya. *Secure5G: A Deep Learning Framework Towards a Secure Network Slicing in 5G and Beyond*. January 2020. doi: 10.1109/CCWC47524.2020.9031158. Pages: 0857.

Vasileios Theodorou, Alexios Lekidis, Theodoros Bozios, Kalman Meth, Adriana Fernández-Fernández, James Tavlor, Pedro Diogo, Pedro Martins, and Rasoul Behravesh. Blockchain-based Zero Touch Service Assurance in Cross-domain Network Slicing. In *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pages 395–400, June 2021. doi: 10.1109/EuCNC/6GSummit51104.2021.9482602. ISSN: 2575-4912.

Bingchuan Tian, Xinyi Zhang, Ennan Zhai, Hongqiang Harry Liu, Qiaobo Ye, Chunsheng Wang, Xin Wu, Zhiming Ji, Yihong Sang, Ming Zhang, Da Yu, Chen Tian, Haitao Zheng, and Ben Y. Zhao. Safely and automatically updating in-network ACL configurations with intent language. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 214–226, Beijing China, August 2019. ACM. ISBN 978-1-4503-5956-6. doi: 10.1145/3341302.3342088. URL https://dl.acm.org/doi/10.1145/3341302.3342088.

C. Tsorouchis, S. Denazis, C. Kitchara, J. Vivero, E. Salamanca, E. Magana, A. Galis, J.L. Manas, Y. Corlinet, B. Mathieu, and O. Koufopavlou. A policy-based management architecture for active and programmable networks. *IEEE Network*, 17 (3):22–28, May 2003. ISSN 0890-8044. doi: 10.1109/MNET.2003.1201473. URL http://ieeexplore.ieee.org/document/1201473/.

Ali Tufail, Abdallah Namoun, Ahmed Alrehaili, and Arshad Ali. A survey on 5g enabled multi-access edge computing for smart cities: Issues and future prospects. *International Journal of Computer Science and Network Security*, 21(6), 2021. URL https://www.researchgate.net/publication/353039842.

Daphne Tuncer, Marinos Charalambides, Gioacchino Tangari, and George Pavlou. A northbound interface for software-based networks. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 99–107, 2018. 11

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, April 2019a. ISSN 1556-6013, 1556-6021. doi: 10.1109/TIFS.2018.2868220. URL https://ieeexplore.ieee.org/document/8453023/.

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, April 2019b. ISSN 1556-6013, 1556-6021. doi: 10.1109/TIFS.2018.2868220. URL https://ieeexplore.ieee.org/document/8453023/.

Xiang Wang, Weiqi Shi, Yang Xiang, and Jun Li. Efficient Network Security Policy Enforcement With Policy Space Analysis. *IEEE/ACM Transactions on Networking*, 24(5):2926–2938, October 2016. ISSN 1063-6692, 1558-2566. doi: 10.1109/TNET.2015.2502402. URL http://ieeexplore.ieee.org/document/7362256/.

Yong Wang, Aiqing Zhang, Peiyun Zhang, and Huaqun Wang. Cloud-Assisted EHR Sharing With Security and Privacy Preservation via Consortium Blockchain. *IEEE Access*, 7:136704–136719, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2943153. URL https://ieeexplore.ieee.org/document/8846684/.

Yiming Wei, Mugen Peng, and Yaqiong Liu. Intent-based networks for 6g: Insights and challenges. *Digital Communications and Networks*, 6(3):270–280, 2020a. ISSN 2352-8648. doi: https://doi.org/10.1016/j.dcan.2020.07.001. URL https://www.sciencedirect.com/science/article/pii/S2352864820302418. 38

Yiming Wei, Mugen Peng, and Yaqiong Liu. Intent-based networks for 6G: Insights and challenges. *Digital Communications and Networks*, 6(3):270–280, August 2020b. ISSN 23528648. doi: 10.1016/j.dcan.2020.07.001. URL https://linkinghub.elsevier.com/retrieve/pii/S2352864820302418.

Wen Wu, Conghao Zhou, Mushu Li, Huaqing Wu, Haibo Zhou, Ning Zhang, Xuemin Sherman Shen, and Weihua Zhuang. AI-Native Network Slicing for 6G Networks. *IEEE Wireless Communications*, 29(1):96–103, February 2022. ISSN 1558-0687. doi: 10.1109/MWC.001.2100338. URL https://ieeexplore.ieee.org/abstract/document/9749222?casa_token=QGQEb4ywy7IAAAAA:pxBeUYR0qmuRyGtbfNQ64w3oh7eFxgFjWp6HCAkoG6WTcHi5qr8O05dAFP2fNtv-hrnKmKYgRA. Conference Name: IEEE Wireless Communications.

Michael Xevgenis, Dimitrios G. Kogias, Panagiotis A. Karkazis, and Helen C. Leligou. Addressing zsm security issues with blockchain technology. *Future Internet*, 15(4): 129, 2023a. doi: 10.3390/fi15040129. URL https://www.mdpi.com/1999-5903/15/4/129. 28

Michael Xevgenis, Dimitrios G. Kogias, Panagiotis A. Karkazis, and Helen C. Leligou. Addressing ZSM Security Issues with Blockchain Technology. *Future Internet*, 15 (4):129, April 2023b. ISSN 1999-5903. doi: 10.3390/fi15040129. URL https://www.mdpi.com/1999-5903/15/4/129. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. A survey on software-defined networking. *IEEE Communications Surveys Tutorials*, 17(1):27–51, 2015. doi: 10.1109/COMST.2014.2330903. 11

Junfeng Xie, F. Richard Yu, Tao Huang, Renchao Xie, Jiang Liu, Chenmeng Wang, and Yunjie Liu. A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges. *IEEE Communications Surveys & Tutorials*, 21(1):393–430, 2019. ISSN 1553-877X, 2373-745X. doi: 10.1109/COMST.2018.2866942. URL https://ieeexplore.ieee.org/document/8444669/. xxi, 12

Yang Yang, Mulei Ma, Hequan Wu, Quan Yu, Ping Zhang, Xiaohu You, Jianjun Wu, Chenghui Peng, Tak-Shing Peter Yum, Sherman Shen, A. Hamid Aghvami, Geoffrey Y Li, Jiangzhou Wang, Guangyi Liu, Peng Gao, Xiongyan Tang, Chang Cao, John Thompson, Kat-Kit Wong, Shanzhi Chen, Merouane Debbah, Schahram

Dustdar, Frank Eliassen, Tao Chen, Xiangyang Duan, Shaohui Sun, Xiaofeng Tao, Qinyu Zhang, Jianwei Huang, Shuguang Cui, Wenjun Zhang, Jie Li, Yue Gao, Honggang Zhang, Xu Chen, Xiaohu Ge, Yong Xiao, Cheng-Xiang Wang, Zaichen Zhang, Song Ci, Guoqiang Mao, Changle Li, Ziyu Shao, Yong Zhou, Junrui Liang, Kai Li, Liantao Wu, Fanglei Sun, Kunlun Wang, Zening Liu, Kun Yang, Jun Wang, Teng Gao, and Hongfeng Shu. 6G Network AI Architecture for Everyone-Centric Customized Services. *IEEE Network*, pages 1–10, 2022a. ISSN 0890-8044, 1558-156X. doi: 10.1109/MNET.124.2200241. URL https://ieeexplore.ieee.org/document/9839652/.

Yang Yang, Mulei Ma, Hequan Wu, Quan Yu, Ping Zhang, Xiaohu You, Jianjun Wu, Chenghui Peng, Tak-Shing Peter Yum, Sherman Shen, A. Hamid Aghvami, Geoffrey Y Li, Jiangzhou Wang, Guangyi Liu, Peng Gao, Xiongyan Tang, Chang Cao, John Thompson, Kat-Kit Wong, Shanzhi Chen, Merouane Debbah, Schahram Dustdar, Frank Eliassen, Tao Chen, Xiangyang Duan, Shaohui Sun, Xiaofeng Tao, Qinyu Zhang, Jianwei Huang, Shuguang Cui, Wenjun Zhang, Jie Li, Yue Gao, Honggang Zhang, Xu Chen, Xiaohu Ge, Yong Xiao, Cheng-Xiang Wang, Zaichen Zhang, Song Ci, Guoqiang Mao, Changle Li, Ziyu Shao, Yong Zhou, Junrui Liang, Kai Li, Liantao Wu, Fanglei Sun, Kunlun Wang, Zening Liu, Kun Yang, Jun Wang, Teng Gao, and Hongfeng Shu. 6G Network AI Architecture for Everyone-Centric Customized Services. *IEEE Network*, pages 1–10, 2022b. ISSN 0890-8044, 1558-156X. doi: 10.1109/MNET.124.2200241. URL https://ieeexplore.ieee.org/document/9839652/.

Yang Yang, Mulei Ma, Hequan Wu, Quan Yu, Ping Zhang, Xiaohu You, Jianjun Wu, Chenghui Peng, Tak-Shing Peter Yum, Sherman Shen, A. Hamid Aghvami, Geoffrey Y Li, Jiangzhou Wang, Guangyi Liu, Peng Gao, Xiongyan Tang, Chang Cao, John Thompson, Kat-Kit Wong, Shanzhi Chen, Merouane Debbah, Schahram Dustdar, Frank Eliassen, Tao Chen, Xiangyang Duan, Shaohui Sun, Xiaofeng Tao, Qinyu Zhang, Jianwei Huang, Shuguang Cui, Wenjun Zhang, Jie Li, Yue Gao, Honggang Zhang, Xu Chen, Xiaohu Ge, Yong Xiao, Cheng-Xiang Wang, Zaichen Zhang, Song Ci, Guoqiang Mao, Changle Li, Ziyu Shao, Yong Zhou, Junrui Liang, Kai Li, Liantao Wu, Fanglei Sun, Kunlun Wang, Zening Liu, Kun Yang, Jun Wang,

Teng Gao, and Hongfeng Shu. 6G Network AI Architecture for Everyone-Centric Customized Services. *IEEE Network*, pages 1–10, 2022c. ISSN 0890-8044, 1558-156X. doi: 10.1109/MNET.124.2200241. URL https://ieeexplore.ieee.org/document/9839652/.

Bo Yi, Xingwei Wang, Keqin Li, Sajal K. Das, and Min Huang. A comprehensive survey of network function virtualization. *Computer Networks*, 133:212–262, 2018a. doi: 10.1016/j.comnet.2018.01.021. URL https://doi.org/10.1016/j.comnet.2018.01.021. 15

Bo Yi, Xingwei Wang, Keqin Li, Sajal K. Das, and Min Huang. A comprehensive survey of Network Function Virtualization. *Computer Networks*, 133:212–262, March 2018b. ISSN 13891286. doi: 10.1016/j.comnet.2018.01.021. URL https://linkinghub.elsevier.com/retrieve/pii/S1389128618300306.

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. doi: 10.1109/VTC2020-Spring48590.2020.9128422. 39

Shunliang Zhang. An overview of network slicing for 5g. *IEEE Wireless Communications*, June 2019. doi: 10.1109/MWC.2019.1800234. URL https://ieeexplore.ieee.org/document/8685766. xxi, 16, 19

F. Zhou, P. Yu, L. Feng, X. Qiu, and Z. Wang. Automatic network slicing for iot in smart city. *IEEE Wireless Communications*, 2020, 2020. doi: 10.1109/MWC.009.2000078. URL https://ieeexplore.ieee.org/abstract/document/9232929/. 17

Appendix

A.1 Included Documents

future internet

MDPI

Article

Enhancing Network Slicing Security: Machine Learning, Software-Defined Networking, and Network Functions Virtualization-Driven Strategies

José Cunha [1,2,*], Pedro Ferreira [1,2], Eva M. Castro [2,3,4], Paula Cristina Oliveira [1,5], Maria João Nicolau [3,4], Iván Núñez [2], Xosé Ramon Sousa [2] and Carlos Serôdio [1,3,*]

[1] Department of Engineering, School of Sciences and Technology, Universidade de Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal; pvieira@optaresolutions.com (P.F.); pcoliveira@utad.pt (P.C.O.)
[2] Optare Solutions, Parque Tecnológico de Vigo, 35315 Vigo, Spain; mpires@optaresolutions.com (E.M.C.); inunez@optaresolutions.com (I.N.); xrsousa@optaresolutions.com (X.R.S.)
[3] Algoritmi Center, University of Minho, 4710-057 Braga, Portugal; joao@dsi.uminho.pt
[4] Department of Information Systems, School of Engineering, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
[5] Centre for the Research and Technology of Agro-Environmental and Biological Sciences (CITAB), Universidade de Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
* Correspondence: jcunha@optaresolutions.com (J.C.); cserodio@utad.pt (C.S.)

Abstract: The rapid development of 5G networks and the anticipation of 6G technologies have ushered in an era of highly customizable network environments facilitated by the innovative concept of network slicing. This technology allows the creation of multiple virtual networks on the same physical infrastructure, each optimized for specific service requirements. Despite its numerous benefits, network slicing introduces significant security vulnerabilities that must be addressed to prevent exploitation by increasingly sophisticated cyber threats. This review explores the application of cutting-edge technologies—Artificial Intelligence (AI), specifically Machine Learning (ML), Software-Defined Networking (SDN), and Network Functions Virtualization (NFV)—in crafting advanced security solutions tailored for network slicing. AI's predictive threat detection and automated response capabilities are analysed, highlighting its role in maintaining service integrity and resilience. Meanwhile, SDN and NFV are scrutinized for their ability to enforce flexible security policies and manage network functionalities dynamically, thereby enhancing the adaptability of security measures to meet evolving network demands. Thoroughly examining the current literature and industry practices, this paper identifies critical research gaps in security frameworks and proposes innovative solutions. We advocate for a holistic security strategy integrating ML, SDN, and NFV to enhance data confidentiality, integrity, and availability across network slices. The paper concludes with future research directions to develop robust, scalable, and efficient security frameworks capable of supporting the safe deployment of network slicing in next-generation networks.

Keywords: network security; SDN; NFV; ML; network slicing

Citation: Cunha, J.; Ferreira, P.; Castro, E.M.; Oliveira, P.C.; Nicolau, M.J.; Núñez, I.; Sousa, X.R.; Serôdio, C. Enhancing Network Slicing Security: Machine Learning, Software-Defined Networking, and Network Functions Virtualization-Driven Strategies. *Future Internet* **2024**, *16*, 226. https://doi.org/10.3390/fi16070226

Academic Editor: Paolo Bellavista

Received: 7 May 2024
Revised: 14 June 2024
Accepted: 24 June 2024
Published: 27 June 2024

1. Introduction

The ongoing rollout of 5G networks and the anticipatory designs of 6G infrastructures represent monumental leaps in telecommunications technology. These advances herald a new era characterised by unprecedented data speeds, massive connectivity, and highly customizable network environments. Central to these innovations is network slicing, a transformative approach that allows multiple virtual networks to operate on the same physical hardware, each tailored to meet specific service requirements.

Network operators are beginning to adopt advanced 5G technologies, including the Stand-Alone (SA) version, which boasts enhanced features [1]. The SA version is a fully independent 5G network that operates without relying on existing 4G LTE infrastructure.

Security and Trust in Open and Disaggregated 6G networks

P.Alemany*, R.Muñoz*, R.Vilalta*, Ll.Gifre*, R.Martínez *, R.Casellas* M.Castro†, P.Ferreira†,
D.Moreira†, J.García†, J.Cunha†, I.Núñez†, G.Gómez‡, S.Castro‡, A.Pastor§, D.López§
* Centre Tecnològic de Telecomunicacions de Catalunya (CTTC-CERCA), Spain
† Optare Solutions, Spain
‡ EVIDEN, Spain
§ Telefónica Innovación Digital, Spain

2024 24th International Conference on Transparent Optical Networks (ICTON) | 979-8-3503-7732-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICTON62926.2024.10647935

ABSTRACT Telecommunication networks are undergoing a significant shift from closed and proprietary systems towards open and interoperable networks. This transition allows for greater flexibility and reduced dependence on single providers. However, this openness also raises substantial security and trust issues, necessitating new approaches like the use of an intelligent and autonomous 6G network slicing security manager to manage security in multi-provider environments. Additionally, trust management is becoming increasingly crucial, with technologies such as Blockchain proposed to ensure the reliability and integrity of operations across this new, decentralized landscape. This model supports a dynamic marketplace where providers can freely negotiate and manage resources, thereby enhancing service delivery and network management.

Keywords 6G networks, multi-provider, network slices, security management, trust management, blockchain.

1. Introduction

Traditionally, network equipment has relied on closed software and hardware systems, integrated and patented by a few providers, creating well-known provider islands in telecom networks. In the last years, there has been a trend toward replacing these closed systems across various network segments (RAN, aggregation, transport, and core) with open and interoperable multi-vendor systems [1]. Industry-led initiatives like the O-RAN Alliance (ORAN) and the Telecom Infra Project (TIP) are promoting this change, which enhances competition, fosters innovation, and supports a more cost-effective and competitive deployment of technologies such as 6G. However, open telecom networks pose significant questions regarding security and trust in this complex multi-provider environment.

Network operators relies on proprietary solutions from closed system providers for network security. However, the emergence of open and disaggregated 6G networks provides a unique opportunity for operators to manage their network security more directly and flexibly using open technologies. The introduction of a 6G network slicing security manager (NSSM) is key for managing security requirements across multi-provider networks [2]. This manager will enable the definition of service level agreements (SLAs) with specific security requirements, ensuring that network slices not only meet technical and quality of service needs but also adhere to stringent security standards. The NSSM will use a range of security policies and tools, including monitoring probes and closed-control loops, to maintain security compliance and reactively address potential security threats.

Trust management is becoming increasingly critical in the transition to open and disaggregated 6G networks, where traditional, reputation-based methods are insufficient. In such networks, the complexity of multi-provider scenarios requires a robust system for measuring and evaluating trust, with Blockchain technology identified as a key solution [3]. Blockchain's attributes—decentralization, immutability, transparency, and verifiability—make it ideal for establishing a new foundation of trust. By verifying the actions and responsibilities of various providers, Blockchain allows for a more dynamic and transparent approach to network management. It replaces the conventional centralized model with a distributed framework where infrastructure and network services can be negotiated and managed in real time, enhancing the overall integrity and efficiency of 6G networks. The trust manager is the key element to compute a set of reputation and trust parameters (based on a set of per domain metrics) and distribute them transparently across the whole system and keeping an immutable history of how trustworthy each provider is to fulfil an operator's request.

2. Network Slicing Security Manager overview

The NSSM is designed with several key components to ensure robust security management across network slices. These include:

- Security SLA & Policies (SSLA&P): This component is responsible for designing and managing the data objects that encapsulate the security service level agreements (SSLA) and associated policies. These policies govern how services are configured initially or adjusted in response to threats. SSLA&P operates at both the end-to-end (E2E) and domain-specific levels.
- Security Closed-loop (CL) Automation: This component processes monitoring data to evaluate security threats by comparing observed events against predefined SSLA thresholds. If a violation is detected, it triggers policies designed to mitigate the threat.

979-8-3503-7732-3/24/$31.00 ©2024 IEEE