Secure deployment of third-party applications over 5G-NFV ML-empowered infrastructures

Ana Hermosilla[1], Jose Manuel Manjón-Cáliz[2], Pedro Martinez-Julia[3], Antonio Pastor[2], Jordi Ortiz[4], Diego R. Lopez[2], and Antonio Skarmeta[1]

[1] University of Murcia, Murcia, Spain
{ana.hermosilla, skarmeta}@um.es
[2] Telef́onica I+D, Madrid, Spain
{josemanuel.manjon, antonio.pastorperales, diego.r.lopez@}telefonica.com
[3] National Institute of Information and Communications Technology, Tokyo, Japan
pedro@nict.go.jp
[4] University Center of Defense at the Spanish Air Force Academy, San Javier, Spain
jordi.ortiz@cud.upct.es

Abstract

In recent decades, mobile technology has undergone significant advancements, transforming our digital landscape. However, these innovations have ushered in a new era of security challenges. The emergence of 5G and beyond networks presents unprecedented opportunities for connectivity and innovation but also introduces unique security considerations, notably with Network Function Virtualization (NFV). However, the shared infrastructure and third-party access expose vulnerabilities that demand attention. This paper proposes a multifaceted solution to address these challenges, also introducing a sandbox environment for preliminary application testing and monitoring to detect anomalies and threats, bolstered by Machine Learning (ML) and Artificial Intelligence (AI). Additionally, comprehensive application testing within the sandbox evaluates functionality and security. In essence, this research offers a comprehensive framework to safeguard the integrity and security of applications within the dynamic landscape of 5G and beyond networks, ensuring a balance between technological advancement and security imperatives.

Keywords: NFV, 5G, Machine Learning, Security

1 Introduction

Over the past few decades, mobile technologies have undergone a remarkable and transformative journey, culminating in relevant technological advancements. These advancements have not only revolutionized the way we interact with the digital world but have also brought forth a host of security challenges that warrant our attention.

The advent of 5G and beyond networks represents a significant leap forward in the realm of telecommunications, ushering in a new era of connectivity and possibilities. However, and as said, along with these advancements come a distinct set of technological features, each with its own unique security considerations. One prominent feature in this landscape is Network Function Virtualization (NFV), which, while offering remarkable flexibility and dynamism, also introduces a fresh set of security challenges

The first notable security concern stems from the increased complexity of the infrastructure. With NFV, networks' functions are virtualized and can run on shared hardware. While this enhances flexibility and resource utilization, reducing CAPEX and OPEX, it also means that the

The 7th International Conference on Mobile Internet Security (MobiSec'23), Dec 19-21, 2023, Okinawa, Japan, Article No.50

infrastructure becomes more intricate, usually correlating with an elevated risk of vulnerabilities and potential threats. The greater the number of components in the system, the greater the potential attack surface.

Furthermore, NFV may introduce an additional layer of risk by enabling third-party access to the shared infrastructure. In traditional network architectures, a provider's infrastructure remained isolated from external entities. However, with NFV, this infrastructure can be made available to third parties. The coexistence of various applications on the same infrastructure raises concerns about data isolation and protection. Vulnerabilities in one application could potentially impact others, leading to unintended consequences and security breaches.

Thus, ensuring the correctness of the applications that are going to be deployed over the infrastructure becomes critical, as well as monitoring their behavior once deployed to detect potential malfunctions or attacks. In that sense, in this paper, a series of mechanisms are proposed: in the shape of a sandbox where running and monitoring the applications before their instantiation in a real production infrastructure; a system capable of monitoring and analyzing the current status of the sandbox's networks, powered by ML/IA mechanisms, to detect possible malfunctions or attacks; and a framework able to execute a set of tests over the applications, to ensure the proper functionality of the applications as well as their security. Finally, these three systems will be combined into a single architecture, to take benefit from each other and reduce the security risks these new infrastructures have arisen.

Hence, the structure of this paper is as follows. In section 2 we will expose the background of some solutions that aim to solve these problems and the way they do it. In section 3 some of these solutions will be shown, presenting them in detail. Later, section 4 will show the framework of the solution presented in this paper, including its architecture joining the different solutions previously explained and a first deployment of it, in the shape of integrating the already existing solutions. Finally, section 5 indicates the next steps to be followed and the current work in progress to further evolve and mature the proposed solution

2 Background

As mobile technologies have evolved, the same has happened with the underlying infrastructure supporting them. However, those advantages have carried on some relevant threads that need to be handled. Furthermore, as these risks involve multiple layers of the infrastructure and multiple components, addressing, remediating, or avoiding such risks is not an easy task. For that reason, multiple approaches with the aim of solving this problem can be found, each one with a different point of view.

On the one hand, there exist solutions whose aim is to detect an attack or an anomalous situation before it occurs, such as authors in [1]. In these, monitoring agents are deployed to surveil the network and critical components. The data obtained from this monitoring is processed by other components, usually ML/IA powered (as shown in [2]) that detect when an anomalous situation is occurring or may occur, in order to deploy countermeasures to solve it. In this line we find H2020 EU projects such as ANASTACIA[3] or INSPIRE5G[4], whose purpose was to increase the security of a system by deploying monitoring agents and deploying countermeasures against an attack if necessary. Coming back to ML/IA powered solutions, we also find ARCA [5, 6, 7], which uses several AI methods to support general network automation by delivering autonomic network management processes that enable a managed network to react to internal and/or external events, such as in the case that a natural disaster will bring a huge amount of traffic to the network of a disaster relieve support system.

On the other hand, other solutions are based on deploying those applications or components from third parties in a controlled environment (for example, a sandbox, an isolated place where deploy components or apply changes with no risk of affecting other components or parts of the infrastructure) first, to evaluate the behavior of the applications without compromising the shared infrastructure. In this way, if any suspicious behavior is detected, it is detected without other entities being affected by it, as it is not sharing the infrastructure with any other entity yet. This concept is used in some projects such as FISHY[8], where they designed a sandbox to provide a virtual environment capable of supporting the execution of components and other relevant functions.

Finally, and similar to the above, are there projects that, even if their main objective is not the security of the deployed applications per se, do offer mechanisms that can be used for this purpose. This is the case of the 5GASP project[9], an H2020 project focused on helping SMEs and small developers to adapt their existing applications or develop new ones in 5G environments. To this end, it offers, among others, a testing platform where a series of tests can be performed on the applications they want to deploy. In this way, it is possible to check both their functionality and their reaction to certain situations before their deployment in a real environment.

Therefore, this paper presents the integration of one solution of each type of the above-mentioned to demonstrate how there can be synergies and collaborations between projects with different purposes but with compatible components. In this way, research centers, small companies, and the R&D departments of large companies working together combine all the knowledge and tools they possess, offering transversal solutions with different approaches to solve a complex problem that affects all of them equally.

3 Selected frameworks

Hence, this section will present a solution for each type of approach, and then these solutions will be integrated into an architecture able to address the problem from multiple different points of view.

3.1 ML/AI powered monitoring framework

Concerning solutions powered by Machine Learning and Artificial Intelligence we highlight the Autonomic Resource Control Architecture (ARCA) [5, 6, 7]. ARCA features a closed-loop management cycle for the elastic adaptation of virtual networks to dynamic demands and requirements. ARCA follows the autonomic management scheme [10], which is the adaptation to management of the broader autonomic computing approach. Thus, ARCA defines a separate component for each activity of the the so-called MAPE cycle—which are monitoring, analysis, planning, execution—, and adapts the activities to the network management, so providing autonomic network management in support of the more general network automation goal.

ARCA components operate as follows:

- The monitoring component is able to receive, on the one hand, raw telemetry data and, on the other hand, processed knowledge objects. The former are provided directly by monitoring elements while the latter are provided by specialized processors, such as TKDP [11] and TLC [12]. By using processed information, computation effort is distributed among all elements involved in the management process, allowing the system to get more precise information—by the distributed processing of a bigger amount of monitoring information. In addition to monitoring elements, this component is able to receive notifications from external event detectors, which can be from social network watchers to physical environment detectors. This means that, for instance, ARCA is able to take decisions based on the use of bandwidth or CPU of the system, together with the occurrence of an event, such as a heavy thunderstorm or earthquake. In such case, ARCA would be able to anticipate a high demand of its system and allocate more resources or even anticipate that some sub-system would be unreachable and deploy alternative sub-systems elsewhere.

- The analysis component processes the telemetry data or knowledge objects provided by the monitoring component in order to identify complex events. To support the huge amount of data provided when raw telemetry data is used, the analysis component incorporates a complex event processor that identifies events in the monitored system after processing streams of data without requiring to store all the data in memory. However, when ARCA uses TKDP, because it is available in the environment, such as when ARCA is connected to OSDM—an extension of OSM [13]—, the analysis component relies on the functions TKDP provides, so ARCA requires much less resources to perform the event identification, although all data is still available if needed—stored and provided by the sources. In general, event detection is performed by learning the map between received information—or knowledge object—and the state of the system. Several methods are used together, such as SVM, random forests, and other ensembles. To get the best results, a considerable amount of initial data must be provided for the learning phase. After that, the system will correct flaws by re-learning the map between the monitoring information provided and new states of the system. Apart from using the learned map, eventual situations are detected by using a rule-based method, which determines if some resource is under or over-used and, therefore, avoids either service disruptions or resource waste in case of a bad guess has been found by the map function.

- The planning component is notified of an event if detected by the analysis component and uses several AI methods to find the actions needed to resolve the event. The first step is to match events and potential solutions. This is achieved by using a case-based reasoner (CBR). Typical CBRs map symptoms to cases. In ARCA, we use a CBR to map events to countermeasures. The principle is the same, the CBR is provided with a vast amount of records to learn the map between events and countermeasures. Then, the CBR is asked for the proper countermeasure for an event. In addition, the planner is able to determine the best way to implement—enforce—the countermeasures. For instance, if a new VNF instance must be moved from place A to place B, this component will generate an action plan that includes the copy of the instance from A to B, the startup of the instance in B, the update of all links and forwarding tables needed to get the instance working in its new place, and the removal of the instance in A.

- The execution component—also referred as enforcement process—will receive the plan provided by the previous component and implement it by using the required tools to contact the involved elements. For instance, if a new VNF instance is needed to be created at some remote OSM instance, the execution component will use SSH to access some controller machine and then use a command to contact OSM and request the creation of the VNF instance. Local scenarios are much simpler by allowing the execution component to directly use the API of the NFV VIM or the OSM mediator.

After some decision—plan—is executed, ARCA loops the same decision, checking that the enforced decision has resolved the problem or, otherwise, obtaining new countermeasures and updating its knowledge about it—implying learning new mappings in both the analysis and planning phases.

3.2 Sandbox framework

Although the original aim of FISHY Sandbox[14] was not to be used as a testing sandbox, as they deployed it as a means to deploy all the required components for its project in an agnostic and transparent way, it suits perfectly the idea of having a dynamic and isolated NFV-testbed to execute the applications with no risk. Furthermore, it is currently being adapted to be part of the HORSE project[15] as a testbed environment able to integrate and support the execution of all the components and the possible relevant functions. For communication purposes, it includes two components: a Network Edge Device (NED) and a SIA (Secure Infrastructure Abstraction) module. The first one is deployed at every domain, establishing all the management communications between components, meanwhile the latter is a northbound interface designed to provide an abstraction of the underlying infrastructure resources available in the domain.

The main architecture of the FISHY sandbox can be seen in Figure 1, and it is comprised of a set of different domains. One of them hosts the control services, meanwhile the others provide the abstraction of the underlying infrastructure offered as a sandbox. In this way, the whole framework is able to support different requirements, e.g., offering different infrastructures as containers, regular virtual machines or even OpenStack domains in a transparent and agnostic manner.

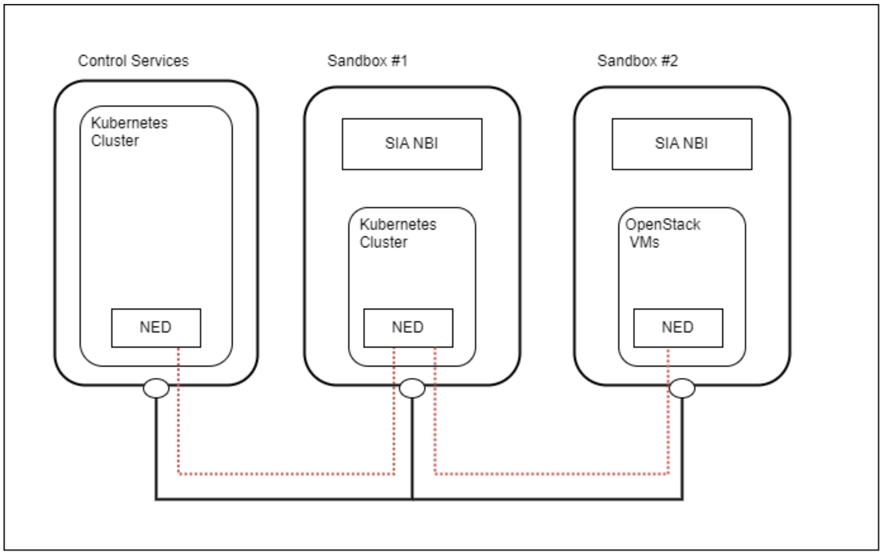

Figure 1: FISHY sandbox architecture

3.3 Testing framework

Finally, we propose the 5GASP testing pipeline[16] as a testing framework. This framework is included in the H2020 5GASP project and is able to automatically perform a series of Robot Tests previously implemented. Concerning those tests and as explained in [17], they are divided into two types: those designed by the developer himself to check the functionality of the application (e.g., if it responds correctly to what it has been programmed for), and those ones that check certain aspects related to security and how the application behaves in certain situations. Apart from the testing framework, 5GASP also offers a methodology for developers [17] to know how to develop their applications as a network application and make them compliant with a 5G network. This methodology also includes a series of best practices to ensure that applications are adapted to the network environment in which they will be deployed.

Although the 5GASP platform includes multiple phases (design and development, onboarding and deployment, and testing and validation), in this paper we will only focus on the last phase, which takes place once the application is deployed and thus the testing and validation process begins. All the aforementioned phases are carried out through and thanks to the inclusion of OpenSlice [18], an Operations Support System (OSS) to deliver Network Slice as a Service (NSaaS)[1]. Through it, apart from the management of the slices (out of the scope of this paper), the execution and gathering of the test results are managed. It is also in charge of the deployment of the applications, delegating it to the correspondent NFV-Orchestrator. The NFV-Orchestrator is the NFV entity in charge of the deployment and life-cycle of the deployed components. Thus, as it is also shown in the workflow of Figure 2, Openslice can be instructed to run a battery of tests after the deployment of the applications. To do so, it delegates the test execution to the CI/CD Manager, which in turn delegates it to the CI/CD Agent of each site. This is due to the fact that the architecture is designed to be multisite, with Openslice and the CI/CD Manager centralized and a CI/CD Agent in each site. The latter is in charge of communicating with the application, executing the tests, and then sending the test results back to the manager, which then sends them to Openslice. In that sense, through this framework the execution of the tests can be automatically performed.

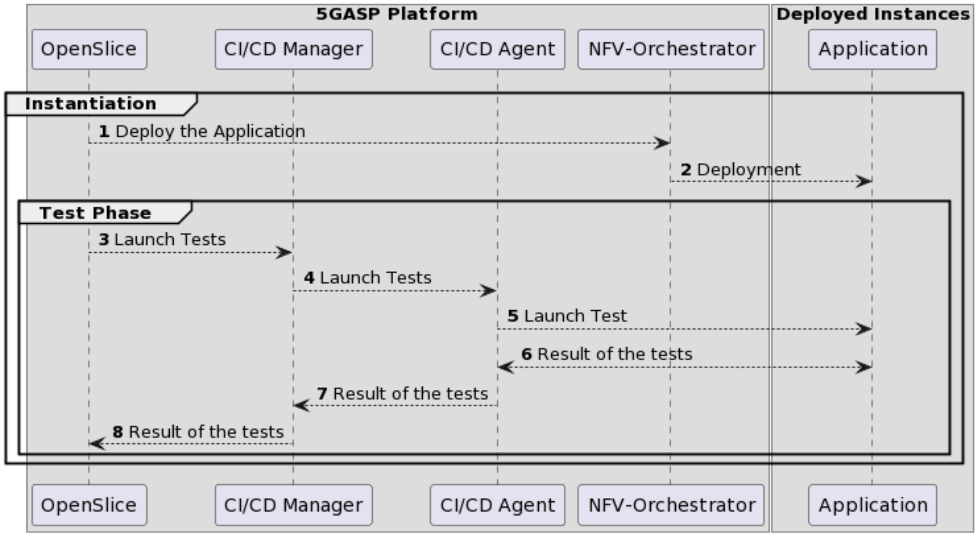

Figure 2: Workflow of the 5GASP's Test Execution

[1] A Network Slice is a concept on which a physical network can be multiplexed – in a virtual way– to offer different independent, isolated, and end-to-end logical networks sharing the same physical network.

4 Proposed Architecture

Once presented the proposed solutions, each of them with a different approach, we proceed to integrate them into a single architecture that will benefit from the different points of view from academia, industry, and research companies, as well as the benefits and solutions offered by each of them. The proposed architecture is shown in Figure 3, and includes the monitoring and reacting capabilities from ARCA framework; the sandbox capabilities from the FISHY sandbox, as well as the Testing pipeline from 5GASP. In that sense, this architecture offers the possibility of deploying an application over an isolated and monitored environment (FISHY sandbox with the monitoring agents from ARCA), and when deployed, through the Testing pipeline, check the functionality of the application, how it reacts against certain incidents, stress it, etc. Moreover, it is also possible to deploy ARCA's countermeasures inside the sandbox, to validate and evaluate them in terms of how they react against the application behavior. This is also useful as this sandbox and testing framework can be used as a training environment for ARCA algorithms and countermeasures, as they can evaluate their efficiency and behavior in an isolated, monitored, and highly controlled scenario.

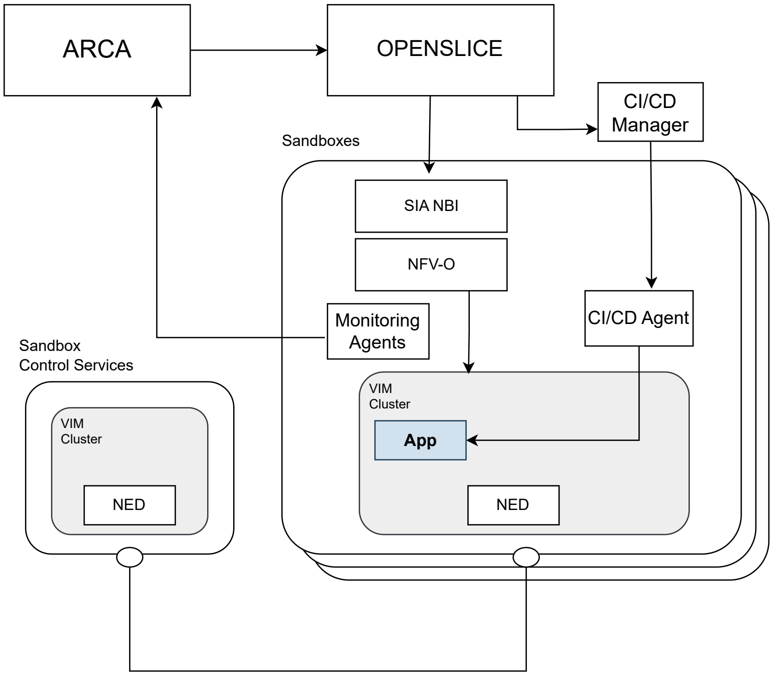

Figure 3: Proposed architecture

In that sense, an example of the use of this architecture would be the following. The administrator of a telecommunications company testbed receives an application to be deployed over their production NFV platform, where other applications are currently running. This administrator can evaluate the new application across this platform, deploying it in one of the FISHY sandboxes available through openslice. The point FISHY offers multiple sandboxes is interesting as it enables the evaluation of the application even if it requires to be deployed in two different sites (for example, a cache or a proxy application where the original source is located in a certain place and the proxy in a different one). As the sandbox offers communication through the NED between sandboxes, this situation can be easily handled. So, once the application and the components it requires are deployed on the sandbox, a suite of tests would be performed over them, to evaluate their behavior and performance. For example, testing the application under network stress situations, or simulating a DDoS attack and monitoring how they react to it. The key part of this is that, as the sandbox is monitored and the gathered data processed by ARCA, the administrator would be able to know whether the new application would pose a risk to the shared infrastructure or to the other applications already deployed on it (and to future applications to be deployed). So, in this example, Openslice requests the deployment of the new application into the sandbox (using the agnostic SIA NBI for that purpose). Once the application is deployed, it asks the CI/CD Manager to start the testing process. This petition is sent to the appropriate CI/CD Agent, who runs the tests over the application and sends back the results. In the meantime, the monitoring agents are analyzing and gathering data from the network and critical components, and sending that information to ARCA. If ARCA does not find any threat and considers the gathered data safe, and the results from the tests are valid (in terms of security, performance, etc.), the process finishes and the administrator has guarantees of the correct functioning of the application. In that sense, that application would be valid to be deployed over the production environment with no risk.

Another benefit of this architecture is that it is agnostic from the NFV-Orchestrator used to manage the sandbox. That means that the administrator or any other entity interested can deploy his applications regardless of their type, i.e., regular OpenStack-based virtual machines, docker containers over a Kubernetes cluster, different NFV-Orchestrators apart from OSM, etc.

4.1 Deployment and integration of the proposed architecture

As a proof of concept and first approach to the deployment of this architecture, it has been considered to use the already deployed components and interconnect them. This approach reflects a realistic environment of multi-domain orchestrated NFV-based networks. As they are located in their correspondent testbed (i.e., FISHY Sandbox in Madrid, 5GASP Testing Framework and Openlisce in Murcia, and ARCA in Tokyo), the first step consisted of adding the necessary elements to interconnect the three sites. For this purpose, wireguard tunnels have been deployed and configured to be used. Wireguard is an open-source tool that securely encapsulates IP packets over UDP, thus offering the functionality of a tunnel or a VPN. Specifically, a wireguard server has been deployed in the Murcia testbed, as well as two wireguard agents in Madrid and Tokyo sites. These agents act as gateways in order to interconnect their own testbed networks with the others. In this way, the components located on any of the three sites are able to access the networks and thus the components of the other sites securely and transparently. Figure 4 shows how this integration has been performed. Every testbed shares a management and a data network with the others. Moreover, the figure represents every component interconnected in each testbed, although the involved ones in the presented use case are the ones highlighted. The three testbeds include a fully functional NFV infrastructure, so after performing the experiment in the sandbox, we could deploy the application in the Murcia site's infrastructure, for example.

In that sense, once the involved networks are reachable from the others, the same process aforementioned can be executed. Openslice requests the deployment of the application in the FISHY sandbox, located in Madrid. In this case, the petition is sent to the OSM in charge, although it could be a different NFV-Orchestrator. Once it is successful, it asks the CI/CD Manager to start the Testing process, who in turn delegates the request to the CI/CD Agent

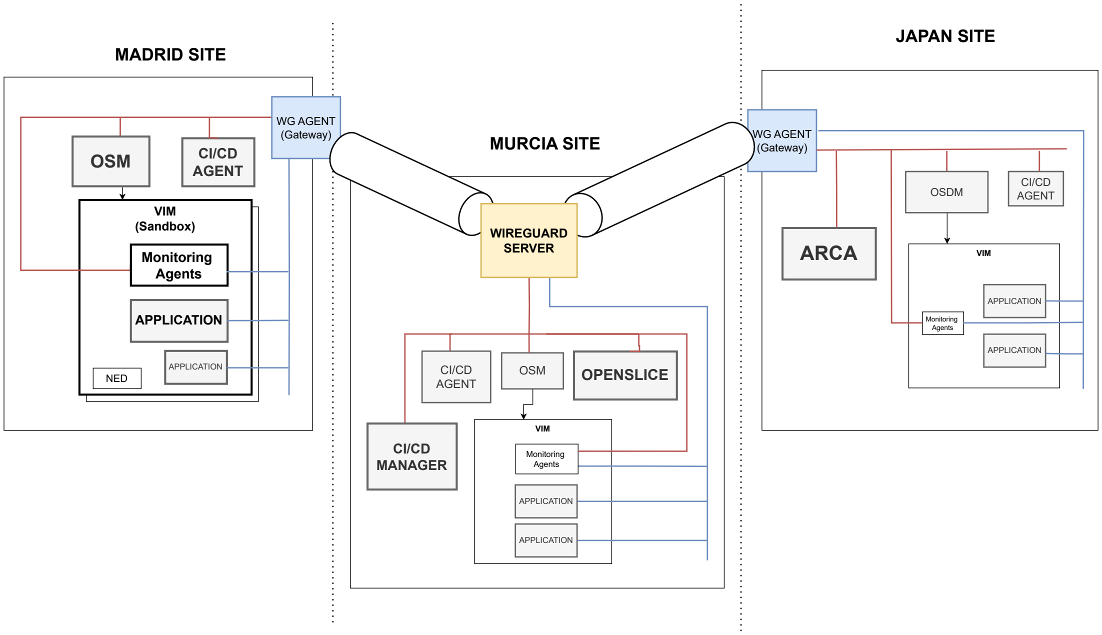

Figure 4: PoC of the proposed architecture. Integration and communication of the three sites through wireguard tunnels

located on the Madrid testbed. This entity performs the tests and sends back the results; meanwhile, the monitoring agents deployed in the sandbox are continuously sending the monitoring data to ARCA. If both processes are successful, the application is considered valid to be deployed in a real production testbed.

5 Conclusions and Future Work

This paper proposes the integration of three existing solutions, all of them from different fields, but highly compatible and able to benefit from each other to offer a more complete solution than separately. Each of them focuses on a particular field, which is extremely interesting as threats concerning security in next-generation networks are too complex to address from a single perspective. In this way, the proposed solution increases the awareness against multiple possible malfunctions or attacks, more than using a single approach. Therefore, we consider this approach a promising idea to bring together three specialized frameworks and make them work together.

In future work, an evaluation of the proposed architecture based on the proposed use case is mandatory, as well as developing more use cases with a higher level of detail, to demonstrate the benefits this architecture can offer. Furthermore, deploying the complete architecture on a single testbed is also considered, so a more exhaustive evaluation of the performance can be conducted with no influence of latencies when communicating sites so widely separated. It is also considered to evaluate different deployment combinations, even deploying some components of the architecture dynamically, taking advantage of the benefits that NFV offers. Evaluating the performance of the countermeasures required by ARCA when an application is malicious or does not react as expected will be also reported.

Finally, it should be noted that both OpenSlice and FISHY Sandbox are currently evolving, the former as part of the EU project HORSE and the latter as an ETSI developing group, so improvements in these components are expected and will be included in the proposed architecture.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2022-00166729). This work has been supported by the Spanish Ministry of Science and Innovation under the DIN2019-010827 Industrial PhD Grant, and co-funded by Odin Solutions S.L.; by the Spanish Ministry of Economic Affairs and Digital Transformation under the projects MAS4CARE (Grant No. TSI-065100-2023-006), PROMETEO-6G (Grant No. TSI-064200-2022-003) and GAIA-6G (UNICO I+D 5G-6G 2022 program - Grant No. TSI-064100-2022-2); by the Spanish Ministry of Science and Innovation (MCIN) and the Spanish Research Agency (AEI), under the project ONOFRE-3 (Grant PID2020-112675RB-C44 funded by MCIN/AEI/10.13039/501100011033); by the European Commission under the 5GASP project (Grant No. 101016448); and from the Smart Networks and Services Joint Undertaking (SNS JU) of the European Comission under the HORSE project (Grant No. 101096342).

References

[1] Alejandro Molina Zarca, Jorge Bernal Bernabe, Ruben Trapero, Diego Rivera, Jesus Villalobos, Antonio Skarmeta, Stefano Bianchi, Anastasios Zafeiropoulos, and Panagiotis Gouvas. Security management architecture for nfv/sdn-aware iot systems. *IEEE Internet of Things Journal*, 6(5):8005–8020, 2019.

[2] Sehar Zehra, Ummay Faseeha, Hassan Jamil Syed, Fahad Samad, Ashraf Osman Ibrahim, Anas W. Abulfaraj, and Wamda Nagmeldin. Machine learning-based anomaly detection in nfv: A comprehensive survey. *Sensors*, 23(11), 2023.

[3] H2020 anastacia project. http://www.anastacia-h2020.eu/.

[4] H2020 inspire-5g+ project. https://www.inspire-5gplus.eu/, 2021.

[5] P. Martinez-Julia, S. Homma, and D. R. Lopez. Artificial intelligence framework for network management. https://datatracker.ietf.org/doc/html/draft-pedro-nmrg-ai-framework, 2023. IETF Draft.

[6] P. Martinez-Julia, V.P. Kafle, and H. Harai. Exploiting external events for resource adaptation in virtual computer and network systems. *IEEE Transactions on Network and Service Management*, 15(2):555–566, 2018.

[7] P. Martinez-Julia, V.P. Kafle, and H. Asaeda. Explained intelligent management decisions in virtual networks and network slices. In *Proceedings of the 23th ICIN Conference (Innovations in Clouds, Internet and Networks, ICIN 2020)*, pages 1–7, Washington, DC, USA, 2020. IEEE.

[8] H2020 fishy project. https://fishy-project.eu/, 2021.

[9] H2020 5gasp project. https://www.5gasp.eu/.

[10] Alex Galis et al. Position Paper on Management and Service-aware Networking Architectures (MANA) for Future Internet. http://www.future-internet.eu/fileadmin/documents/prague_documents/MANA_PositionPaper-Final.pdf, 2010.

[11] Pedro Martinez-Julia, Ved P. Kafle, and Hitoshi Asaeda. Telemetry knowledge distributed processing for network digital twins and network resilience. In *Proceedings of the 2023 IFIP/IEEE*

Network Operations and Management Symposium (NOMS), pages 1–6, Washington, DC, USA, 2023. IEEE.

[12] Pedro Martinez-Julia, Ved P. Kafle, and Hitoshi Asaeda. A novel telemetry compression method for enhancing network resilience. In *Proceedings of the 2023 26th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, pages 174–178, Washington, DC, USA, 2023. IEEE.

[13] OSM. OSM Release Five Technical Overview, 2019. OSM White Paper .

[14] UC3M. Fishy sandbox. https://github.com/Networks-it-uc3m/FISHY-Sandbox-development/.

[15] Eu horse-6g project. https://www.horse-6g.eu/.

[16] Kostis Trantzas, Christos Tranoris, Spyros Denazis, Rafael Direito, Diogo Gomes, Jorge Gallego-Madrid, Ana Hermosilla, and Antonio Skarmeta. An automated ci/cd process for testing and deployment of network applications over 5g infrastructure. In *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 156–161, 2021.

[17] 5GASP. D4.2 final methodology for designingand developing netapps. https://www.5gasp.eu/assets/documents/deliverables, 2021.

[18] ETSI. Openslice. https://osl.etsi.org/.