This work has been accepted for publication in 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN).

NFV-driven intrusion detection for smart manufacturing

Daniel Behnke*, Marcel Müller*, Patrick-Benjamin Bök*, Stefan Schneider†, Manuel Peuster†, Holger Karl†, Alberto Rocha‡, Miguel Mesquita‡, and José Bonnet‡
*Weidmüller Group: {daniel.behnke, marcel.mueller, patrick-benjamin.boek}@weidmueller.com
†Paderborn University: stefan.schneider@upb.de, manuel.peuster@upb.de, holger.karl@upb.de
‡Altice Labs: {alberto.rocha, miguel.mesquita, jose.bonnet}@alticelabs.com

***Abstract*—The significant progress in softwarization of hardware components with technologies like Network Function Virtualization (NFV) enables manifold applications for the industry, especially for smart manufacturing. The gained agility and flexibility leverages data gathering and analysis. In this work, we focus on a very important precondition for networked manufacturing: cyber security. We provide concepts and a first proof-of-work for an cloud-native NFV-driven Intrusion Detection System using Kubernetes, stating challenges we solved during the process and the used software tools. Focusing on traffic monitoring and filtering to enable certain guidelines to ensure the integrity of the factory network by an automatic reconfiguration of the Network Services.**

I. Introduction

The ongoing trend of increasing digitalization in manufacturing leverages automation, facilitates to gather data and analyze the data to optimize the manufacturing processes. Key enablers in terms of enhanced commnication systems are 5G and Network Function Virtualization (NFV) [1]. With NFV, smart manufacturing scenarios can be softwarized and implemented as flexible network services consisting of interconnected virtual network functions (VNFs), which can run on commodity servers.

In recent years, first research has been started on 5G and especially NFV usage in manufacturing. Existing work focuses on the potential benefits and architecture concepts [2], [3]. Now, testbeds and first experiments are the obvious continuation.

Besides proofing the usability of NFV-technology for smart manufacturing, common challenges experienced by all companies which drives digitalization have to be addressed. An important challenge is cyber security, saving data and data transmissions to avoid loss of data and prevent potential attacks on the manufacturing. The Federal Office for Information Security in Germany listed in [4] manifold threats like malware or hacking attacks for companies. One solution to detect ongoing attacks and react to them accordingly are Intrusion Detection Systems (IDS), which have gained significant attention of researchers in recent years. In [5], the authors present a multi-agent approach for hybrid intrusion detection in industrial networks. They create network analysis agents using Zeek[1] on Raspberry Pi which indicates that this approach

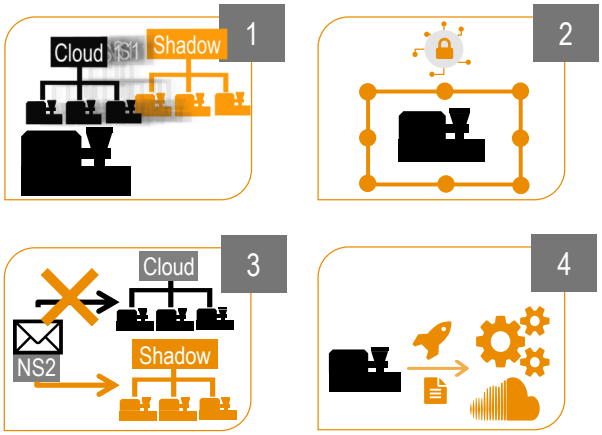

Fig. 1: A softwarized IDS detects a potential threat, NFV technology facilitates to isolate the machine and to re-route the data.

is based on additional hardware. In [6], the authors propose an IEC 61499 Service Interface Function Blocks (SIFB) based Network Intrusion Detection and Prevention System (IDPS) solution to protect programmable logic controllers (PLCs), where the security functions are provided through SIFB executing Snort[2].

Leveraging the experience of Weidmüller Group[3], a large-scale manufacturer, we previously designed an NFV-based architecture for smart manufacturing with multiple use cases [7], [8]. This work is done in the framework of the Horizon2020 research project 5GTANGO. The system architecture consisting of a verification & validation platform, an SDK, and a service platform for the management and orchestration of all Network Services is introduced in [9].

We demonstrated the scalability of our architecture by emulating several interconnected and globally distributed factories that use our developed network services simultaneously [10].

The work has shown how NFV might be used in manufacturing in general. The focus of this work is on the specific use case of intrusion detection. IT security, and here especially cyber security, is a major concern of companies connecting

[1]https://www.zeek.org/

[2]https://www.snort.org/

[3]https://www.weidmueller.com

their machines with cloud services and further web-based services.

The focus has recently shifted towards so-called cloud-native VNFs, facilitated by tools like Docker[4] and Kubernetes (K8s)[5]. Rather than using VMs, these VNFs are implemented in light-weight, stateless containers, which can be started and stopped in seconds. This enables quick recovery from failures as well as flexible scaling and migration, which are crucial for NFV [11].

In this work, we introduce the architecture of the cloud-native NFV-driven smart manufacturing solution in Secion II, before we focus on the integration of Intrusion Detection into the framework in Section III and the MANO workflow to react to intrusion alarms accordingly by the reconfiguration of Network Services in Section IV. We conclude the work with an conclusion and outlook in Section V.

II. NFV-driven Intrusion Detection System

Fig. 1 depicts the envisaged use case for such an IDS. In [7], we have introduced the concept for the interconnection of injection moulding machines to a machine park-wide network. Each machine is represented network-wise by a Network Service, the machine interconnection network service (NS2), which is connected to the factory edge network service (NS1), another Network Service running on the SONATA service platform[6]. One important element of an NS2 is the machine data collector (MDC), it connects the machine itself to the network via the data exchange format Euromap63[7] interface and forwards the data using MQTT[8] to the NS1.

Each instance of NS2 contains an IDS to monitor incoming traffic and detect potential threats. In case of a detected threat, the connection to NS1 is immediately dropped and replaced by an isolated quarantine instance of NS1 without access to the factory cloud. In doing so, the incoming data of this machine is isolated, so a spread of the attack is avoided, but still stored in a separate local database so that no data is lost. Once the threat is contained, the data can be merged and pushed to the factory cloud.

The now possible quarantine of specific machines or components facilitates the integration of IoT components into factory networks. IoT hardware often lacks sufficient security mechanisms, isolating each component behind individual firewall and intrusion detection minimizes the risks.

III. System overview of a cloud-native IDS

Nowadays security is a must. Aware of this fact, a security barrier was designed and included, built of open source products, to protect traffic between machine parks and the MDC of the NS2 sited at an edge or core data center. By attaching an Intrusion Detection System (IDS) next to the MDC we will be able to inspect traffic and allow or block certain traffic patterns. Moreover, adding an ELK stack [9] allows the processing and filtering of relevant logged data at Logstash, its storage at Elasticsearch's time series database and its visualization on Kibana dashboards.

The filtering can now be done at the IDS virtualization deployment unit (VDU), by adding or changing IDS rules, or at the Logstash VDU, by processing the incoming events at its filter plugin. This document is focused on the latest scenario.

A. The building blocks of the "security barrier"

Following the actual IT tendency to adopt containerized VNF, benefiting from its flexibility and lightweight hypervisor technology, there was a decision to adopt a cloud native approach. Then, all the Network Services are built of containerized VNFs' (aka, CNF or VNFC on ETSI lingo) orchestrated by Kubernetes. Here is the list of used components as depicted in Fig. 2:

- **Suricata**[10] IDS supported by OISF was the chosen IDS because it is based on a new generation of IDS with multi-thread capabilities. An IDS runs in promiscuous mode to capture every packet arriving at its network interface. Instead of building a new Docker image from the scratch, a fully functional Suricata Docker image was selected from Docker Hub after a few functional tests.
- **Filebeat**[11] is the tool from Elastic Beats family that is specialized in collecting logs and send them to the Logstash pipeline or directly to the Elasticsearch time series database. It mounts the EVE json output from Suricata and package the events to the Elastic stack. Under the circumstance they are candidates to live together as two VDU's of a single VNF, where Filebeat VDU acts has a sidecar of Suricata VDU. The official Filebeat Docker image was selected from Docker Hub.
- **Elasticsearch**[12] is a time series database with a built-in analytics engine and accessed by a REST API for insertion and search. The official Elasticsearch Docker image was selected from Docker Hub.
- **Logstash**[13] is a central dataflow engine in the Elastic Stack for gathering, enriching, and unifying all of your data regardless of format or schema. It provides a processing pipeline for data extraction, transformation and loading, ie, a streaming ETL engine for data logistics. It accepts multiple data sources and it contains several common filters. It is possible to include your own specific filters and upload them to a persistent data store like Elasticsearch. The official Logstash Docker image was selected from Docker Hub.
- **Kibana**[14] is a web application to visualize any kind of structured and unstructured data indexed in Elasticsearch

[4]https://www.docker.com
[5]https://kubernetes.io/
[6]http://www.sonata-nfv.eu/content/service-platform
[7]http://www.euromap.org/files/eu63.pdf
[8]http://mqtt.org/
[9]https://www.elastic.co/de/what-is/elk-stack
[10]https://suricata-ids.org/
[11]https://www.elastic.co/products/beats/filebeat
[12]https://www.elastic.co/
[13]https://www.elastic.co/products/logstash
[14]https://www.elastic.co/products/kibana

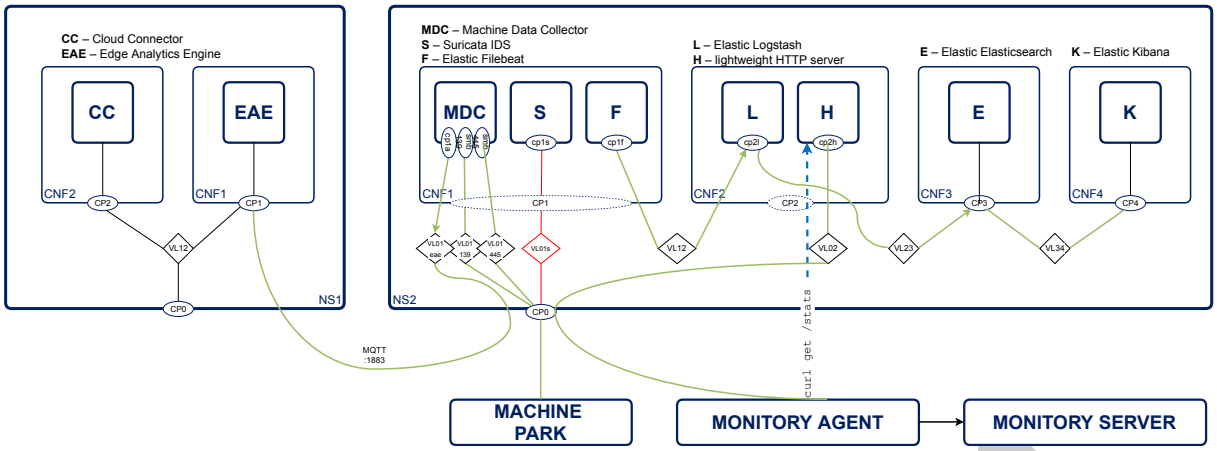

Fig. 2: IDS Architecture.

(among other features). The official kibana Docker image was selected from Docker Hub.

In our scenario, the goal of IDS is to inspect all messages directed to MDC, so the best solution would be to run Suricata as a sidecar of MDC: making use of a fundamental concept of Kubernetes that all containers in a Pod share the same 'localhost' interface inside that Pod, then Suricata is able to capture all the traffic directed to MDC because they share the same network interface.

Note that there is an alternative topology by decoupling IDS from MDC and attaching the former to the host network interface. However this solution has a few drawbacks like having to filter ALL the network traffic and not only the one directed to MDC and an inconvenient dependency on the naming convention of the network interface name (instead of be named 'eth0', like in the Pod, it could be named 'ens3p0' or 'em3' etc., depending on the operating system distribution, ie, more logic is needed to identify the correct string and to use the right name; additionally, the migration is more difficult if the configuration is dependent on the physical host).

As soon as Logstash starts receiving logging messages from Filebeat it will process those messages by applying the defined filter and acting as a monitory entity. Actually, all received messages are discarded except those that can anticipate an attack and that are forwarded to the ELK stack. An example for a filter is the block of non expected traffic (to prevent the establishment of unauthorized connections to unknown IP addresses or ports). Whenever an unexpected message occurs, an event is generated to trigger a specific action for the unforeseen occurrence.

B. The processing of an suspicious event

Any possible intrusion event detected by Logstash is also registered to a lightweight HTTP server (1) that also runs as a Logstash's sidecar. This is the simplest way to interact with the ETSI NFV compliant's Service Platform (SP) to take advantage of its MANO features, benefiting from the monitory agent pull mechanism (2). The monitory agent collects those events and forwards every new event to the monitory server

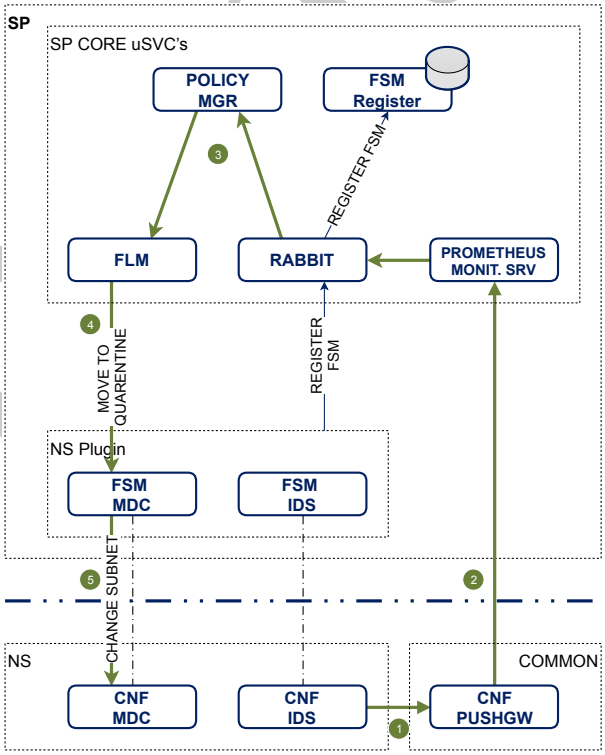

Fig. 3: Workflow of alarm triggering.

running at the SP. Now, the SP is handling the information flow: the monitory server notifies the Function Lifecycle Manager (FLM)(3) and the FLM notifies the Function Specific Manager (FSM)(4) of this Network Service. The workflow is depicted in Fig. 3.

According to the micro services architecture of the SP, each CNF should have its own manager, the FSM. This FSM is the only entity that knows the specifics of the according CNF. Afterwards, the subnet under attack is moved to another subnet if the attack is detected. The resources are isolated on a quarantine network for further handling by the support

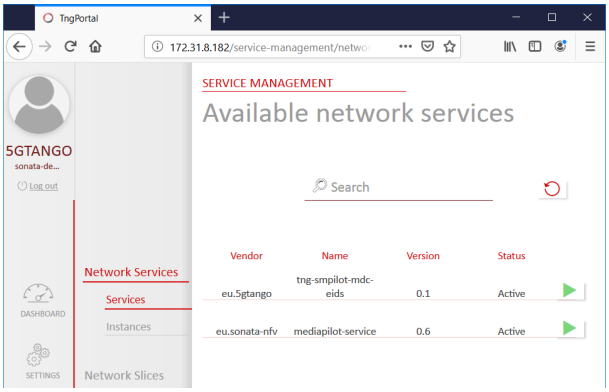

Fig. 4: Deployment of the Network Service.

team. More details on the MANO control loop are provided in section IV.

C. SMP Network Service Deployment

The Service Management Platform (SMP) official repository contains YAML files for manual deployment[15] to a K8s cluster using 'configMap' primitives to configure the specifics of a particular environment. And it also contains individual ready-to-go images to create a package with SDK tools to deploy via SP.

NFV as defined by ETSI prescribes the automated management and orchestration of the virtualized network functions. The deployment and operation of a Network Service make use of descriptors to indicate the required resources from the provisioned infrastructure (NVFI) and the behavior of the VNF in the presence of changes on the runtime environment.

The SP use SONATA-NFV schema for its own NS Descriptors (NSD) and VNF Descriptors (VNFD). The schema extends the basic functionality of ETSI NFV Deployment Templates that are based on OASIS TOSCA Simple Profile for NFV.

Besides common fields like Vendor, Version, and Name, the VNFD for a cloud native CNF must contain the image source and its name (and optionally the required container resources like vcpu, memory and storage), the Connection Points (CP) and the Virtual Link (VL) for that CP to connect to other CNFs. The VNFDs for SMP are published here[16].

Again, besides common fields like Vendor, Version, and Name, the NSD must reference all the CNF images that build the NS, their CPs and their VLs, and the forwarding graphs to build the NS topology as illustrated on the initial picture. The NSD for SMP is published here[17].

The value of this solution is that it can be applied to other scenarios without customization code but simply by changing the configuration files, namely adding or changing Suricata rules or Logstash filters.

In Fig.4 the started Networt Services are shown in the web interface of the 5GTANGO platform.

IV. Putting MANO in the loop

Deploying smart manufacturing functionalities within virtualized services greatly improves their flexibility and manageability [8]. Key enabler for this is the fact that deployments are fully automated and end-to-end lifecycles are entirely managed by MANO systems. There are, however, much more opportunities that result from the introduction of logically centralised MANO systems. In particular if the latest generation of MANO systems is considered which offers advanced programmability features providing new degrees of freedom [12]. The following sections explain how a smart manufacturing IDS can make use of this new generation of MANO systems to establish an end-to-end control loop and provide automated defence mechanisms for intrusion threats.

A. Closing the control loop

Running an IDS as a virtualized NFV service, as described in Section III, already provides a lot of benefits. For example, it can be automatically deployed and reconfigured on-demand. Further it is possible to let the IDS automatically adapt to load and environment changes, e.g., by scaling up and down its resources. This degree of automation is, however, not the limit and we show that in a fully softwarized environment even more automation and interoperation between components can be achieved.

Our goal is that, once the IDS detects an intrusion, the system can automatically reconfigure the involved smart manufacturing services as a counter measure to protect the rest of the factory network. Instead of just disconnecting and terminating the attacked smart manufacturing services, we move parts of them into isolated environments to decouple them from the factory network and still keep the possibility to further investigate the detected intrusion.

To build such a system, the virtualized IDS needs to be able to reconfigure other parts of the deployment, which could be done by using custom IDS and service implementations, specifically tailored to this use case. Even though this is a valid design option, it has a lot of shortcomings, e.g., it requires a direct coupling of the IDS with the reconfigured service components, and substantial development efforts.

A better design option is to make use of already existing interfaces of the NFV deployment. There are, for example, monitoring interfaces that expose information from the deployed services to the NFV platform, i.e., the MANO system [13], [14]. Moreover, existing NFV platforms offer management interfaces towards the deployed components to manage their lifecycles and do reconfigurations [12]. Using these two interfaces, it is possible to build a closed control loop among the involved service components, i.e., VNFs, and let one component trigger the reconfiguration of another one—all possible without requiring any additional or customized interfaces between the service components.

[15]https://github.com/sonata-nfv/tng-industrial-pilot/tree/master/k8s/ns2-e-ids-v2

[16]https://github.com/sonata-nfv/tng-industrial-pilot/tree/master/sdk-projects/tng-smpilot-ns2-k8s-eids/vnfd

[17]https://github.com/sonata-nfv/tng-industrial-pilot/tree/master/sdk-projects/tng-smpilot-ns2-k8s-eids/nsd

B. The need for MANO programmability

Putting MANO systems into the control loops of custom services has manifold benefits, e.g., the MANO's monitoring features can directly be exploited to give the platform or service owner direct feedback from the deployed services. Still, MANO systems can also limit the degree of freedom of such control loops, if they lack a mechanism to customize their behavior, e.g., react to complex events or perform custom actions. In our recent work, we showed the advantages of customizable MANO solutions and demonstrated that it is hard to find a "one-size-fits-all" solution for these tasks [14], [15].

In this paper, we argue that we can simplify the creation of a closed control loop and build a reactive system for the presented smart manufacturing use case with a minimal amount of custom code. To do so, we exploit the advanced customization features, like service- and function-specific programmability, of modern MANO solutions, such as SONATA-NFV [16] or OSM [17].

The SONATA-NFV platform calls these service- and function-specific control components, *service-* and *function-specific managers* (SSM/FSM). These SSMs or FSMs are containerized components that contain custom control code which is bundled and on-boarded together with a network service or a VNF. They are deployed as part of SONATA's extensible MANO platform if the corresponding services or functions are instantiated [14], [15]. These small control components can then be used to customize (overwrite) tasks that the MANO system performs for a specific service or function. This is combined with a policy management component that allows to trigger actions, e.g., trigger the execution of code in SSMs and FSMs, based on rules matched against system information, e.g., monitoring metrics. Those rules are also bundled and on-boarded together with the services and thus allow to be fully customized by a service developer. Still, the MANO systems ensures that the actions performed by such control components always are within predefined limits, like an sandbox environment, to not create vulnerabilities towards the MANO system.

Similar to the FSM/SSM does OSM offer an approach called *service-* and *function-primitives* that are realized using Juju Charms[18] to enable advanced MANO programability. Those Charms can react to system or monitoring events and are able to manage and reconfigure VNFs during their entire lifecycle.

The SONATA-NFV and OSM approaches for customized MANO operations can both be used to implement the presented smart manufacturing use case, but we limit ourselves to explain the detailed workflow, as it is implemented in our prototype, using the SONATA-NFV platform. Nevertheless, the concepts and workflows presented in the next section can be applied to both platforms.

[18]https://www.jujucharms.com

C. Resulting end-to-end information flow

Figure 5 shows the end-to-end workflow of the proposed smart manufacturing use case that utilizes the MANO system to react to an attack detected by the IDS to reconfigure parts of the machine interconnection service (i.e. MDC).

Starting from a high level, the figure can be divided into two parts. The left part shows the virtualized network services and some of their components, such as IDS and MDC. This part is automatically deployed on NFV infrastructure and controlled by a MANO system. Parts of this MANO system are shown in the right half of Figure 5.

If the system, more specifically the virtualized IDS, detects an attack (1), it fires an alarm (2) that is captured by the monitoring backend of the virtualized service. This monitoring backend, consisting of and ElasticStack deployment[19], forwards the alarm to the default monitoring interfaces of the used MANO system depicted by the *Monitoring Adapter* (3). This adapter is responsible to expose the collected information to the MANO system itself (4) where it is stored. At this stage the attack information crosses the border between virtualized service and the MANO domain. It is important to note that up to this stage no custom interfaces or custom code was needed to propagate this information through the system.

Once the attack information arrives in the MANO system, the *Policy Engine* is informed and can match rules against the arriving information. At this stage, custom policy rules can be used to react on the arriving attack information (6) and possibly trigger further actions (7). Those custom rules could easily be defined from the service developer and do not require changes of the MANO system's code. In the SONATA platform, those actions are forwarded to the *Function Lifecycle Manager* (FLM) which reacts to them. The behavior of this FLM, in turn, can be overwritten by adding FSMs and/or SSMs to the system (8). In our example, the action is overwritten by one or more FSMs, which know how to react and implement the required logic to reconfigure the involved Network Services (9). This reconfiguration task is then performed by the normal management interfaces used by the MANO system to control the deployed services and VNFs (10). This results in a reconfigured service upon the detection of an alarm and closes the control loop. It is worth noting that the shown IDS and reconfiguration use case is just an example and such a "MANO-in-the-loop system" allows for arbitrary control actions that can be triggered by all kinds of events.

V. Conclusion

In this work, we introduced a system approach and implementation of an intrusion detection system using the NFV-based 5GTANGO platform. We highlighted challenges during creation and provide potential solutions for the traffic monitoring and alarm triggering using 5GTANGO's service platform.

We will integrate the developments into demonstrations as a proof-of-concept in future work showing the potential of NFV systems to secure future factory networks facilitating the

[19]https://www.elastic.co

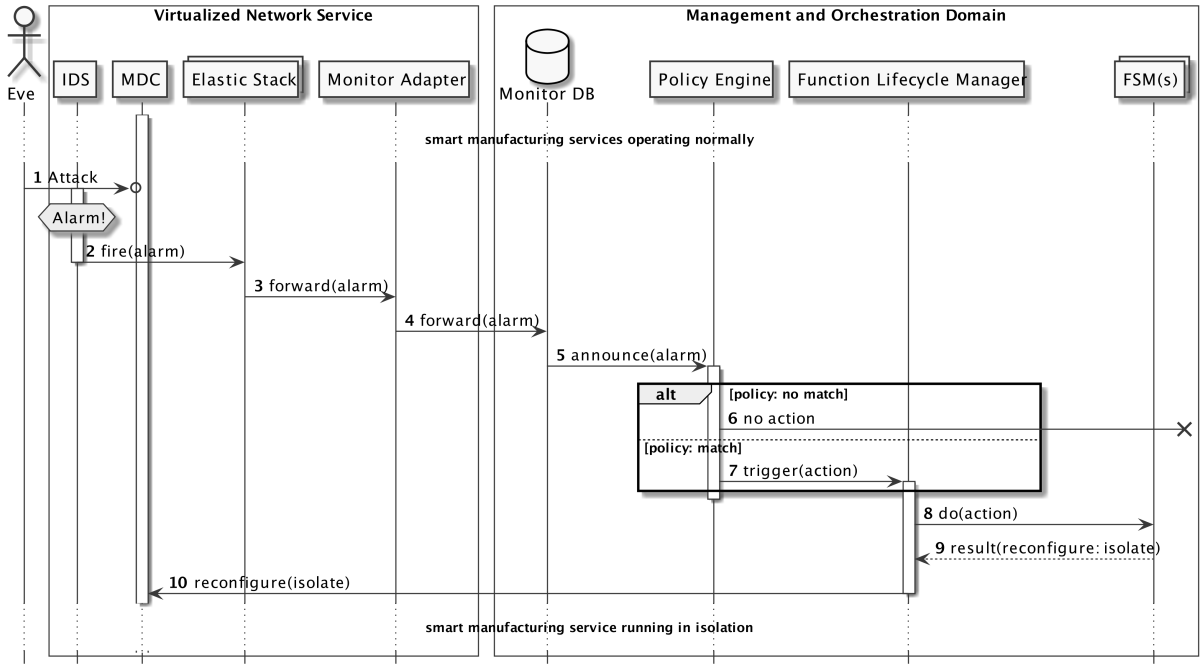

Fig. 5: End-to-end information flow of IDS control loop, utilizing the programmability of the SONATA-NFV service platform.

integration of industrial IoT components and improving the agility of factory networks.

ACKNOWLEDGMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. H2020-ICT-2016-2 761493 (5GTANGO), and the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901).

REFERENCES

[1] 5GPPP, "5G empowering vertical industries," 5GPPP, White Paper, 2016.

[2] B. Blanco, J. O. Fajardo, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-Romero, I. Trajkovska, P. S. Khodashenas, L. Goratti, M. Paolino *et al.*, "Technology Pillars in the Architecture of Future 5G Mobile Networks: NFV, MEC and SDN," *Computer Standards & Interfaces*, vol. 54, pp. 216–228, 2017.

[3] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[4] F. O. for Information Security in Germany (BSI), "The state of it security in germany 2018," Federal Office for Information Security in Germany (BSI), Tech. Rep., 2018.

[5] C. V. Martinez, M. Sollfrank, and B. Vogel-Heuser, "A Multi-Agent Approach for Hybrid Intrusion Detection in Industrial Networks: Design and Implementation," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, 2019.

[6] A. Tanveer, R. Sinha, S. G. MacDonell, P. Leitao, and V. Vyatkin, "Designing Actively Secure, Highly Available Industrial Automation Applications," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, 2019.

[7] S. Schneider, M. Peuster, D. Behnke, M. Müller, P.-B. Bök, and H. Karl, "Putting 5G into Production: Realizing a Smart Manufacturing Vertical Scenario," in *European Conference on Networks and Communications (EuCNC)*, 2019.

[8] M. Müller, D. Behnke, P.-B. Bök, M. Peuster, S. Schneider, and H. Karl, "5G as Key Technology for Networked Factories: Application of Vertical-specific Network Services for Enabling Flexible Smart Manufacturing," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, 2019.

[9] M. Zhao, F. L. Gall, P. Cousin, R. Vilalta, R. Muñoz, S. Castro, M. Peuster, S. Schneider, M. Siapera, E. Kapassa, D. Kyriazis, P. Hasselmeyer, G. Xilouris, C. Tranoris, S. Denazis, and J. Martrat, "Verification and Validation Framework for 5G Network Services and Apps," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017, pp. 321–326.

[10] M. Peuster, S. Schneider, D. Behnke, M. Müller, P.-B. Bök, and H. Karl, "Prototyping and Demonstrating 5G Verticals: The Smart Manufacturing Case," in *IEEE Conference on Network Softwarization (NetSoft*, 2019.

[11] S. Schneider, S. Dräxler, and H. Karl, "Trade-Offs in Dynamic Resource Allocation in Network Function Virtualization," in *IEEE Global Communications Conference (GLOBECOM) Workshops*, 2018.

[12] S. Dräxler, H. Karl, M. Peuster, H. R. Kouchaksaraei, M. Bredel, J. Lessmann, T. Soenen, W. Tavernier, S. Mendel-Brin, and G. Xilouris, "SONATA: Service programming and orchestration for virtualized software networks," in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 973–978.

[13] P. Trakadas, P. Karkazis, H.-C. Leligou, T. Zahariadis, W. Tavernier, T. Soenen, S. Van Rossem, and L. Miguel Contreras Murillo, "Scalable monitoring for multiple virtualized infrastructures for 5g services," in *SoftNetworking 2018, The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization*, 2018, pp. 1–4.

[14] T. Soenen, F. Vicens, J. Bonnet, C. Parada, E. Kapassa, M. Touloupou, E. Fotopoulou, A. Zafeiropoulos, A. Pol, S. Kolometsos, G. Xilouris, P. Alemany, R. Vilalta, P. Trakadas, P. Karkazis, M. Peuster, and W. Tavernier, "Sla-controlled proxy service through customisable mano supporting operator policies," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, April 2019, pp. 707–708.

[15] H. R. Kouchaksaraei, S. Dräxler, M. Peuster, and H. Karl, "Programmable and flexible management and orchestration of virtualized network functions," in *2018 European Conference on Networks and Communications (EuCNC)*, June 2018, pp. 1–9.

[16] SONATA consortium, "SONATA Project," http://sonata-nfv.eu, 2017.

[17] ETSI OSM, "Open Source MANO: Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV," Website, 2016, online at https://osm.etsi.org.