5GZORRO

Grant Agreement 871533

H2020 Call identifier: H2020-ICT-2019-2
Topic: ICT-20-2019-2020 - 5G Long Term Evolution

D4.4: Final Design of Zero Touch Service Management with Security and Trust Solutions

Dissemination Level		
☒	PU	Public
☐	PP	Restricted to other programme participants (including the Commission Services)
☐	RE	Restricted to a group specified by the consortium (including the Commission Services)
☐	CO	Confidential, only for members of the consortium (including the Commission Services)

Grant Agreement no: **871533**	Project Acronym: **5GZORRO**	Project title: **Zero-touch security and trust for ubiquitous computing and connectivity in 5G networks.**

Lead Beneficiary: **UMU**	Document version: **V1.1**

Work package: **WP4 – Zero Touch Automation with Trust, Security and AI**

Deliverable title: **D4.4: Final Design of Zero Touch Service Management with Security and Trust Solutions**

Start date of the project: **01/11/2019** **(duration 30 months)**	Contractual delivery date: **30.04.2022**	Actual delivery date: **10.05.2022**

Editor(s) Gregorio Martínez Pérez (UMU)

List of Contributors

Participant	Short Name	Contributor
Universidad de Murcia	UMU	Gregorio Martínez Pérez, Manuel Gil Pérez, José María Jorquera Valero, Pedro Miguel Sánchez Sánchez
Ubiwhere	UW	Filipa Martins and Carlos Jorge
Atos Spain	ATOS	Guillermo Gómez Chavez
Altice Labs	ALB	Bruno Santos
Fundaciò i2CAT	I2CAT	Javier Fernández Hidalgo, Adriana Fernández, Carlos Herranz Claveras, M. S. Siddiqui
IBM Israel Science and Technology	IBM	David Breitgand, Avi Weit, Katherine Barabash
Nextworks	NXW	Pietro G. Giardina, Juan Brenes, Elena Bucchianeri, Giacomo Bernini
Intracom	ICOM	Alexandros Valantasis, Vasileios Theodorou
Fondazione Bruno Kessler	FBK	Rasoul Behravesh

List of Reviewers

Participant	Short Name	Contributor
Fondazione Bruno Kessler	FBK	Rasoul Behravesh
Nextworks	NXW	Pietro G. Giardina, Giacomo Bernini
Fundaciò i2CAT	I2CAT	M. S. Siddiqui

Change History

Version	Date	Partners	Description/Comments
0.0	24-03-2022	UMU	Table of Contents and editing assignments
0.1	15-04-2022	NXW, ICOM, UMU	1st round of contributions to Chapters 2, 4 and 5.
0.2	19-04-2022	ICOM, UW, ATOS, FBK, UMU	2nd round of contributions to Chapters 2 and 4, 1st round of contributions to Chapter 3. Contributions to Introduction and Conclusions.
0.3	27-04-2022	ICOM, UW, ATOS, IBM	3rd round of contributions to Chapters 2 and 4. 2nd round of contributions to Chapter 3. Review version.
0.4	29-04-2022	FBK, NXW, UW, i2CAT, UMU	Internal review by FBK and NWX. Refinement round after internal reviews.
v1.0-QA-TM	04-05-2022	NXW	Internal review by TM
v1.1	10-05-2022	NXW, ALB, ICOM, UW, UMU, IBM, FBK	Comments from TM addressed. Consolidation and final edition.
V1.1 final	10-5-2022	I2CAT	Final quality check by Project Coordinator.

DISCLAIMER OF WARRANTIES

This document has been prepared by 5GZORRO project partners as an account of work carried out within the framework of the contract no 871533.

Neither Project Coordinator, nor any signatory party of 5GZORRO Project Consortium Agreement, nor any person acting on behalf of any of them:

- makes any warranty or representation whatsoever, express or implied,
  - with respect to the use of any information, apparatus, method, process, or similar item disclosed in this document, including merchantability and fitness for a particular purpose, or
  - that such use does not infringe on or interfere with privately owned rights, including any party's intellectual property, or
- that this document is suitable to any particular user's circumstance; or
- assumes responsibility for any damages or other liability whatsoever (including any consequential damages, even if Project Coordinator or any representative of a signatory party of the 5GZORRO Project Consortium Agreement, has been advised of the possibility of such damages) resulting from your selection or use of this document or any information, apparatus, method, process, or similar item disclosed in this document.

5GZORRO has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871533. The content of this deliverable does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in the deliverable lies entirely with the author(s).

Table of Contents

List of Tables

List of Figures

Executive Summary

This document specifies the final design of the 5GZORRO platform modules dedicated to automated service management with security and trust, by detailing the high-level design concepts presented in deliverables D2.2 and D2.4 *(Design of the 5GZORRO Platform for Security & Trust*). Concretely, this document is designed as a self-contained document, which takes the basis of D4.1 so as to update the required software components based on new 5GZORRO architecture functionalities during the development and integration phases in WP3 and WP4. Therefore, one of the main objectives of this document is to consolidate a single source of information for the design of zero-touch service management with security and trust solutions. It includes a description of the design of each software component, the principal KPIs covered by each one, a set of main interfaces, and lastly, the information elements linked to each component.

The modules described in this deliverable encompass from a service-based perspective the complete functionality ecosystem required to perform automated service management and orchestration, including the necessary security and trust establishment, and leverage resource and service exchange in distributed multi-party 5G scenarios.

Different technologies and enablers at multi-domain and single-domain are integrated together to provide the following main services:

- Inter-domain trust management, encompassing the stakeholder trust chain through an automated trustworthiness computation technique, based on verifiable reputation records, and the application of trusted execution environments, ensuring offloaded computation tasks.
- Intra- and inter-domain security management, including both secure connections and threat/attack detection at the business level.
- Automated network slice and service optimization based on intelligent orchestration and service mesh management.
- Automated 3rd party resource planning, optimizing when and how to manage external resources and how to deploy the available virtual resources.

To successfully provide the aforementioned services, numerous improvements are required in the current solutions for NFV management and orchestration (MANO), network slicing, security, trust, etc. which are also included in the descriptions and architectures presented as part of this design document.

This final deliverable contains all details and critical technical aspects for the implementation of the software modules related to Zero-Touch service management including security and trust. In that sense, the document at hand details, for each module, its context, the 5GZORRO specific enhancements in the area, the final design details related to the KPIs, and the module APIs/Interfaces. Moreover, the information models are presented, one for each module.

1 Introduction

New services have emerged with 5G which are directly focused on network stakeholders, such as trading computing resources (i.e., selling or renting), segmenting and distributing the entity resources in different domains. In this environment, the automated management of the services, with minimal human intervention, also known as zero-touch management, has become an essential requirement to ensure the proper functioning of these services, enabling real-time responses to possible incidents or scalability needs.

However, as the resources that form a 5G-enabled service are based on logically segmented and geographically distributed virtualized infrastructures, zero-touch management demands new solutions capable of accurately control these network resources into an end-to-end service with identical performance as if resources were physically deployed within the stakeholder's domain.

In addition, along with the growth of the network, its performance and the number of services offered, there is also an enormous amount of security threats to be covered, both internal to the stakeholder domain and in the communications between resources deployed in different domains.

Another aspect to consider in the relationships among different stakeholders in the 5G scenario is the trust links that may exist among them. This is a critical aspect that can determine whether the business is successful or not. However, trust is something subjective, based on previous information and experiences among stakeholders, including indirect relationships. Thus, trust management becomes very complex in environments with numerous entities involved, requiring new solutions in this area that are adapted to the relationships underpinned by 5G services.

Security and trust solutions need to be integrated with the lifecycle management of the network environment. However, due to the already mentioned distributed nature of the 5G services, the service and resource management processes will also require several changes. These changes bring the need for modern slice and service management solutions, enabled by AI-based automation.

In this context, 5G network services need to make use of the latest technologies including artificial intelligence, to offer the capabilities to cover the following needs:

- *Security and Trust Orchestration*, providing automated trust chain establishment in multi-domain scenarios and zero-touch security management to guarantee secure connectivity between 5G domains. This security and trust orchestration must be integrated with the management of services, so that its application is automatic and complete throughout the lifecycle of the resources and services deployed.
- *Intelligent and Automated Slice and Service Management* enabling automated resource and service composition in 5G multi-stakeholder networks and integrating the latest AI-based solutions for auto-scaling and flexible management.
- *Enhanced MANO and slicing solutions*, providing the baseline technologies required for the automated network component management, including VNFs, network slices, and any other necessary element.

Figure 1-1 depicts the 5GZORRO High Level reference architecture in which four major logical sub-systems, grouping different types of functionalities according to previously defined service-centric architectural model principles, can be visualized. Specially, the deliverable 4.4 is focused on the final design of Security and Trust, Zero Touch Management and Orchestration and Analytics & Intelligence for AIOps sub-systems. By means of these components, the 5GZORRO core platform implements the three capabilities mentioned above, in order to derive an intelligent zero-touch solution to orchestrate the trustworthy establishment of secure services in distributed multi-party 5G scenarios.

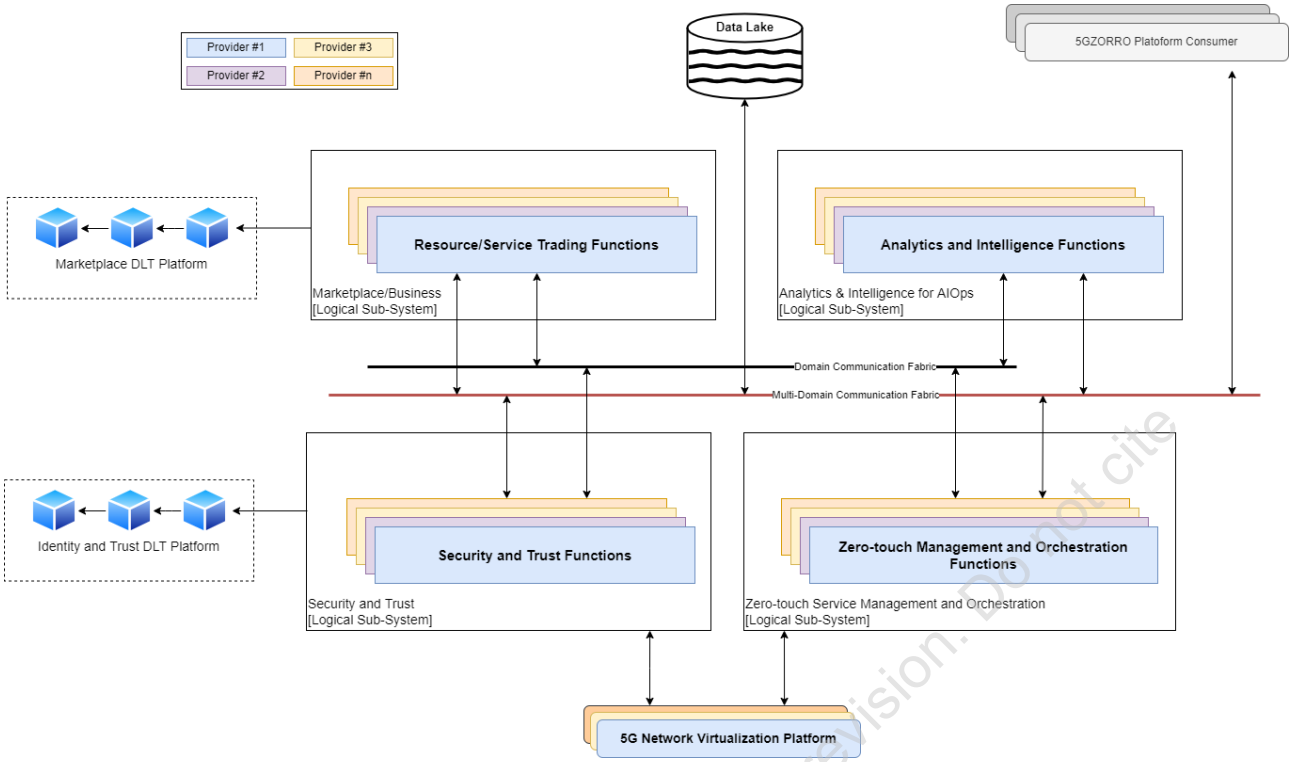

Figure 1-1: 5GZORRO High Level reference architecture

It should be pointed out that this deliverable takes the content of D4.1 and D4.2 as a basic and updates the design of the 5GZORRO core platform components. In this sense, this deliverable describes new functionalities in the platform components, details of the previously described component architectures, as well as new interfaces or any modification of them after starting the development and integration phases. In addition to that, this deliverable also includes some adjustments with respect to the information elements previously presented in D4.1, which had their origin in feedback from the software component implementation and experimental validation activities. As a result, this document is intended to be the final outcome of the 5GZORRO design of zero-touch service management with security and trust, and in consequence, includes all those updates to the software module components of the 5GZORRO architecture. In practice, this means that only part of this document presents new text and content, while some specific parts are imported as-is from D4.1 for the sake of completeness of this deliverable.

Especially, most of the updated and new content is included in Section 2 and 4, for the description of final capabilities and interfaces provided by their software modules belonging. In the case of Section 3, it introduces multiple design updates with respect to Intelligent and Automated Slice and Service Management Optimizer (ISSM-O), ISSM Mobile Edge Computing Manager (ISSM-MEC) and ISSM Workflow Manager (ISSM-WMF) being the latter proposed a per-domain service in this new deliverable. Finally, Section 5 is also updated since some of the information elements such as the 5G-enabled Trust and Reputation Management Framework, the Security Analysis Service, the Network Service Mesh Manager, or e-Licensing have been updated from the D4.1.

1.1 Document outline

This document is structured as follows:

- Section 2 illustrates the cross-domain security and trust orchestration service offered by the 5GZORRO architecture. The target of the 5GZORRO security and trust service is to supply novel mechanisms for orchestrating trust establishment and secure communications in multi-tenant and

multi-stakeholder environments. This section merges the content of the D4.1 and updates it for the four main components under the security and trust orchestration.

- Section 3 describes the 5GZORRO domain intelligent services by means of an automated slice management approach that utilises security and trust services to build cross-domain slices and services in distributed multi-party 5G scenarios. This section introduces relevant changes at design level such as the new per-domain approaches of ISSM-WFM and Cloud Native MEC platform, as well as merges the content of D4.1 highlighting slightly updates for ISSM-O and ISSM-MEC.
- Section 4 covers the development of NFV-MANO and network slicing enhancements to implement the 5GZORRO extensions. This section is also a merge of D4.1 and D4.2 (when it comes to any Resource Manager (xRM)) contents with important new updates in some of the software components such as the Network Service Mesh Manager or the e-Licensing, to name but a couple.
- Section 5 introduces technical information related to 5GZORRO functional entities and services deployed in previous sections. The 5GZORRO information elements are intended to integrate design and implementation decisions for the 5GZORRO ecosystem. This section merges the connection of the D4.1 but also updates several information elements due to new modifications appearing during the development and integration phases.
- Section 6 concludes the document with a table where a subset of objectives and KPIs are mapped to the sections of the document where they are addressed.

2 Security and Trust Orchestration

In a multi-domain and multi-stakeholder scenario, Security and Trust Orchestration are complex and modular tasks because of the number of fronts to be covered, ranging from the internal security deployment for each domain, the trust evaluation of the different stakeholders, the security of the communications among different domains and lastly the need for ensuring security when certain critical workloads go across different tenants and different stakeholders.

For these reasons, the 5GZORRO security and trust subsystem is divided into various modules according to their scope (see Figure 2-1). Concretely, the modules considered in this deliverable are:

- ***5G-enabled Trust and Reputation Management Framework (5G-TRMF)***, which manages the computation of trust values among different stakeholders based on previous experiences and the trust chain with other intermediary entities involved in the trust link. By means of this framework, end-to-end trustworthiness relationships can be established.
- ***Trusted Execution Environment*** (TEE) ***Security Management*****,** which orchestrates the triggering of Trusted Execution Environments for secure computation of critical tasks, assuring security, reliability and privacy-preserving to the actions deployed within this environment.
- ***Security Analysis Service*** **(SAS)**, previously called Intra-domain security at the business level, which is in charge of detecting and mitigating possible vulnerabilities and attacks inside the network of each stakeholder, enhancing internal security for resources and services.
- ***VPN-as-a-Service*** **(VPNaaS),** previously cited as Inter-domain security establishment at the communication level, manages the establishment of secure and trusted connections between different domains in the 5GZORRO environment, guaranteeing privacy and integrity properties without sacrificing performance.

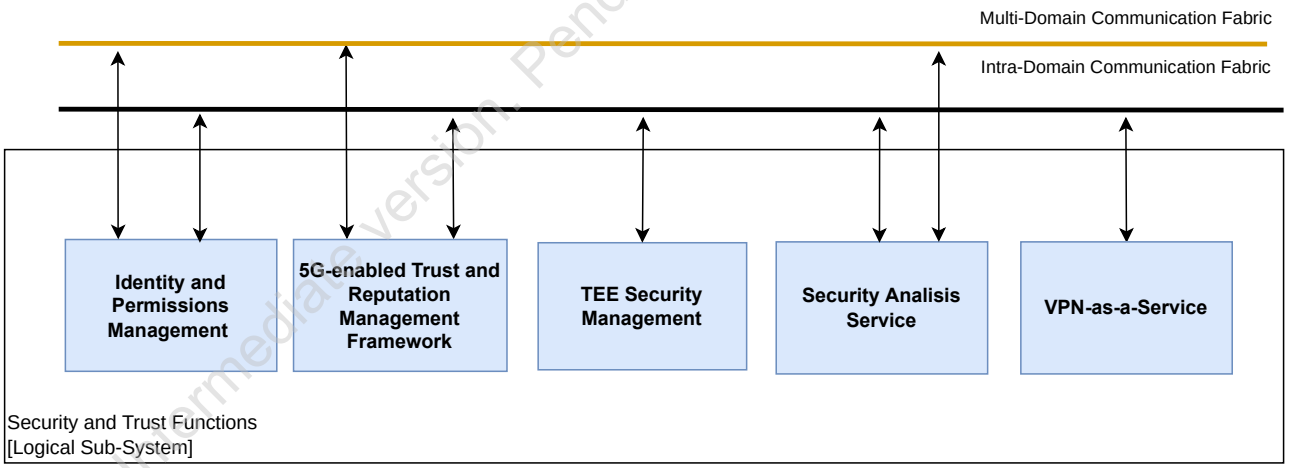

Figure 2-1: Security and Trust Orchestration modules

These modules act as the tools providing the needed security and trust functionalities to other orchestration and management components of the 5GZORRO platform. In this sense, security and trust orchestration acts as an enabler for the proper platform functioning, guaranteeing a high reliability. Next section depicts in depth the enhancements and design updates of the Security and Trust Orchestration modules.

2.1 5G-enabled Trust and Reputation Management Framework

This module covers the design of the trust and reputation management framework required to integrate end-to-end trustworthiness establishment for distributed stakeholder environments such as the one constituting the 5GZORRO ecosystem. Trust is a key element when defining and establishing business relationships between two or more partners. Trust, or the lack of it, influences the selection of partners for business relations, the way they are carried out and under which conditions. Therefore, trust is very important when trading (purchasing or selling) resources and services, allocated in a third-party infrastructure, one of the main 5GZORRO features.

2.1.1 Design Updates

In the 5GZORRO platform, the main functionality of this framework is to manage the entire lifecycle of the trust evaluation of the relationships among different 5GZORRO entities. The actions performed by the 5G-TRMF include:

- deciding the information sources to be used by the algorithms when gathering trust information,
- trust assessment,
- trust assessment report,
- constant update of trust values.

Additionally, this module also manages the indirect trust derived from the relationships between two external stakeholders. Therefore, this framework allows the evaluation of a trust level between any two different stakeholders based on their shared business relationships (modelled by means of SLAs and Smart Contracts) and previous interactions. The framework has to be suitable for making decisions about what resource or service providers are the most adequate for the establishment of an end-to-end relationship.

As commented before, this module has a decentralized nature, unlike most of the existing trust management frameworks, mainly relying on centralized scenarios. Thus, one of its enhancements is the possibility of being deployed both on intra-domain and inter-domain scenarios. Related to its decentralized nature, the trust and reputation management framework allows for end-to-end enforcement differentiating it from other approaches that only assess the trust level for a particular segment of the network.

Due to the fact that automatization is a core design principle of the 5GZORRO platform, the 5G-TRMF module is compatible with 5GZORRO services, such as Intelligent SLA Monitoring and Breach Predictors, whose design is based on the ETSI ZSM architecture [1]. The module approaches zero-touch functionalities by means of multiple intra- and inter-domain policies, rules, triggers, and intelligent techniques that ensure an automatization for all the steps and modules, whereas enables its adaptation to the current scenario requirements or needs.

Similarly, the 5G-TRMF module is also inspired by NIST's Zero Trust Architecture [2] and it contemplates some essential zero trust principles, starting with the basic one of no implicit trust granted to any entity, regardless of whether it is intra- or inter-domain. Such is the relevance of this critical principle that it is also being highly debated by European R&D projects many of which under the 5GPPP framework [3].

The trust and reputation management framework plays a key role in fulfilling the following project KPI:

> *Provide mechanisms for zero-touch trust automation in multi-domain scenarios on top of a 5G service management framework. The target for this KPI is: "The 5GZORRO system MUST cover up to 4 different stakeholders as part of the automated trust establishment process and to enable its automatic renegotiation when a stakeholder is joining or leaving the trust link."*

In order to deliver the aforementioned capabilities, Figure 2-2 renders the four generic phases of the 5GZORRO trust and reputation management framework. Furthermore, Figure 2-2 also depicts the main

modules and interfaces that the 5G-TRMF contemplates in order to ensure cross-domain trust relationships. On the left side of the diagram below (see Figure 2-2), we can see a set of interfaces the framework gets information through, and, on the right side, we can see the interface used for making available the information on the evaluation results.

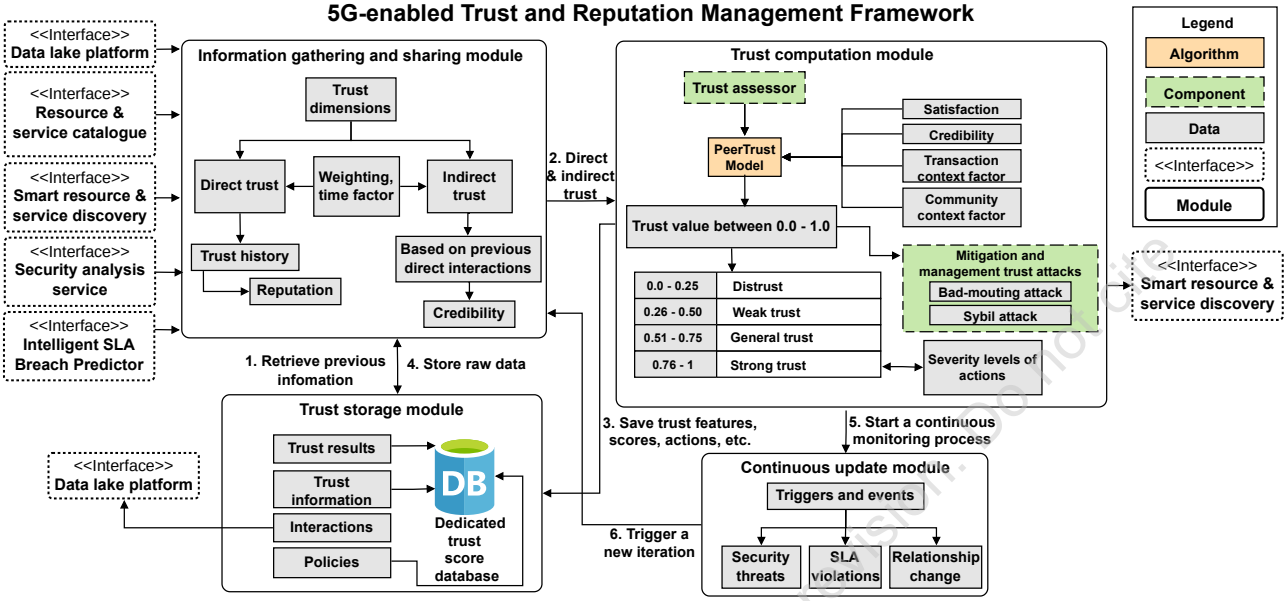

Figure 2-2: 5G-TRMF module architecture

The first phase is about the gathering of *trust information* from a set of trusted *sources*. This is an initial step of trust models and one of the most relevant, since both the trusted information sources selected and the statements acquired from them will subsequently determine the credibility and effectiveness of the assigned trust scores. Hence, the trust and reputation management framework should start by figuring out what are the available sources. Considering the 5GZORRO platform, there are three main sources for trust statement recollection, depicted at the left side in Figure 2-2, as input to the 5G-TRMF module, from which to derive and infer features: the Data lake platform, the resource and service catalogue, and the smart resource and service discovery. Note that, the trust and reputation management service interface is not contemplated as a data recollection source, since it contains the available interfaces that allow a stakeholder to interact with the 5G-TRMF. For instance, a stakeholder is able to enable or disable continuous data collection for a specific third-party. Once information sources have been determined, the 5G-TRMF will generate from this information a set of trust scores.

As can be observed in Figure 2-2 and Figure 2-3, the 5G-TRMF considers both direct and indirect trust. On one side, direct trust is information that can be collected from direct interactions (trust history) with the entity that is intended to assess the trust level. Through trust history, a stakeholder can create a reputation on another stakeholder, i.e., to gather different previous trust assessments with the same entity in order to have an estimate or reference of how a future trust relationship would be. In this vein, we consider objective properties to compute direct trust, such as SLA attributes or QoS properties, in order to avoid subjectivity problems. On the other side, indirect trust is acquired from trust relationships that an intermediate entity (recommender), which is not directly involved in the current trust establishment, has on a trustee. However, the trustor must have previous relationships with the intermediate entity. Since this information source contemplates feedback from other entities, it is necessary to determine how trustworthy the reply is (its *credibility*). Therefore, the credibility will be a factor that affects the final value for indirect trust.

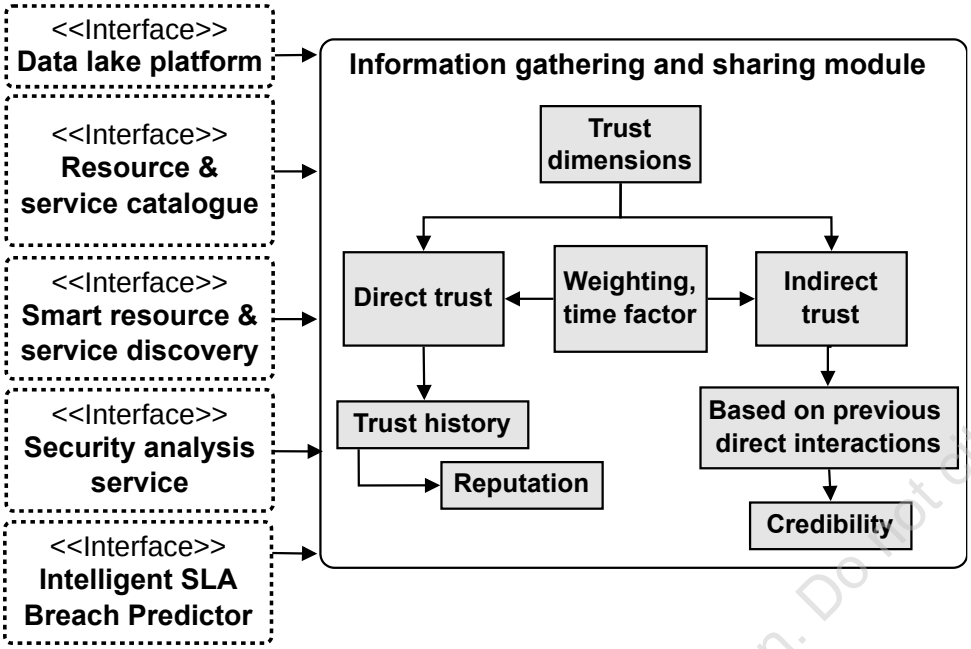

Figure 2-3: Information gathering and sharing module

After getting information based on direct and indirect trust, the next step is to evaluate a trust score for the entity. In this regard, the *trust computation* module plays an essential role (see Figure 2-4). In order to evaluate a trust score, this module contemplates as feasible techniques to be applied both a set of ML and DL algorithms [3] and trust models, even though the final selection of an ML/DL algorithm should be based on the determined features and scenario properties. As described in [3], when it comes to ML/DL algorithms, they should be privacy-aware since they are continuously analysing and managing users' data, and therefore, ML/DL techniques may reveal private personal information. Similarly, ML/DL algorithms should also be resilient against attacks, i.e., the adversarial attack, to avoid perturbations in the pre-processing or decision-making processes, as well as to be interpretable and explainable to enable trust. Figure 2-4 displays a decentralized trust model named PeerTrust which considers statistical measures (e.g., stakeholder's credibility and satisfaction) to forecast trust scores. In the current design, the 5GZORRO trust and reputation management framework leverages the PeerTrust model as the main technique to compute a trust score between two stakeholders, and in consequence, establish a trustworthy business relationship. No particular ML or DL techniques are being used as part of current PeerTrust design and implementation in 5GZORRO, as it is a model based on statistical measures. It follows a decentralized approach and considers both user satisfaction and credibility. Other key points to choose PeerTrust are its robustness and transitivity, as well as considering an adaptive time window-based approach and the presence of a reward-punishment mechanism. In addition, the *trust computation* module should withstand common trust attacks such as bad-mouthing attack (dishonest recommendations) and Sybil attack (multiple identities, associated with the same entity, increasing/decreasing reputation).

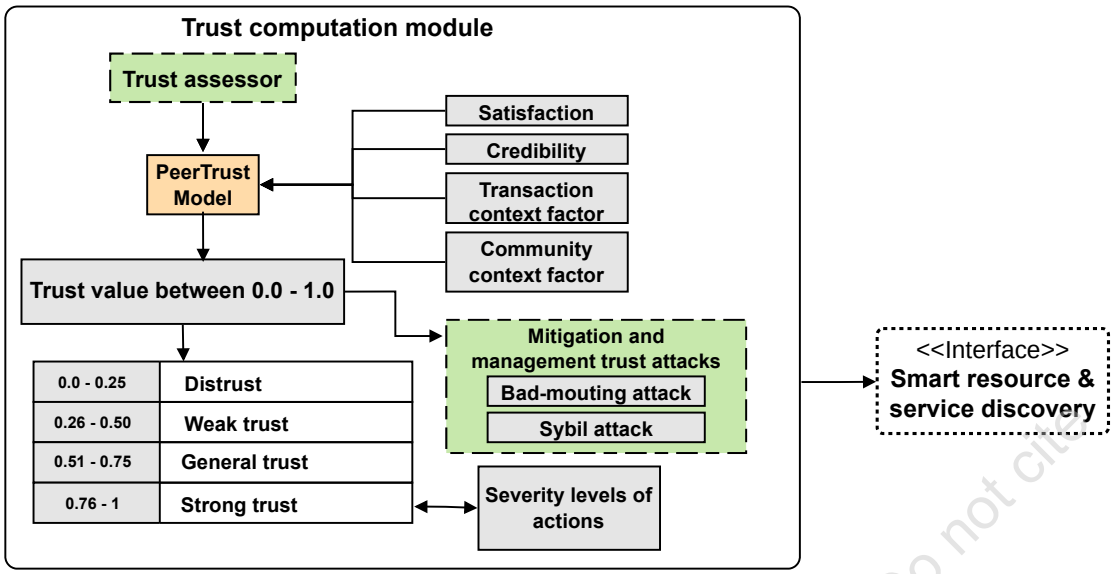

Figure 2-4: Trust computation module

Similarly, another conventional problem related to trust models, and in particular to indirect trust or recommendations, is the subjectivity problem. In order to evade such problem, the 5GZORRO trust and reputation management framework ought to utilise percentiles as a measure for a recommender to provide insight into an objective. Thus, they do not use an absolute value but a relative quantity to estimate recommended trust. Hence, a percentile value indicates the recommender perception of a target in relation to the other recommendations that the recommender has rated in the past. Finally, the 5G-TRMF should also consider task sensitivity in order to provide a final score according to the type of action and conditions that will be carried out.

Once a trust score has been calculated, next step is about recording *trust results and evidence storage* (see Figure 2-5). Since trust is not a one-time process, but it is a long-term one, it is crucial to keep the track over time. For the 5GZORRO ecosystem, two storage sources are mainly contemplated, either data lakes or a dedicated trust score database, which will store the output of trust assessment module. The type of storage source will be determined based on information. In the case of a stakeholder wanting to store sensitive information, the *trust results and evidence storage* module will make use of the trust score database. This storage source is a local repository where only stakeholders of the domain can access the information. Furthermore, this repository can also record intra- and inter-domain policies and rules that the 5G-TRMF may utilise to make decisions. In the case of non-sensitive information, it will be stored on 5GZORRO Data lake since this could be shared with other stakeholders participating in the 5GZORRO platform.

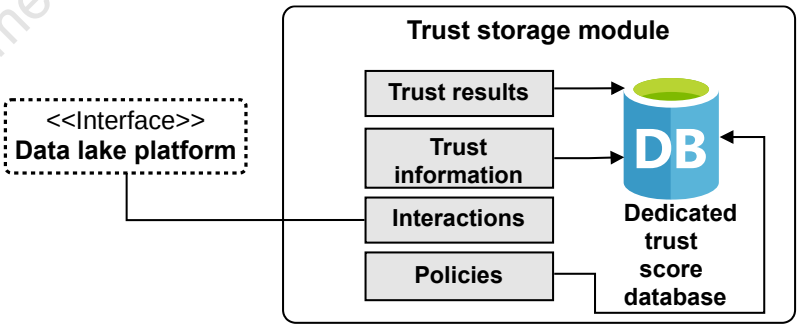

Figure 2-5: Trust storage module

Last but not least, the 5G-TRMF contains the *trust level update module* (see Figure 2-6). Since trust is a dynamic concept, which is modified over time, it is paramount to identify a set of triggers and events that enable to update the current trust scores. By means of this module, the trust and reputation management

framework is able to continuously update scores, at the same time it fulfils the zero-touch approach. Thus, this module utilises intra- and inter-domain policies and rules recorded by the previous module, whereas it enables large-scale adaptive systems to dynamically change their behaviour in response to changing environments or requirements. Similarly, security threats, SLA violations, and changes in relationships are also contemplated in order to recalculate the trust scores of an active relationship through two reward and punishment mechanisms. In the case of security threats, these are inferred from the Security Analysis Service after establishing a trust relationship. By means of the Zeek tool [13], the SAS component is able to monitor multiple traffic network events, such as the tracking of general information regarding TCP, UDP and ICMP traffic, unusual or exceptional activities related to malformed connections or misconfigured hardware, among others. Such events are gathered during sliding time windows, and afterward, they are analysed by the 5G-TRMF to increase or decrease consequently a trust score. In a similar way, the SLA violations are inferred from the Intelligent SLA Breach Predictor module, and in consequence, the trust score is updated based on how possible violations have been managed.

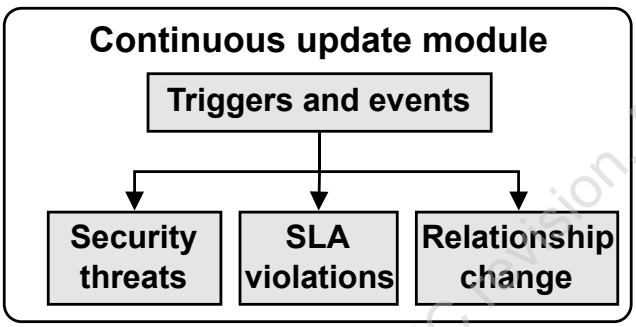

Figure 2-6: Continuous update module

To support this, the trust and reputation management framework service will provide the necessary APIs to enact all trust actions. Table 2-1 and Table 2-3 depict a general view of the main capabilities supported by the 5G-TRMF service, as well as its level of support (i.e., M-mandatory, O-Optional). Table 2-2 and **Error! Reference source not found.** provide more detailed information on the above capabilities introduced in Table 2-1 and Table 2-3, in the form of service interfaces.

Table 2-1: Definition of Trust and Reputation Management Framework service (per-domain level)

Service name**: Per-domain Trust Management**		Type**: *Per-domain***
Capabilities	**Support (O|M)**	**Description**
Start Data Collection	**M**	This capability is responsible for handling the cold start of the Trust and Reputation Management Framework
Stop Trust Relationship	**M**	This capability disables the process of collecting trust information about a stakeholder as well as a relationship with it.
Request Trust Scores	**M**	This capability sends a list of offers to the 5G-TRMF received by the Smart Resource and Service Discovery.
Gather Information	**M**	This capability activates the process of collecting trust information about a stakeholder from data sources.
Compute Trust Level	**M**	This method calculates trust level of the stakeholder to some internal resource using the previously acquired parameters.
Store Trust Level	**M**	This capability enables to records the previously calculated trust level and the utilized information in the available storage sources.
Update Trust Level	**M**	This capability enables to update a trust score of an ongoing relationship-based time-driven and event-driven mechanisms.

Notify Final Selection	**M**	This method is employed by the ISSM-WFM to notify the 5G-TRMFthe final resource or service/slice selected and on which the Gather Information service will be launched internally.
Notes		
none		

Table 2-2: Definition of Trust and Reputation Management Framework service interfaces (domain level)

Operation name: **startDataCollection**		
Description	This method is responsible for starting continuous collection data (e.g., Monitoring Analytics, Security Management Service, etc.) from the Data Lake platform.	
Input Parameters	**Type**	**Description**
trustorDID	String	Stakeholder's DID of who launches the process of continuous information recollection.
trusteeDID	String	Stakeholder's DID on which it launches the process of continuous information recollection.
offerDID	String	Offer's DID on which it launches the process of continuous information recollection.
Output Parameters	**Type**	**Description**
data_collection_status	Boolean	It returns true or false if the data collection cycle is active or not.
Notes:		

Operation name: **stopTrustRelationship**		
Description	This method is responsible for finishing an ongoing trust relationship.	
Input Parameters	**Type**	**Description**
offerDID	String	Distributed identifier of the final selected resource or service/slice to be finished.
Output Parameters	**Type**	**Description**
status	Boolean	It returns true or false if the relationship is stopped properly or not.
Notes:		

Operation name: **requestTrustScores**		
Description	This method is employed by the Smart Service and Resource Discovery application in order to evaluate trust scores of a list of pre-classified offers.	
Input Parameters	**Type**	**Description**
trustorDID	String	Stakeholder's DID of who launches the process of analysing Product Offerings.
offer_list	List	Set of product offerings to be analysed.
Output Parameters	**Type**	**Description**
trust_parameters	Dictionary	Dictionary with paramount data to calculate trust level.
Notes:		
More detailed information about the dictionary can be found in section 5.1 (trust and reputation management framework information model).		

Operation name: **gatherInformation**

Description	This method is responsible for acquiring trust information (previously collected), from Data Lake Platform, which will be used to derive trust parameters.	
Input Parameters	**Type**	**Description**
stakeholder_did	String	Stakeholder's DID on whom the previously collected data is to be recovered.
offerDID	String	Distributed identifier of the product offering on which it collects information.
Output Parameters	**Type**	**Description**
trust_parameters	Dictionary	Dictionary with paramount data to calculate trust level.
Notes:		
More detailed information about the dictionary can be found in section 5.1 (trust and reputation management framework information model).		

Operation name: **computeTrustLevel**		
Description	This method allows calculating a trust level score from previous data collected. Then, this value will be used to determine the most feasible stakeholder with which to establish a connection.	
Input Parameters	**Type**	**Description**
stakeholder_did	String	Stakeholder's DID.
trust_parameters	Dictionary	Dictionary with paramount data to calculate trust level.
Output Parameters	**Type**	**Description**
result	Double	Trust level previously calculated.
Notes		
Operation name: **storeTrustLevel**		
Description	This method allows the storage of trustworthy information in any of the available destinations.	
Input Parameters	**Type**	**Description**
stakeholder_did	String	Stakeholder's DID that wants to store the data for future trust computations.
trust_information	Dictionary	Dictionary containing relevant data on the trust level and confidence parameters used.
Output Parameters	**Type**	**Description**
result	Boolean	It indicates if the operation has been completed successfully or not.
Notes		

Operation name: **updateTrustLevel**		
Description	This method allows continuously updating an established trust score between two stakeholders.	
Input Parameters	**Type**	**Description**
trust_information	Dictionary	Dictionary containing relevant data on the trust level and confidence parameters used.
Output Parameters	**Type**	**Description**
result	Boolean	It indicates if the operation has been completed successfully or not.
Notes		

Operation name: **notifyFinalSelection**		
Description	This method is employed by the ISSM-WFM to notify the 5G-TRMF the final resource or service/slice selected.	
Input Parameters	**Type**	**Description**
offerDID	String	Distributed identifier of the selected product offering.
Output Parameters	**Type**	**Description**
result	Boolean	It indicates if the operation has been completed successfully or not.
Notes		

Table 2-3: Definition of Trust and Reputation Management Framework service (cross-domain level)

Service name**: Cross-domain Trust and Reputation Management**		Type**: Cross-domain**
Capabilities	**Support (O|M)**	**Description**
Query Trust Level	**M**	This capability enables to request the current trust level of a particular resource.
Notes		
none		

Table 2-4: Definition of Trust and Reputation Management Framework service interfaces (cross-domain level)

Operation name: **queryTrustLevel**		
Description	This method requests the last trust score of a particular stakeholder, if there is no previous value or it was calculated too much time ago, it triggers the trust level assessment process.	
Input Parameters	**Type**	**Description**
trustorDID	String	Stakeholder's DID of who launches the process of requesting a trust level.
truteeDID	String	Stakeholder's DID from whom to retrieve trust info.
offerDID	String	Distributed identifier of the product offering on which it requests a trust value.
Output Parameters	**Type**	**Description**
result	Double	Trust level previously computed.
Notes		

2.2 Trusted Execution Environment Security Management

The Trusted Execution Environment Security Management module provides functionalities that allow 5GZORRO to protect their tenant service or application running in a computing node against a stakeholder with malicious intentions.

For that purpose, this module will integrate commercial TEEs (Trusted Execution Environments) in the execution of some 5GZORRO software components, enhancing the security and trust of the software executed under these capabilities.

TEE-based software execution enables critical workloads to go across different tenants and different stakeholders with no losses in security, such as VIMs in a third-party infrastructure. By implementing an API for "TEE-as-a-Service", abstracting the low-level details of the commercial TEEs available, any service or application can be executed in a secure enclave on the 5GZORRO platform.

Since a TEE includes a zero-trust hardware platform, it is a key component to establish a root-of-trust and end-to-end secure communications. Thus, the presence of these capabilities in the infrastructure offered by some stakeholders also improves the trust level perceived by service consumers, as this component provides extra security at the hardware level to the resources and services offered by service providers.

It is crucial that the 5GZORRO security management solution not only protects data and software while they are running on the secure enclave, but also while data is in transit and at rest. Therefore, while multiple vendors provide different solutions to TEE (TrustZone for ARM, SGX for Intel, PSP for AMD and MultiZone Security for RISC-V), a secure enclave, by itself, is only part of the solution to achieve a complete application-oriented security solution. By definition, the data and the application should be inaccessible to the stakeholder in all states: during runtime, at rest and in transit.

2.2.1 Design Evolution

In order to implement the aforementioned enhancements, the 5GZORRO project bounds to fulfil a set of specific objectives related to the project key technical requirements and KPIs.

For instance, the 5GZORRO project expects to support the integration of zero trust hardware platforms (TEE - Trusted Execution Environments) as a root of trust for the monitoring of information and the establishment of end-to-end secure communications enabling critical workloads to go across different tenants and different stakeholders. In this vein, during the initial phase of the design, 5GZORRO carried out research in order to discover commercial TEE platforms that would cover the requirements associated with its ecosystem, providing added value to the project. For this, 5GZORRO has deconstructed the concept of TEE platform into hardware and software and the specifications found in this deliverable (particularly this section and 5.2) already reflect the outcomes of these considerations and validated assumptions. Specifically, the TEE platforms to be used, adapted, and integrated into 5GZORRO can be understood as the following:

1. Abstract software API exposing, at a higher level, the capabilities offered by existing TEE platforms such as Intel SGX interfaces (x86) [7] and that of GlobalPlatform's TEE Client API, so that we have a common and single API for a subset of architectures.
2. X86 hardware TEE Platform with SGX capabilities – several hardware options were analysed, taking into account the interconnectivity between the testbeds, practicality and cost. Intel NUC was the selected platform to be used as an environment for TEE. This platform allows 5GZORRO to access a pool of compute resources whose underlying CPUs have built-in native SGX support.
3. Building on top of the aforementioned x86 hardware TEE Platform, 5GZORRO deploys SCONE Confidential Computing – a platform that has been initially developed in the context of different H2020 Projects (mostly Sereca [4] and Secure Cloud [7], while others have been exploiting it and enhancing it [9]). More details on SCONE's functionalities can be found in section 5.2.

As enhancements to the current TEE ecosystem, 5GZORRO's TEE Security Management does not focus on the development of low-level TEE solutions, but on the integration of those with the 5GZORRO platform.

In this sense, 5GZORRO envisions to enhance the security and trustworthiness of the system by providing the execution of some of its core services and components in a TEE.

Mainly, two critical elements in the 5GZORRO platform are being integrated with TEE capabilities in order to improve their execution security, if needed. Those elements are:

- Monitoring Data Aggregation (MDA). As the MDA is responsible for processing and aggregating data, doing those operations in a TEE will provide the means to establish a high level of trust in that process, as SCONE will make the process tamper-resistant. In particular, when the data being processed is deemed trustworthy, we keep the same level of trust in the resulting processed data.
- Secure SLA Monitoring and Breach Prediction (as a Secure Oracle). Since SCONE also includes attestation mechanisms, it can ensure that sensitive operations, such as the SLA monitoring and breach predictor computations or authenticity proofs for smart contracts can run inside a tamper proof environment, where neither the off-chain data nor its computation can be tampered As it will sign the messages indicating a violation of an agreement, the signing of such message will be protected by the TEE itself, as the private key will never be exposed to outside actors (other processes running on the same machine), nor will it ever be in unprotected transit. Using an enclave for this particular matter would essentially allow for the bootstrap of a Secure Oracle – the off-chain entity which may interface with the Blockchain and feed secure and trusted data (which is critical, providing that the on-chain business logic must trust this information being provided, having in mind the associated business operations that will be automatically unlocked as a result of this). SCONE and its primitives (section 5.2) expose the methods used to build this solution, which is something 5GZORRO has focused on.

SLA Monitoring, Intelligent SLA Breach Predictor and Monitoring Data Aggregation (MDA) have been primarily selected to be running under a TEE, in order to assure the aggregation, processing/computation integrity of SLA monitoring data and to guarantee that the detection and prediction of SLA violations occur in a safe environment. Additionally, to establish a chain of trust for the metrics, the components involved in the origin of the data will also be considered as candidates to run under SCONE in a TEE environment.

The TEE Management service serves as a source of truth by providing the capability to attest the components running under a TEE. This functionality is made available by the underlying services provided by SCONE.

The following tables introduce the operations that can be supported by the TEE Security Management service.

Table 2-5: Definition of Trusted Execution Environment Security Management service (domain level)

Service name**: TEE Capabilities Management**		Type**: *Per-domain***
Capabilities	**Support (O|M)**	**Description**
Get TEE components	**O**	Retrieves list of object components instantiated on a TEE and their attestation information.
Get TEEs	**O**	This capability enables to acquire the current TEEs available in which code and service execution can be carried out.
Create TEE connection	**O**	This capability initializes and configures the connection with the available or selected TEE.
Execute command in TEE	**O**	This capability allows to execute commands in the TEE once a connection has been established.
Delete TEE connection	**O**	This capability deletes the connection with the TEE, deleting all the data left in the environment.
Notes		
*These functionalities are provided by SCONE set of APIs and may not be directly exposed by the TEE Management service.		

Table 2-6: Definition of Trusted Execution Environment Security Management service interfaces

Operation name: **getTEEComponents**

Description	This method consults the current platform components running under a TEE and returns their attestation information by checking if the measurement corresponds to an expected measurement value of the enclave.	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
result	Array	Array of component instances with information regarding their identifiers, names and attestation status
Notes		

Operation name: **getTEEs**		
Description	This method consults the current status among the TEEs configured on the platform to select the most appropriate option.	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
sessionResult	Dictionary	Dictionary of TEE's instances with information regarding their identifiers, names, status, type, etc.
Notes		

Operation name: **createTEESession**		
Description	This method initializes the required elements in order to generate and initialize a trusted connection with the chosen TEE.	
Input Parameters	**Type**	**Description**
context	String	It is a pointer to an initialized TEE Context.
session	String	An identifier in order to differentiate several sessions inside the same TEE.
connectionMethod	String	Type of identity credentials that the Client Application uses to determine access control permissions (e.g., DIDs plus VCs).
connectionData	String	Any necessary data required to support the connection method chosen (e.g., DID Document).
returnOrigin	String	A pointer to a variable which will contain the return origin.
Output Parameters	**Type**	**Description**
returnOrigin	String	Static variable providing information on successful or failed creation.
Notes		

Operation name: **executeTEESession**		
Description	This method executes the introduced command using the available TEE. The TEE should be already created.	
Input Parameters	**Type**	**Description**
context	String	It is a pointer to an initialized TEE Context.
session	String	An identifier in order to different several sessions inside the same TEE.

	commandID	Integer	An identifier used to indicate which of the exposed Trusted Application functions should be invoked.
	operation	Array	Set of instructions to be performed.
	returnOrigin	String	A pointer to a variable which will contain the return origin.
Output Parameters		**Type**	**Description**
	returnOrigin	String	It returns a static variable indicating if the command was correctly executed and its output.
Notes			

Operation name: **finalizeTEESession**		
Description	This method stops the connection between a client and a TEE.	
Input Parameters	**Type**	**Description**
context	String	It is a pointer to an initialized TEE Context.
session	String	An identifier in order to distinguish among sessions inside the same TEE.
Output Parameters	**Type**	**Description**
returnOrigin	Boolean	Returns true/false if the connection was correctly closed.
Notes		

2.3 Security Analysis Service

The Security Analysis Service (SAS) is the module that provide the security services in charge of performing network diagnostics to detect possible network vulnerabilities, attacks or threats for Network Services inside each domain, but also to apply the required countermeasures for the mitigation of these adverse events. The scope of this module covers an intra-domain perspective where each stakeholder deploys the services enabled by this module to enhance the internal resource and service security. Furthermore, the intra-domain security deployment in the internal organization infrastructure also enriches the external trust from other stakeholders, as these services can be seen as an additional security guarantee for possible delegated resources or services.

2.3.1 Design Updates

The SAS module applies security analysis mechanisms for the collection and monitoring of network service metrics in order to efficiently analyse the network traffic and detect possible network vulnerabilities or malicious behaviour. In addition, these diagnostics could be stored as aggregated reporting/statistics in order to be further provided to 5G-TRMF component.

Initially, regarding the detection procedure, this module identifies effectively network traffic vulnerabilities, malicious or unexpected activity, cyber-security threats or attacks and overall statistics regarding the network traffic of Network Services deployed/requested from resource providers/consumers. As for the network traffic, this can be mirrored to the Security Analytics VNF component through a virtual Tap (vTap) VNF which is responsible for connecting the different Network Services of a Network Slice and also mirroring the respective traffic. In fact, vTAP VNF uses a virtual TAP configuration [10] to capture a copy of the data flowing between the deployed Network Services of a certain domain. This happens for example through the presence of a virtual switch on the virtual links between the respective services. In addition, the aforementioned VNFs (Security Analytics, vTAP) are deployed alongside with the Network Slice requested from the stakeholders, as there are responsible for the Network Services hosted on the specific Slice.

Secondly, if security threats have been detected, the SAS, and especially a dedicated Security Analytics VNF module, is capable of applying countermeasures and mitigation procedures by both performing a set of traffic rules and configuring the necessary firewall policies. Finally, the obtained statistics and diagnostics from the SAS are further stored as an aggregated report and provided to 5G-TRMF component. As a result of the process, the 5GZORRO consumers can benefit from continuous operation and reliability on the Network Services requested.

In order to monitor the network traffic from the 5GZORRO platform consumers, i.e., the 5G core network or mobile edge infrastructures, the Security Analysis Service is deployed as a set of Security VNFs (vTAP, Security Analytics & ELK) in conjunction with the Network Services VNFs requested from 5GZORRO platform consumers in the same Network Slice descriptor according to MANO specification (see section 4.2). In addition, the Security Analytics VNF can be activated to start analysing the network traffic of the Network Services via automated and on demand mechanisms. Also, the diagnostics obtained from the Security Analytics VNF are further send to the ELK VNF, which is responsible for storing and visualizing these statistics. Note that the vTAP and Security Analytics VNFs are deployed per slices, however, the ELK VNF is set up per domain, so the ELK VNF will store all the metrics provided by multiple networks slice under the same domain. For the deployment of Network Services, the ETSI OSM MANO [12] orchestration framework has been used as it also provides the necessary tools for performing the respective automated and on demand configurations (day-1 or day-2 configurations). The high-level architectural diagram of this module is shown in Figure 2-7.

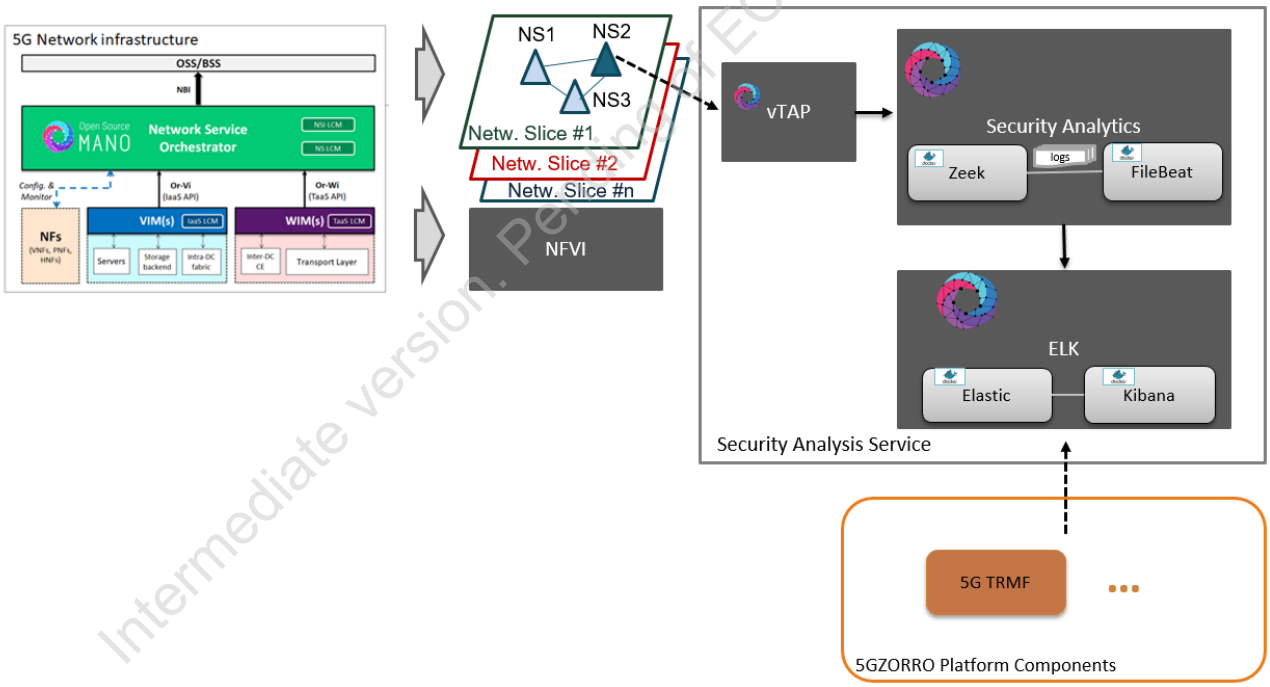

Figure 2-7: Security Analysis Service module architecture

The Security Analytics VNF component encapsulates the functionality and the support of the Zeek network security monitor [13], integrated with a FileBeat instance for data collection. In addition, this data can be further sent to the ELK VNF component that encapsulates both an Elasticsearch for storing the data and a Kibana for the visualization of the statistics [15]. The complete high-level integration encapsulated inside the respective VNFs is illustrated in Figure 2-8.

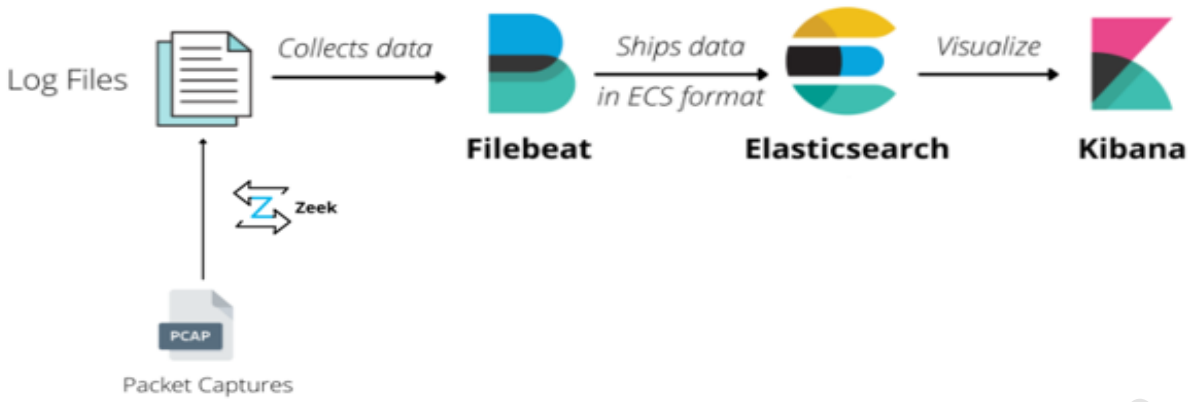

Figure 2-8: Security Analytics & ELK internal Design

Finally, these services are implemented as a set of on demand and automated functionalities in order to perform a variety of actions upon the Security Analysis Service module. These actions can be applied remotely by the user from the orchestration frameworks as on demand configurations in order to enable the Security Analysis Service functionalities and customize their behaviour. Regarding the behaviour and especially on how the security and network traffic analysis will be performed, an imperative approach was used as we apply a default configuration file for the network analysis performed by the Security Analytics component.

More details about the Security Analysis Service capabilities and interfaces are outlined in Table 2-7 and Table 2-8, respectively.

Table 2-7: Definition of Security Analysis Service services (per-domain level)

Service name**: Intra-domain security**		Type**: *Per-domain***
Capabilities	**Support (O|M)**	**Description**
Set network traffic mirroring	**M**	This capability sets the live mirroring of network traffic of different network services to an active monitoring interface through a virtual TAP VNF.
Set network traffic monitoring interface	**M**	This capability sets the monitoring interface for the collection and monitoring of incoming network traffic.
Start network analysis of the collected data	**M**	This capability allows to perform real-time security network analysis of the collected data in order to obtain diagnostics and store statistics.
Stop network analysis of the collected data	**M**	This capability disables the performance of real-time network analysis.
Notes		
none		

Table 2-8: Definition of Security Analysis service interfaces

Operation name: **setNetworkTrafficMirroring**		
Description	This method sets the live mirroring of network traffic of different network services to an active monitoring interface through a virtual TAP VNF.	
Input Parameters	**Type**	**Description**
ipsOfNetworkServices	String	The ips of Network Services that will be configured to mirror their incoming/outcoming network traffic to to an active monitoring interface through a virtual TAP VNF.
Output Parameters	**Type**	**Description**
none	N/A	N/A
Notes		

Operation name: **setNetworkTrafficMonitoringInterface**		
Description	This method sets the monitoring interface for the collection and monitoring of incoming network traffic.	
Input Parameters	**Type**	**Description**
interfaceName	String	The name of the network interface that real-time network data will be collected and analyzed
Output Parameters	**Type**	**Description**
none	N/A	N/A
Notes		

Operation name: **startNetworkAnalysisofCollectedData**		
Description	This method starts the network analysis of the Network Service's traffic	
Input Parameters	**Type**	**Description**
ipELK	String	The ip of ELK for storing, indexing and visualizing the obtain statistics
configFiles	Files	Files for the configuration of Zeek and FileBeat behavior
Output Parameters	**Type**	**Description**
status	String	The result (i.e., OK or failure) of starting the Security Analytics
containerID	String	The identifier of the running containers of Zeek and FileBeat.
logFiles	Files	A variety of network log files based on the analysis of the collected network traffic data.
Notes		

Operation name: **stopNetworkAnalysisofCollectedData**		
Description	This method stops the network analysis of the Network Service's traffic	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
status	Boolean	The result (i.e., OK or error) of stopping the Security Analytics
Notes		

2.4 VPN-as-a-Service

This module aims at providing the capabilities for establishing secure and trusted connections between different domains in the 5GZORRO environment, guaranteeing privacy and integrity but without sacrificing performance. It has an important role when it comes to performing network slicing and integrating resources located at a third-party infrastructure. These resources are purchased via the marketplace and integrated into the requester domain network.

The main idea behind this module is to use the cryptographic material linked with the DIDs, to derive shared keys between the elements located in different domains. After that, a VPN-type connection will be generated enabling the integration of these resources and services in the purchasing domain. The main novelty provided by 5GZORRO with the introduction of this module is the integration of DID information stored in a DLT with the generation of a secure connection at VPN level, integrating a resource physically located in an external domain with the rest of the infrastructure deployed in the client domain.

2.4.1 Design Updates

Figure 2-9 represents a secure cross-domain communication establishment. Normally, this context appears in the 5GZORRO ecosystem when an operator A detects a lack of capabilities in its own domain and decides to select certain resources/services available at the Marketplace in order to extend its current capabilities. In that sense, the diagram below depicts a real scenario where a secure connection between two operators is necessary.

First and foremost, Operator A detects that it is not able to cover the performance indicated in its SLA, and therefore, it selects additional services and signs a Smart Contract with Operator B. After that, the Network Service Mesh Manager (NSMM) is the entity in charge of providing an end-to-end multidomain slicing and service, guaranteeing privacy and security (see section 4.3). Thereby, a secure and private communication channel is required to transmit network traffic across domains. To this end, the NSMM leverages both the Identity and Permission Manager (Id&P) and the VPNaaS.

Since 5GZORRO leverages DIDs and Verifiable Credentials (VCs) for identification, authentication, and authorization, these may also be utilised as the asymmetric keys needed to derivate the symmetric key pair for a VPN tunnel. Therefore, the NSMMs request to each Id&P Agent a DID and a key pair which are afterward forwarded to the VPNaaS so as to be configured. It should be pointed out that the previous information is shared as an encoded payload, in consequence, only the appropriate VPNaaS would be able to obtain the final information. One of the most important steps of the initial configuration is the verification of the DID and public key shared by the other VPNaaS. In this regard, each VPNaaS must verify through its Id&P Agent the object received from the other end, that is the DID, the public key (pubKey), and the timestamp. After that, and only if the verification was successfully carried out, Operator A starts the shared key generation process. By means of Operator B's public key acquired from the configuration phase, Operator A forwards an authenticity proof to Operator B. If the answer is satisfactory, Operator A will generate and send a symmetric key to Operator B, which will subsequently be utilised to share information securely and confidentially. Eventually, the configuration process will be finished, and the VPN will be set up with the purchased resource.

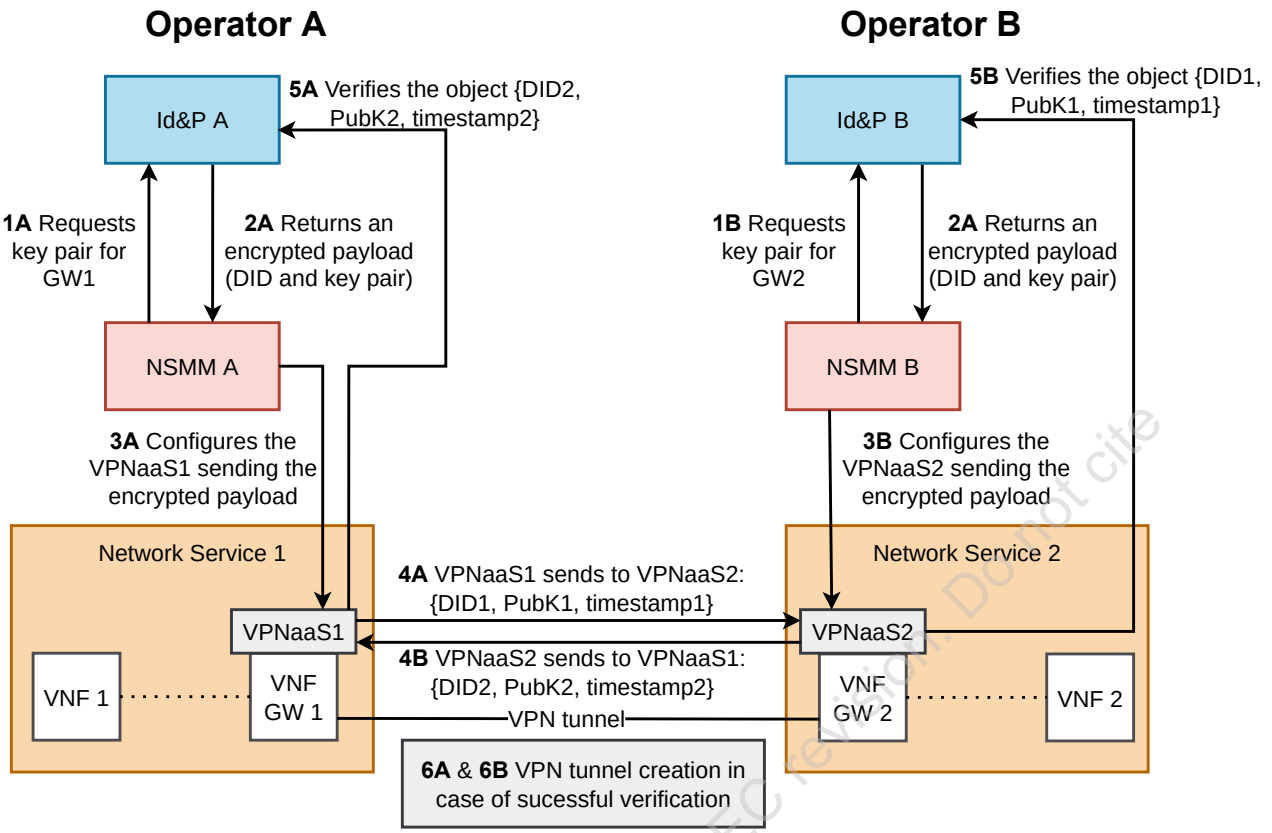

Figure 2-9: High-level VPNaaS architecture

To integrate both the VPN solution and DID-based identification, it is necessary that the VPN solution makes use of the API of the Identity Management module as the source where to obtain the keys and identifiers of the elements involved in the secure connection as well as to verify the keys.

Thus, it is critical for the correct integration that the format of the identifiers and keys used to generate the DIDs and VCs is compatible with the formats supported by the VPN. Once common formats are used in both elements, their integration is relatively straightforward, since the same public and private keys can be used to authenticate the entities and derive shared keys when generating the secure connection.

To support this, the VPNaaS module will provide the necessary APIs to deploy a Virtual Private Network which utilises DIDs as identification mechanism. In the case of the Table 2-9, this depicts a general view of the main capabilities supported by VPNaaS module, as well as its level of support (i.e., M-mandatory, O-Optional). The Table 2-10 provides a greater level of detail on the capabilities previously presented in the Table 2-9.

Table 2-9: Definition of VPNaaS service (per-domain/cross-domain level)

Service name: **VPN-as-a-Service**		Type: ***Per-domain & Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Launch	**M**	This capability is in charge of configuring the VM in which Wireguard will be instantiated.
Get Configuration	**M**	This capability enables to know, from client side, the necessary configuration to later establish a connection through a VPN.
Connect VPN	**M**	This capability enables to launch a secure communication between two stakeholders through a tunnel.
Disconnect VPN	**M**	This capability enables to finish a previous connection.

Notes

Table 2-10: Definition of VPNaaS service interfaces

Operation name: **launch**		
Description	This method allows downloading configuration options from server (client side).	
Input Parameters	**Type**	**Description**
ip_range	String	IP range to be used in the wg0 interface.
net_interface	String	Network interface to be used to forward VPN traffic.
port	Integer	Network port where the VPN listens for incoming connections (server port)
IdM_payload	String	Sensitive info such as private and public keys, timestamp, and DID.
endpoint_IdM	String	IdM Agent endpoint to verify key pair.
Output Parameters	**Type**	**Description**
result	Boolean	It will be 200 or 400 depending on whether the installation was completed satisfactorily or not.
Notes		

Operation name: **getConfiguration**		
Description	This method allows downloading configuration options from server (client side).	
Input Parameters	**Type**	**Description**
none		
Output Parameters	**Type**	**Description**
did	String	Public identifier associated with server stakeholder, which will be used to gather server's public key.
public_key	String	Public key linked to the DID, which will be used as an authenticity proof.
IP_range	String	IPs associated with the VPN server.
VPN_port	Integer	Port in which the VPNaaS is listening.
Notes		

Operation name: **connectVPN**		
Description	This method establishes a new client connection using DIDs as the authentication mechanism.	
Input Parameters	**Type**	**Description**
ip_address_server	String	It is an address where VPN server is running.
port_server	Integer	Specific port where VPN server will be available.
IP_range_to_redirect	String	Range of IP directions that will be redirected to that VPN peer.
destination_IP_range_to_redirect	String	IP of the VIM to be redirected the network traffic.
Output Parameters	**Type**	**Description**
result	Integer	It indicates if the tunnel has been established successfully or not.
Notes		

Operation name: **disconnectVPN**		
Description	This method carries out the completion of the established safe tunnel.	
Input Parameters	**Type**	**Description**
ip_address_server	String	It is an address where VPN server is running.
port_server	Integer	Specific port where VPN server will be available.
Output Parameters	**Type**	**Description**
result	Integer	It indicates if the tunnel has been disconnected successfully or not.
Notes		

3 Intelligent and Automated Slice & Service Management

The 5GZORRO Intelligent and Automated Slice and Service Management focuses on automation of managing secure cross-domain slices and services within them. ISSM is responsible for enforcing business transactions both at the system level by interacting with 5GZORRO MANO and Slicing extensions described in Section 4 and with alternative slicing technologies, as well as by managing business transaction context across the entire 5GZORRO platform allowing a principled, repeatable, auditable, and trustworthy interaction among the multiple components of the platform to realize a specific business flow.

The Intelligent and Automated Slice and Service Management is responsible for the optimization of resource allocation to inter-domain slices subject to SLAs and cost-efficiency targets. Two scenarios for slice optimization are considered: continuous and discreet time.

In summary, these capabilities are provided by an Intelligent and Automated Slice and Service Manager (ISSM) module that comprises three main components:

- ***ISSM Workflow Manager (ISSM-WFM)***: executes orchestration workflows in a context of a business transaction, such as extending a slice across a second domain in cooperation with the Network Slice and Service Orchestration (see Sec. 3.1).
- ***ISSM Optimizer (ISSM-O):*** optimizes cost-efficiency trade-off of network services and slices required to be created in a context of a specific business transaction and continuously optimizes services and slices that have been already set up in previous transaction flow executions.
- ***ISSM MEC Manager (ISSM-MEC)***: facilitates declarative cloud native style of managing applications executing in a MEC environment while collaboratively managing MEC infrastructure and MEC services at the host and system levels based on the intents communicated by an application control plane. One particular important emerging MEC environment explored in 5GZORRO, is Kubernetes. It is used as "smart" NFVI that does not require VIM and which can natively deploy CNFs and services managed by Operators acting as VNFMs in combination with Kubernetes native workflow engines.

Figure 3-1 depicts a high level ISSM architecture. Figure 3-2 shows a software architecture that backs the high-level design.

Personas

- A 5GZORRO Platform Participant: is an MNO that owns an account on the 5GZORRO platform and is eligible to request execution of business workflows, such as cross-domain slice establishment. In addition to triggering a business flow within ISSM, a 5GZORRO Platform Participant can inquire about the progress, pause and cancel the business flow. A 5GZORRO Platform Participant is not eligible to change a business flow. The business flows in ISSM are certified pre-coded flows that collectively form ISSM's workflow management and orchestration functionality.
- A 5GZORRO platform developer is eligible to develop new ISSM flows and update and delete the existing ones.
- A MEC application developer develops applications that will be executed with a slice, i.e., on a MEC platform that is deployed in a capacity of 3GPP Application Function (AF) for vis-a-vis 5G Core (control plane) and User Packet Function (UPF, data plane) of a slice. A MEC application developer is interested in a cloud-native environment in a MEC. In essence, she is not interested in knowing the details of MEC implementation or Telecommunication standards that govern MEC orchestration. Rather, a typical MEC application developer is interested in seeing cloud-native MEC as either an extension of a public cloud that she already uses or a Kubernetes environment, which over the last

few years became a de-facto environment for deploying and operating container based microservices. Cloud-native applications are not configured statically. They dynamically adapt to the geo-spatial and temporal workload distribution and dynamically acquire and release resources to match workload patterns. When running in MEC, an application cannot directly acquire resources, because the application is not exposed to the MEC internal structure and cannot directly access virtualization infrastructure such as Kubernetes. Rather, an application control plane declaratively specifies its intents to ISSM-MEC which in turn takes the needed orchestration actions (the MEC application control plane is shown in Figure 3-1 as an entity external to ISSM). Alternatively, modules such as Intelligent SLA Breach Predictor and/or SLA Monitoring can inform ISSM-WFM about pending or occurring SLA breach on behalf of the application and ISSM-WFM can formulate intents to ISSM-MEC to remediate the situation (e.g., by scaling out the application).

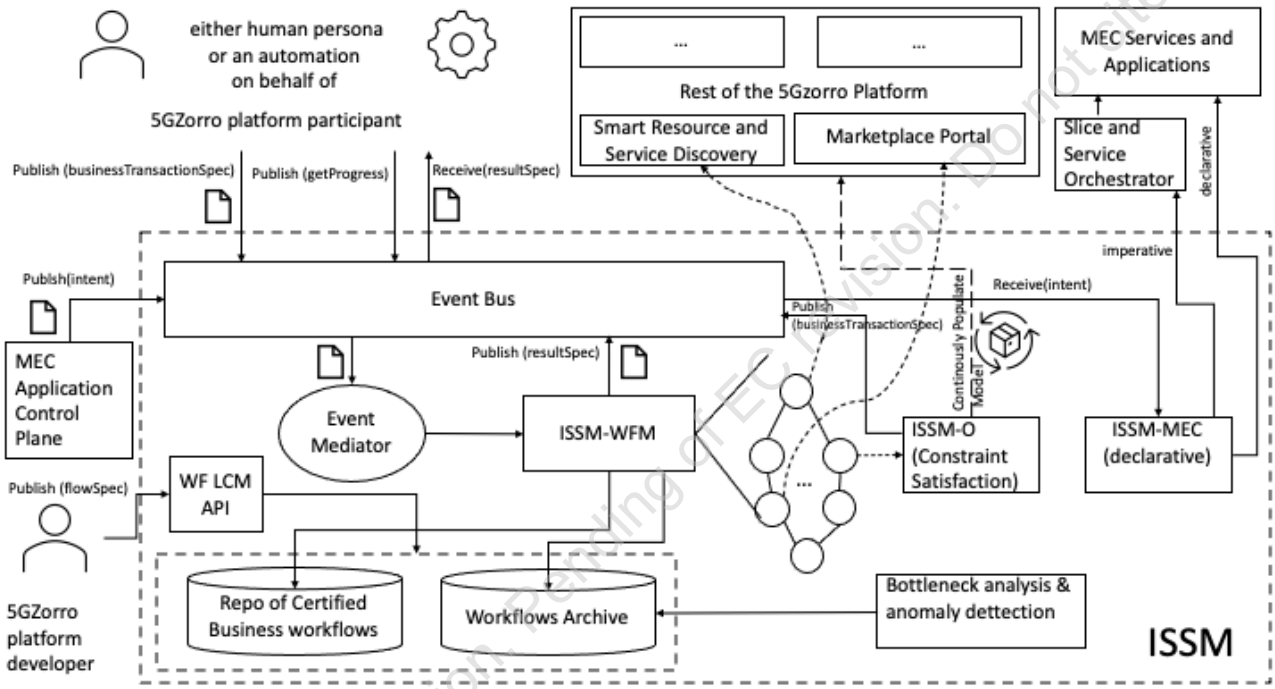

Figure 3-1: High Level Architecture of ISSM

Software Architecture

In Figure 3-2, we describe a software architecture that we explore in 5GZORRO to support the high-level design generic design of ISSM in Figure 3-1. Our main guiding principles for ISSM implementation include:

- Portability and cloud-nativeness: our main proposition is to create a software platform that will be portable to different cloud/edge/telco environments and approachable by all the 5GZORRO personas. Following the trend for cloud native convergence, we aim ISSM to be cloud-native by design.
- Scalability: we pursuit a system design that can provably scale to thousands of nodes and hundreds of clusters.
- Sustainability and impact: to implement the ISSM we select high traction projects with high likelihoods of industrial impact.
- Each constituent service of ISSM is intended to be self-contained and can be deployed in combination with other third-party services (e.g., different workflow management engines, backend optimizers or MEC systems).

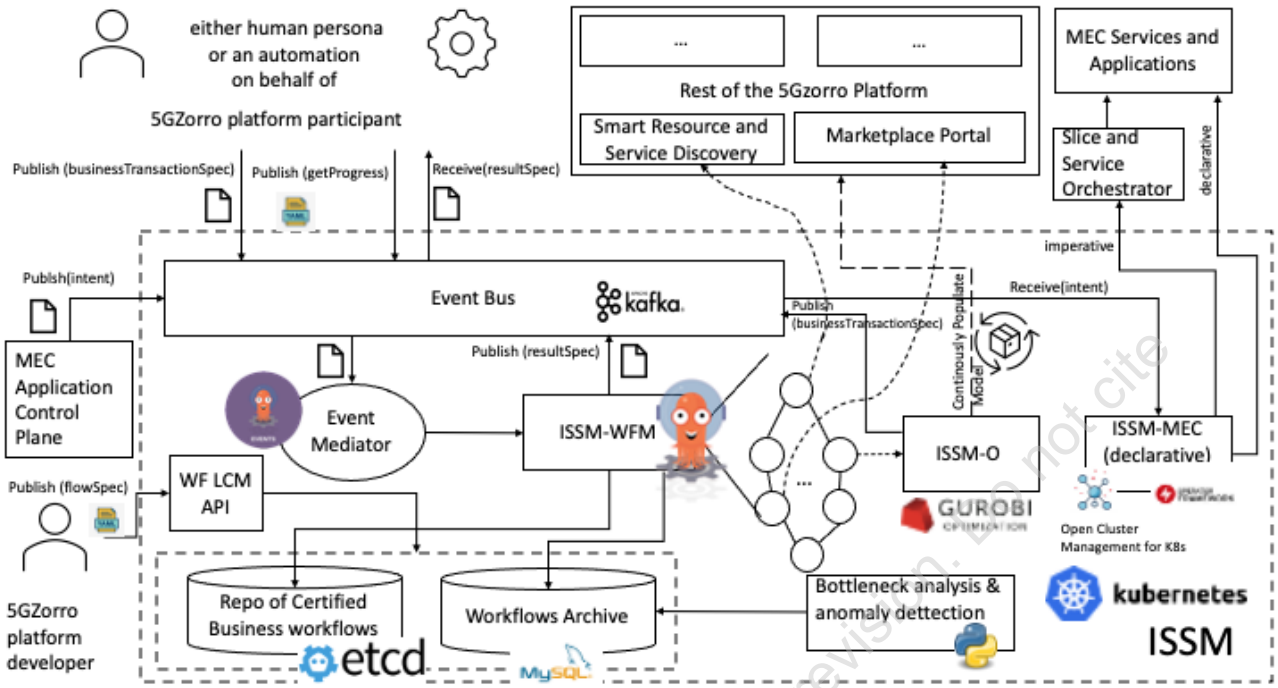

Figure 3-2: ISSM Software Architecture

At the base of the ISSM system there are several open-source projects and commercial products that have been selected for initial implementation in 5GZORRO. Please note that since cloud native landscape is extremely fluid, the below list is by no means final. Rather it comprises project with high *probability* of success and traction as perceived at the time of writing of this deliverable and continuous re-evaluation and agile adaptation will be performed to ensure that 5GZorro software deliverables will be relevant and appealing to developers when the project terminates.

3.1 ISSM-WFM

Figure 3-3 depicts the update high level design of ISSM-WFM. In the previous design, 5GZORRO Platform Participant persona (i.e., an MNO participant in the 5GZorro platform) published business transaction specification (a document) on a well-known topic of a centrally supported Event Bus and the centralized workflow engine executed the workflow spanning multiple components (as illustrated in the workflows of D2.4) corresponding to the intent.

In the current updated design, each ISSM-WFM workflow, e.g., a workflow that implements scale-out of a slice/service or a slice set-up across two MNOs starts locally in an MNO domain that initiates the operation. The initiation is triggered either via Portal or CLI or by receiving a notification from Intelligent SLA Breach Predictor or SLA Monitoring service.

This design evolution was planned as the second iteration in design and development to make ISSM-WFM fully aligned with the 5GZorro design philosophy after sufficient hands-on experience with the 5GZORROuse cases and the corresponding workflows was accumulated. The current design also follows up on the feedback that the project received at the mid-term review and reflects a suggestion by the reviewers to make ISSM-WFM more distributed and symmetric. The current design is backward compatible. No changes are required on the client side.

Each per-domain ISSM-WFM installation comprises Event Bus (which is being part of communication fabric), Event Mediator, Workflow Engine, and Workflow Repository. A cross-domain part of ISSM-WFM has the same components. This way ISSM-WFM design is fully symmetric. In addition, an optional component, Multi-cluster Manager, is installed in the cross-domain component of ISSM-WFM and Multi-cluster Agent is installed in each domain. The Multi-cluster Manager allows a logically centralized management of ISSM-WFM resources across domains. Multi-cluster agents subscribe to the Multi-cluster Manager channels and pull resources defined in the cross-domain ISSM-WFM component that are intended to these Multi-cluster Agents (e.g., workflows that should be distributed to the domains). The Multi-cluster Manager channels support group operations (i.e., the agents can be logically grouped, and a resource can be defined and distributed to the whole group by a single operation), which facilitate scalable management. We designate this component as optional because its functionality can be implemented in a minimalist way using Event Bus and Event Mediator. Yet, a richer functionality, such as bookkeeping, versioning, policies are required when managing distributed workflows' lifecycle.

A typical ISSM-WFM workflow would start in a specific domain (for the sake of discussion and without losing generality, let us assume that a workflow starts in Domain A triggered by a Requestor, which can be either Portal or some automated functionality, such as Intelligent SLA Breach Predictor, SLA Monitor, ISSM-O or any other future manual or automated component being part of zero touch slice management and orchestration cycle.

The business transaction specification is published on the Event Bus. An Event Mediator receives it from the bus and triggers an appropriate workflow that was previously onboarded to the Workflow Repository through Multi-Cluster Manager. The local orchestration workflow starts executing. The steps can span multiple components of the 5GZORRO platform (see Figure 3-1 for an illustration of the possible steps that previously executed as one centralized flow). At some step of the workflow running in Domain A, an orchestration sub-workflow might be required to be executed in Domain B. To that end, the workflow step in Domain A publishes a message (containing parameters and an entry point for a workflow in Domain B. Event Mediator of Domain B is subscribed on the topic dedicated to this domain in the cross-domain Event Bus. It receives the message and triggers an appropriate workflow in Domain B. At some step in the workflow of Domain B, the control might have to be passed back to Domain A. Possibly also status and variables must passed back. To that end, the workflow of Domain B publishes a message on the cross-domain Event Bus. This might trigger an optional "stitching workflow" in the cross-domain Workflow Engine, after which control will be passed to Domain A by publishing a message on a dedicated topic for Domain A. The Event Mediator of Domain A receives the message from the cross-domain bus and continues with the Domain A workflow. This way, control can be passed back and forth arbitrary number of times across different per-domain ISSM-WFM components with support for "stitching" and synchronization by the cross-domain ISSM-WFM component and arbitrarily complex business level orchestration flows can be developed by the 5GZORRO platform developer persona.

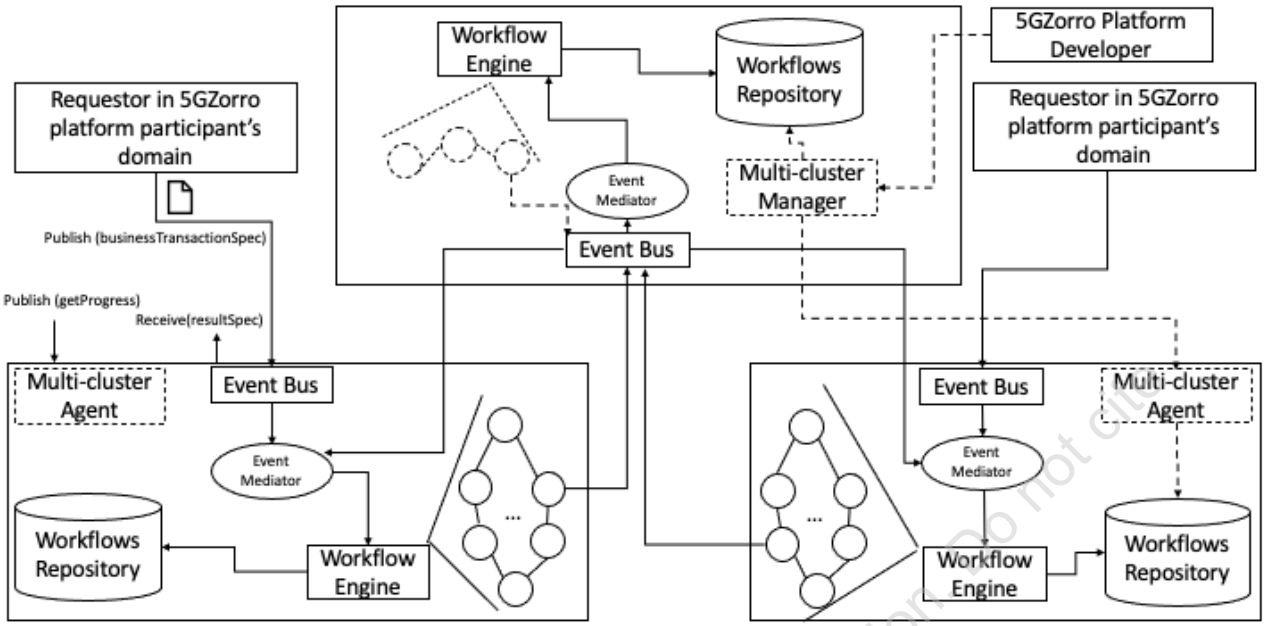

Figure 3-3: Updated ISSM-WFM design

ISSM-WFM triggers an appropriate workflow from the workflow repository. Workflow is a document that defines the Direct Acyclic Graph (DAG) of tasks that comprise the business workflow. In a specific business context, each task might reach out to different components of the 5GZORRO platform, such as Marketplace Portal, Slice & Service Orchestrator, Smart Resource and Service Discovery, Data Lake, etc. In the interest of conserving space and for the sake of simplicity, not all elements of the 5GZORRO platform are shown in the Figure 3-1 and Figure 3-2.

The 5GZORRO Platform Participant who triggered the workflow, can query its progress, pause it or cancel it at any time. A 5GZORRO Platform Participant sees only those workflows that belong to it. All steps of a workflow execution are memorized and archived for future bottleneck analysis and anomaly detection.

For example, to procure resources from the marketplace, ISSM workflows interact with Smart Resource and Service Discovery Service specifying criteria (constraints) for the resources. The resources information obtained via this interaction is used to populate a model that is fed to constraint satisfaction engine, ISSM-O. The goal of ISSM-O is to achieve the most cost-efficient slice or service resources allocation subject to constraints such as trust, security, and performance. Resources selected by ISSM-O are procured from the marketplace via the DLT mechanism. Since transactions are performed concurrently for multiple MNOs, this cycle of resources discovery, constraint satisfaction and procurement are potentially performed multiple times by a flow run by ISSM-WFM.

When workflow terminates either normally or abnormally, ISSM-WFM publishes a resultSpec document on a well-known topic of Event Bus to be consumed asynchronously by any of the involved personas in a specific business context of the workflow. To continue the above example, resource procurement workflow, after a series of resource discovery and optimization steps, results in a concrete declarative specification of service to be provisioned by the involved resource providers, e.g., network slices, network services, MEC systems, MEC application instances, etc. To accomplish this, resultSpecs created by the procurement flow are processed by ISSM-MEC and realized through interaction with the external actuators as depicted in Figure 3-1.

To support these capabilities, Table 3-1 and Table 3-2 introduce more details about the necessary operations to cover the ISSM Workflow Manager service.

Table 3-1: Definition of ISSM-WFM Service (per-domain level)

Service name**: ISSM-WFM**		Type**: cross-*domain***
Capabilities	**Support (O|M)**	**Description**
Create Workflow	**M**	Allows a 5GZORRO developer who has developed a new business workflow to onboard it onto ISSM. In the Argo based K8s native implementation, creation of a workflow is implemented corresponds to creating a Custom Resource, an instance of Argo Custom Resource Definition (CRD) "Workflow" and storing it in a repository for future instantiation. In addition, an Event Mediator instance will be created for this flow with event source being a well-known topic of the Event Bus and the sink being this is this Workflow Argo controller. At run time events targeted to this workflow will be mediated using the instantiation ID obtained by this workflow instance upon instantiation.
Delete Workflow	**M**	Allows a 5GZORRO developer to remove a previously created workflow.
List Workflows	**M**	Allows a 5GZorro developer to explore existing flows.
Get Workflow	**M**	Allows a 5GZORRO developer to inspect a specific flow.
Instantiate Workflow	**M**	Instantiates a workflow. Implementation wise, instantiation of a workflow corresponds to applying the workflow definition against K8s API server.
List Flow Instance	**O**	Allows to inspect currently existing flow instances.
ReStart Flow Instance	**O**	Restarts previously paused workflow. Implementation wise this operation corresponds to Argo restart API call.
Pause Flow Instance	**O**	Pauses currently executing flow. Implementation wise this operation corresponds to Argo pause API call
Cancel Flow Instance	**M**	Cancels a flow in progress. This capability might have side effects, since business workflows are not atomic. The progress achieved by a cancelled workflow will be reflected in progressSpec.
Get Flow Instance Progress	**M**	Allows to inspect the current progress of an executing workflow
Update Workflow	**O**	This capability is included for convenience and completeness. It can be achieved via Delete/Create Workflow operations.
Notes:		
1. This is an internal service of the platform. It is not directly accessible to any persona described in Personas subsection, except 5GZorro Platform Developer, who can manage lifecycle of the ISSM workflows to support business operations of the platform (see Figure 3-1). 2. In case of Argo Workflows and Events backing ISSM-WFM, as shown in Figure 3-2, all capabilities are transparently translated to Kubernetes API requests (Argo CLI is just a shell on top of K8s CLI and all requests can also be issued directly against K8s API Server). 3. All other personas cannot manage lifecycle of the 5GZORRO workloads and only indirectly trigger Execute/Pause/Cancel/Progress capabilities above via secured Event Bus that might be additionally proxied by a POP gateway (not shown in Figure 3-1 and Figure 3-2 for simplicity).		

Table 3-2-: Definition of ISSM-WFM service interfaces

Operation name: **createFlow**

Description	Validates workflow and creates an entry in the ISSM workflow repository.	
Input Parameters	**Type**	**Description**
flowSpec	YAML	ISSM flow specification expressed in Argo YAML dialect.
Output Parameters	**Type**	**Description**
flowID	String	A unique ISSM assigned identifier or an empty string in case of a failure.
status	String	A code of operation completion and error information.
Notes		

Operation name: **deleteFlow**		
Description	Deletes a flow with a specified ID from ISSM.	
Input Parameters	**Type**	**Description**
flowID	String	Unique ID of the flow obtained as output of a previously called createFlow operation.
Output Parameters	**Type**	**Description**
status	String	A code of operation completion and error information.
Notes		

Operation name: **getFlows**		
Description	Lists all flows defined in ISSM-WFM.	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
workflowList	YAML	A list of registered (i.e., previously created workflows) and their metadata. The output can be used to extract fowIDs and use them as input to getFlow to obtain a full specification of an individual flow.
status		A code of operation completion and error information.
Notes		

Operation name: **getFlow**		
Description	Returns a full specification of a given flow.	
Input Parameters	**Type**	**Description.**
flowID	String	Unique ID of the flow obtained as output of a previously called createFlow operation.
Output Parameters	**Type**	**Description**
flowSpec	YAML	Full YAML specification of the flow.
status	String	A code of operation completion and error information.
Notes		

Operation name: **instantiateFlow**

Description	Starts execution of a flow with the specified ID.	
Input Parameters	**Type**	**Description**
flowID	String	Unique ID of the flow obtained as output of a previously called createFlow operation.
inputSpec	YAML	A YAML document specifying all key/value parameter pairs (dependent on a specific workflow).
Output Parameters	**Type**	**Description**
flowInstanceID	String	A unique ID of an instance of a previously created flow. A value of *flowInstanceID* is created by conventionally concatenating a flowID (a static ID of a flow obtained upon creation) with a dynamically generated unique ID. One should think about createFlow as "creating a class" and instantiateFlow as" instantiating an object" of that class.
status	String	A code of operation completion and error information.
Notes:		
Workflows execute asynchronously. Therefore, to find out the results of the workflow execution, one either has to call the getProgress operation or specify a push notification configuration in the inputSpec, so that notifications can be pushed either via Kafka or a Webhook, for example		

Operation name: **pauseFlowInstance**		
Description	Pauses a flow in progress.	
Input Parameters	**Type**	**Description**
flowInstanceID	String	A runtime ID of a flow that was previously instantiated via instantiateFlow.
Output Parameters	**Type**	**Description**
status	String	A code of operation completion and error information.
Notes		

Operation name: **restartFlowInstance**		
Description	Restarts a previously paused workflow.	
Input Parameters	**Type**	**Description**
flowInstanceID	String	A runtime ID of a flow that was previously instantiated via instantiateFlow.
Output Parameters	**Type**	**Description**
status	String	A code of operation completion and error information.
Notes		

Operation name: **cancelFlowInstance**		
Description	Cancels an executing flow.	
Input Parameters	**Type**	**Description**
flowInstanceID	String	A runtime ID of a flow that was previously instantiated via instantiateFlow.
Output Parameters	**Type**	**Description**
status	String	A code of operation completion and error information.
Notes		

Operation name: **getProgress**		
Description	This operation returns a point in time flow progress. It is idempotent and can also be invoked on the flows that already completed.	
Input Parameters	**Type**	**Description**
flowInstanceID	String	A runtime ID of a flow that was previously instantiated via instantiateFlow.
Output Parameters	**Type**	**Description**
progressSpec	YAML	A YAML document describing progress of the flow at the time of the operation invocation. If the flow has already been completed progressSpec will comprise the complete progress specification of the workflow.
status	String	A code of operation completion and error information.
Notes		

Operation name: **updateFlow**		
Description	This operation allows to update an existing flow definition in ISSM-WFM.	
Input Parameters	**Type**	**Description**
flowID	String	Unique ID of the flow obtained as output of a previously called createFlow operation.
flowSpec	YAML	A YAML document defining a new version of the workflow specification.
Output Parameters	**Type**	**Description**
status	String	A code of operation completion and error information.
Notes		

Operation name: **listFlowInstances**		
Description	This operation allows to update an existing flow definition in ISSM-WFM.	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
flowInstancesList	Array	An array containing instance IDs for all the flows visible to the caller.
Notes		

3.2 ISSM-O

ISSM Optimizer (ISSM-O) is a cross-domain component that optimizes cost-efficiency and cost-trustworthiness trade-offs of network services and slices required to be created in a context of a specific business transaction subject to constraints such as security and service area and continuously optimizes services and slices that have been already set up in previous transaction flow executions.

3.2.1 Design Updates

Figure 3-4 depicts ISSM-O design of Figure 3-1 in greater detail. ISSM-O performs two types of optimizations:

1. Continuous Slice and Service optimization: when it identifies an opportunity for reallocation of resources resulting in accrued benefit that is higher than projected management overhead of reallocation, it initiates an appropriate workflow on behalf of a persona that owns a service or a slice;
2. Initial Slice and Service optimization: this optimization is triggered by instantiation of some business flows, such as slice and service provisioning across domains. The goal of this optimization is to find initial allocation of resources to satisfy constraints (e.g., related to geographic localization, trust, performance, security and cost) fast.

In essence, these two optimizations are very similar. What differentiates between them is the time horizon (mid/long term vs short term), time budget an operator of the 5GZORRO platform is ready to invest in searching for an optimized solution, admitted optimality gap and the overhead cost budget that can be invested into searching for an optimized solution.

ISSM-O is an internal service of the 5GZORRO platform and, as such, it is not directly accessible by any persona except 5GZORRO component or by any platform component except ISSM-WFM. While ISSM-O can be used in multiple capacities (e.g., for collaborative optimization at the RAN, resource, Slice, SDN, MEC resources and service levels and we expect to involve ISSM-O in multiple scenarios, our design demands creation of a flow that would address ISSM-O for any reason).

An interaction with ISSM-O is declarative Kubernetes style. In this final design, ISSM-O is concerned with cost-efficiency optimization of slices and services at the marketplace resources selection. A requestor of slice or a service specifies an intent for the slice or service that comprises a YAML document that includes:

1. Cost: operational cost of resources as advertised on the marketplace resource offers;
2. Trust: reputational trust levels of resources as inferred from the historic data by the Trust Management module;
3. Security: security levels of resources as advertised in the marketplace resource offers;
4. Geographic location: geographic location of resources (should match the geographic feasibility constraints of a slice or a service intent);
5. Performance: performance of the resources as advertised in the marketplace resource offers.

The domain model to support this optimization uses the resources model defined in deliverable D3.1 and D3.3.

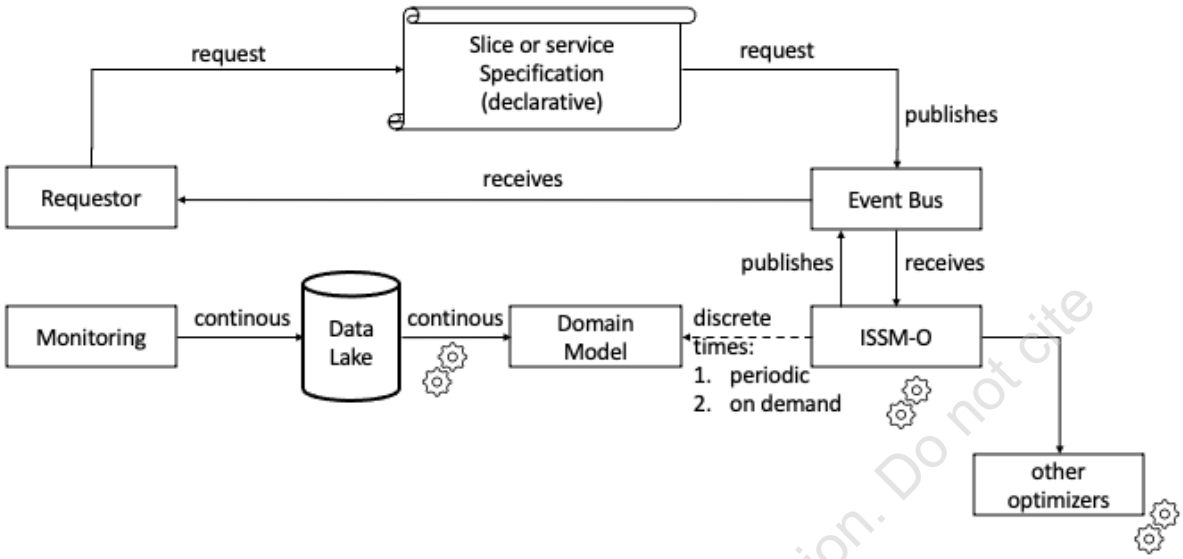

Figure 3-4: ISSM-O High Level Design

The main architectural challenge regarding ISSM-O is making it pluggable to ensure sustainability. Since optimization problems faced by ISSM-O are NP-hard, to address a specific problem a specialized heuristic or approximation usually is used to best exploit the problem properties. If a problem is sufficiently small and benign it can be solved *exactly* by commercial grade solvers, such as CPLEX [26] or Gurobi [22]. In other cases, methods like relaxation and rounding, column generation, etc. can be deployed.

As mentioned earlier, ISSM-O deals with NP-hard problem, which is to embed logical networks on top of shared substrate network across all the technological domains. In this regard, we have studied the problem of service embedding and resource allocation in the implement it in the ISSM-O component of 5GZORRO. The main issue with this joint problem of service embedding and resource allocation is that it becomes interactable when the problem size grows, meaning more nodes are added to the network or more services are issue to be deployed in the network. The state-of-the-art works in the domain mainly seek individual embedding of service instances (i.e., virtual networks) with every Virtual Network Function (VNF) instance and every individual forwarding path defined when a user session occurs. On the contrary our proposed model is Fluid. Our heuristic solution comprises two phases: (1) slice capacity planning using Linear Programming (LP) and (2) greedy Protocol Data Unit (PDU) session allocation when sessions are initiated by the slice/service consumer. Using this technique, we are able to scale to extremely big network with thousands of users making service requests.

Concerning the first phase of our proposed method, which is to plan the capacity in the network using LP techniques, we use mathematical solvers to reach an optimal solution to the problem. There are many solvers out there than can be employed for solving such problem, but each with their own characteristics and a set of limitations. After evaluating several possibilities for open-source tools, we have selected PuLP [27] as a tool to generate general Linear Program implementations that can then be given as input to different backend solvers. Our main goal in the ISSM-O design is to make it compatible with different optimization solvers, where based on the requests from the customer and the time required to solve a model, the backend solver can change.

The main appealing feature of PuLP is that it is essentially capable of generating Mathematical Programming System (MPS) or LP files and call GLPK [28], COIN-OR CLP/CBC [29], CPLEX [26], GUROBI [22], MOSEK [30],

XPRESS [31], CHOCO [32], MIPCL [33], SCIP [34] to solve linear. Our implementation of ISSM-O can be easily tuned to employ any of the mentioned solvers to be used for the optimization.

Finally, it should be noted that other optimization engines and mechanisms can be chained with ISSM-O for collaborative optimization (e.g., optimization at different levels of the stack). More details about the ISSM-O service capabilities at cross-domain level are outlined in Table 3-3 and Table 3-4, respectively.

Table 3-3: Definition of ISSM-O service (cross-domain level)

Service name**: ISSM-O**		Type**: *Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Allocate Slice or Service	**M**	This capability provides for cost-efficient allocation of resources present at the marketplace to implement the required slice or service.
Optimize	**M**	This capability reoptimizes resource selection for the resources allocated to a service or a slice to explore benefit from the new offerings.
Set Optimization Cron	**O**	This capability allows to set up optimization cron Jobs to perform optimization at desired intervals/dates
Configure Optimization	**M**	This capability allows to fine tune specific optimization heuristic.
Notes		

Table 3-4: Definition of ISSM-O service interfaces

Operation name: **optimizeShortTerm**		
Description	This method is called to obtain an optimized allocation of resources to a slice or a service to optimize cost-efficiency.	
Input Parameters	**Type**	**Description**
spec	YAML	This YAML document describes an intent of the requestor and will be translated into constraints that will be combined into a populated domain model used by the optimizer.
Output Parameters	**Type**	**Description**
solutionSpec	YAML	This YAML document contains a solution specification (selected resources from the marketplace to optimize cost-efficiency).
status	String	A code of operation completion and error information.
Notes		

Operation name: **optimizeLongTerm**		
Description	This method reoptimizes an entire portfolio of slices and services specified by a requestor. This is a long-term optimization. It cannot be achieved via individual greedy optimizations.	
Input Parameters	**Type**	**Description**
spec	YAML	This parameter is a YAML document that describes services and slices that should be re-optimized. A wildcard can be specified, which will cause a global re-optimization attempt. In addition, it contains a cron task specifications to cause periodic or scheduled optimizations. Besides the optimization scope and cron spec, this document might include optimization configuration key/value pairs.
Output Parameters	**Type**	**Description**

	solutionSpec	YAML	This YAML document contains a solution specification (selected resources from the marketplace to optimize cost-efficiency).
	status	String	A code of operation completion and error information.
Notes			

3.3 ISSM MEC Manager

ISSM-MEC Manager is a cross-domain component that realizes control plane of the Cloud-Native MEC Platform (CNMP) the latter is per-domain component. Architecturally, using the terminology of ETSI MEC, ISSM MEC Manager belongs to the *MEC System*, i.e., the control plane of MEC, and CNMP represents *MEC Hosts*.

The key idea behind ISSM MEC Manager is to use Kubernetes (k8s) as the orchestrator for CNFs that are hosted by MEC to extend the cloud native experience also to the control plane itself rather than treat only as NFVI that should be externally orchestrated.

3.3.1 Design Updates

ISSM MEC Manager (ISSM-MEC) module is translating the intent-based requests for instantiating/managing Services, Slices, and Edge applications into the interactions with the orchestration software. Being part of 5GZORRO ISSM, ISSM-MEC participate in fine-grained event-driven workflow-based design of and acts as an intelligent link to the Network Slice and Service Orchestrator-NSSO (Section 4.2). In order to receive an up-to-date list of edge resources and services, it periodically contacts the NSSO and synchronises its knowledge on edge resources, mapping them with intent-oriented models. Additionally, it receives the MEC application intents through the Event Bus of ISSM (Figure 3-2). The intents are linked to the available resource type at the edge level (compute, storage, and network) or to the location of the edge platforms or Points of Presence (PoP). In addition to NSSO, we implemented an experimental cloud-native MEC platform, which is treated as VIM-less K8s NFVI in which orchestration ant the CNF and Service level is being achieved using K8s native Operator Framework and Workflow mechanisms. ISSM-MEC Manager is fully aligned with the new distributed design of ISSM-WFM. The latter to dispatches declarative deployment requests to it as depicted in Figure 3-1 and Figure 3-2.

Figure 3-5 depicts the internal structure of ISSM-MEC which is very simple by design and acts as a universal translation layer between the rest of ISSM and the external environment specific actuators. Currently, we support two actuators: Network Slice and Service Orchestrator (NSSO) and the experimental cloud native MEC Platform (CNMP) described in the next sub-section. In the future, we envision extensibility to support additional actuators.

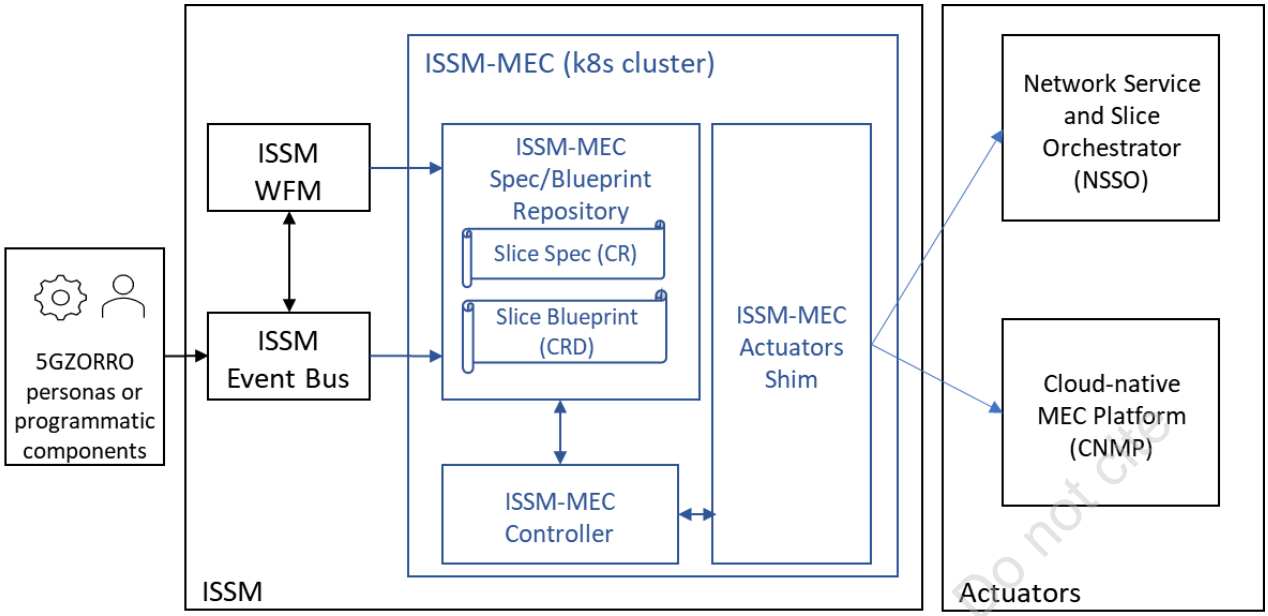

Figure 3-5: ISSM-MEC architecture and its interaction with the rest of ISSM and the external actuators

As a sub-module of ISSM, ISSM MEC Manager does not implement any individual service or has any associated APIs. It interacts with the rest of ISSM through the message bus and adheres to the information model for fetching/parsing messages posted there. On the 'southern' side, ISSM-MEC is a client for the relevant APIs exposed by the actuators, namely NSSO and CNMP.

3.4 Cloud-Native MEC Platform

Cloud Native MEC Platform (CNMP) is a per-domain component. External to ISSM, it was designed to represent and experiment with the emerging cloud native MEC environment, to be managed/federated under 5GZORRO. This new architectural component was conceived during the first year of the project, following the industry trend for cloud native transformation. In 5GZORRO its role is twofold: first, to demonstrate that 5GZORRO platform can integrate with cloud native k8s based MEC and, second, to extend 5GZORRO use cases validation to this emerging type of platform.

3.4.1 Design Updates

Figure 3-6 represents the Cloud Native MEC Platform (CNMP) and its interaction with the ISSM. Being external to ISSM and ISSM-MEC, CNMP is based on the same underlying technology as ISSM-MEC and thus can easily be integrated with it. We adopt multi-cluster view on the overall design. The main idea behind this design is to use the same K8s NFVI for both slices and application and services that use them thus creating a uniform cloud native environment. The control plane of CNMP comprises K8s Operators (i.e., CRDs and Controllers) and K8s Workflows (which are a special case of an operator). To deploy a slice or a service, its declarative specification (CRD) is delivered to the right K8s clusters using the same Multi-cluster Manager component (implemented using Open Cluster Management for K8s) and appropriate operators and flows are triggered in a distributed fashion to trigger local deployments, configuration, orchestration, and stitching across domains. For example, a UPF is locally deployed in one domain while 5GC is deployed in another domain using appropriate operators with UPF to 5GC stitching workflows running in the cross-domain ISSM-WFM component.

In our current design we use free5GC [35], which we "kubernetized" from its basic "dockerized" distribution, to provide each CNF as K8s service. Furthermore, we developed novel 5G Core and UPF Operators that act as VNFM for these CNFs with Argo Workflows acting as NFV Orchestrator.

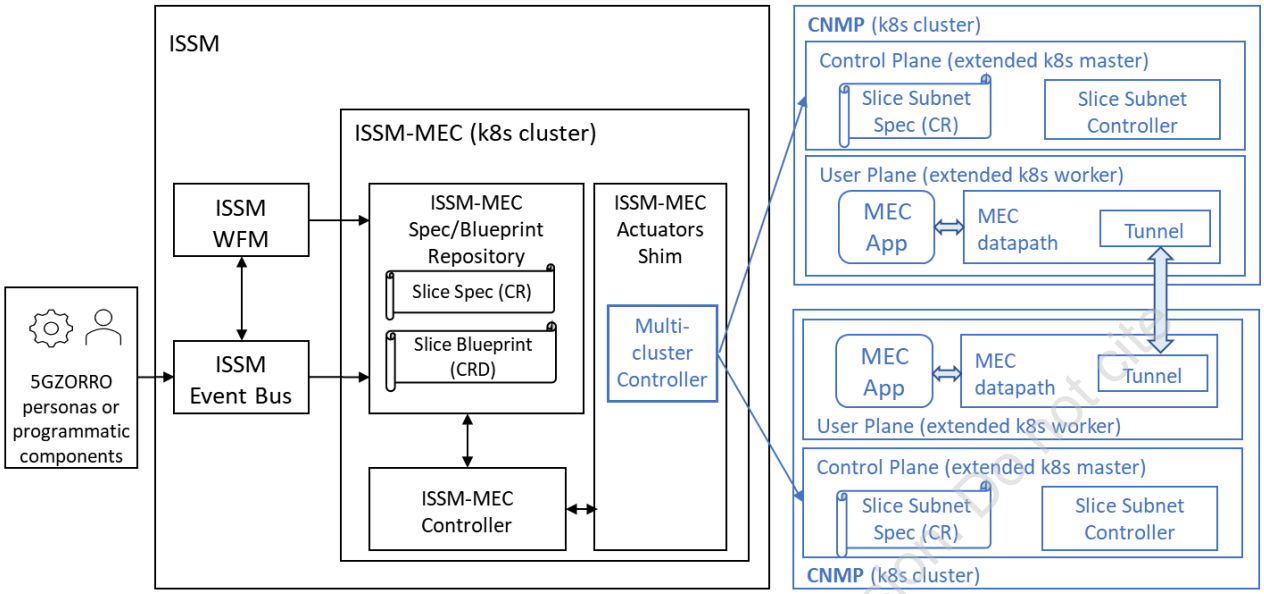

Figure 3-6: Cloud-native MEC platform

More details about the Cloud-native MEC Platform service capabilities at domain and cross-domain levels are outlined in Table 3-5 and Table 3-6, respectively.

Table 3-5: Definition of CNMP service (per-domain/cross-domain level)

Service name: **CNMP**		Type: ***Per-domain / Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Edge Slice Lifecycle Management	**M**	This capability allows deploying network slices with edge-derived properties such as in which locations the slice is required and what is the slice's resource budget limits.
Edge Slice Subnet Lifecycle Management	**M**	This capability allows deploying slice subnets in the existing remote k8s clusters and managing lifecycle changes of these objects. Beyond CRUD, we plan to support attaching/detaching slice subnets to infrastructure slices, e.g., slices created by the infrastructure provider and/or by ISSM through infrastructure-capable actuators such as NSSO.
Edge Application Lifecycle Management	**O**	This capability allows deploying edge applications across multiple managed k8s clusters and managing these applications according to their SLAs. Beyond CRUD, we plan to support moving existing application instances to a different cluster, extending the application spread to additional clusters, and removing the application from some of the clusters according to changes in the demand and in the cluster capacity.
Notes		

Table 3-6: Definition of CNMP service interfaces

Operation name: **createEdgeSlice**		
Description	This method, upon receiving the intent document, creates a blueprint for the required slice and dispatches it to the managed clusters on a need-to-know basis.	
Input Parameters	**Type**	**Description**
edgeSliceManifest	YAML	Yaml file containing the specification for the slice to be created.
Output Parameters	**Type**	**Description**
edgeSliceID	String	Unique identifier of the deployed slice instance.
Notes		

Operation name: **destroyEdgeSlice**		
Description	Delete the slice instance.	
Input Parameters	**Type**	**Description**
edgeSliceID	String	The identifier for the slice to be deleted.
Output Parameters	**Type**	**Description**
status	String	Return code describing operation result.
Notes		

Operation name: **updateEdgeSlice**		
Description	Update the slice instance; possible updates can be changes in SLA or decisions to consume different resources	
Input Parameters	**Type**	**Description**
edgeSliceID	String	The identifier for the slice to be modified
edgeSliceManifest	YAML	Yaml file containing the specification for the slice after modification
Output Parameters	**Type**	**Description**
status	String	Return code describing operation result
Notes		

Operation name: **getEdgeSlice**		
Description	Allows to retrieve the desired and the actual states of the slice instance.	
Input Parameters	**Type**	**Description**
edgeSliceID	String	The identifier for the slice to be retrieved.
edgeSliceManifest	YAML	Yaml file containing the specification for the slice after modification.
Output Parameters	**Type**	**Description**
edgeSliceManifest	YAML	Yaml file containing the specification for the slice desired state.
edgeSliceStatus	YAML	Yaml file containing the description for the slice actual state.
status	String	Return code describing operation result.
Notes		

Operation name: **createEdgeSliceSubnet**

Description	This method, upon receiving the intent document, creates a blueprint for the required slice subnet and dispatches it to the managed clusters on a need-to-know basis.	
Input Parameters	**Type**	**Description**
edgeSliceSubnetManifest	YAML	Yaml file containing the specification for the slice subnet to be created.
Output Parameters	**Type**	**Description**
edgeSliceSubnetID	String	Unique identifier of the deployed slice subnet instance.
Notes		

Operation name: **destroyEdgeSliceSubnet**		
Description	Delete the slice subnet instance.	
Input Parameters	**Type**	**Description**
edgeSliceSubnetID	String	The identifier for the slice subnet to be deleted.
Output Parameters	**Type**	**Description**
status	String	Return code describing operation result.
Notes		

Operation name: **updateEdgeSliceSubnet**		
Description	Update the slice subnet instance; possible updates can be changes in SLA or decisions to consume different resources.	
Input Parameters	**Type**	**Description**
edgeSliceSubnetID	String	The identifier for the slice subnet to be modified.
edgeSliceSubnetManifest	YAML	Yaml file containing the specification for the slice subnet after modification.
Output Parameters	**Type**	**Description**
status	String	Return code describing operation result.
Notes		

Operation name: **getEdgeSliceSubnet**		
Description	Allows to retrieve the desired and the actual states of the slice subnet instance.	
Input Parameters	**Type**	**Description**
edgeSliceSubnetID	String	The identifier for the slice subnet to be retrieved.
Output Parameters	**Type**	**Description**
edgeSliceSubnetManifest	YAML	Yaml file containing the specification for the slice subnet desired state.
edgeSliceSubnetStatus	YAML	Yaml file containing the description for the slice subnet actual state.
status	String	Return code describing operation result.
Notes		

4 MANO and Slicing Enhancements

In this section is described a set of modules designed to complement the MANO and the Network Slicing management mechanism. Such modules aim to enhance the aforementioned mechanisms by encompassing several functionalities that facilitate Zero-touch management of virtual network elements, such as Slices, Network Functions and Services. The main entity belonging to such a set of modules in the Network Slice and Service Orchestrator (NSSO), in charge of managing the lifecycle of Vertical/Network Service and Slice in the intra-domain context. The NSSO interacts with the eLicensing Manager, implementing the required mechanism to validate and continuously check the right to use a given network resources by a certain consumer. The NSMM provides functionalities to manage networks at level of the VIM, required to automatically establish secure cross-domain connectivity in collaboration with the VPNaaS module. Finally, the Any Resource Manager (xRM) provides an abstraction of 5G Virtualized infrastructure exposed towards the 5GZORRO Platform. In Figure 4-1 is depicted the set of modules and its positioning with respect to the 5GZORRO reference architecture.

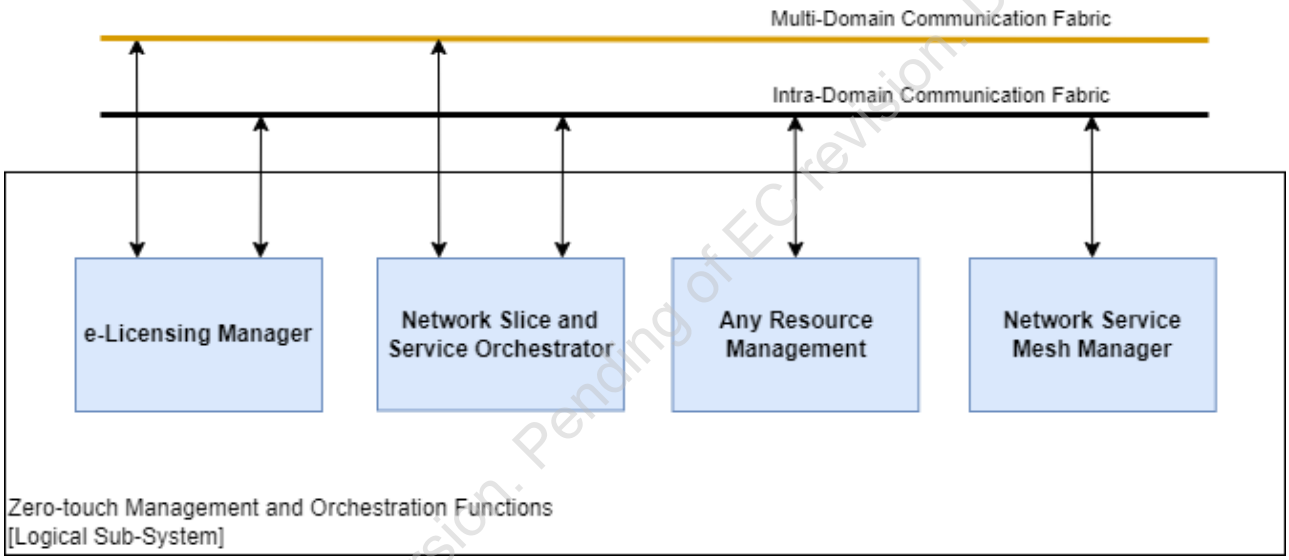

Figure 4-1 – MANO and Slicing Enhancement Modules positioned in the 5GZORRO reference architecture

4.1 Any Resource Manager

The Any Resource Manager (xRM) is a module in the 5GZORRO platform that directly interacts with the underlying 5G Virtualized platform offering, towards the upper layer applications, a set of services related to the resources monitoring and management, including a direct support to the 5GZORRO Offering Catalogue. In previous WP4 deliverable, specifically D4.1 [49] and D4.2 [50], the xRM was initially referred as Virtual Resource Manager (VRM) nevertheless, the module interacts with different set of resources so, in order to avoid misunderstandings, the name has been changed.

4.1.1 Design Updates

The xRM is designed to be a multi-container application, with the aim to provide the high level of flexibility required to immediately tackle any changes/updates in both resource composition and in the underlying 5G

virtualised infrastructure. In this section, the final design of xRM is reported which evolves what already reported in both D4.1 and D4.2

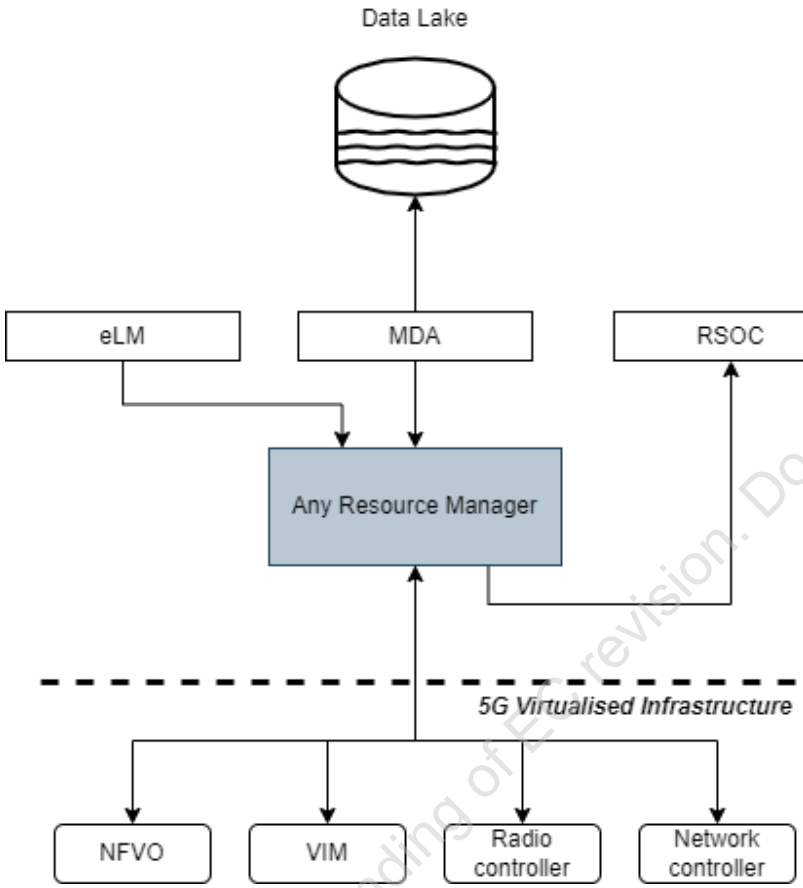

Figure 4-2: xRM interaction with other components of the 5GZORRO platform

The xRM contains the logic to manage the resource DB and interact with the Resource and Service Offer Catalogue (RSOC), which is the target of the resource definition translation process. The Resource Manager also offers an interface that Monitoring Data Aggregator (MDA) exploits for retrieving information used to reach the different components of the 5G Virtualisation platform, namely NFVO, VIM, Radio and Network Controller. Same interface is consumed by the e-Licensing modules to retrieve the coordinates to access the NFVO. Figure 4- shows the xRM relationship with the other components of the 5GZORRO platform.

As depicted in Figure 4-, xRM consists of several modules in charge to implement the functionality of resource storage (descriptors and, where required, packages), model translation, and 5G virtualisation infrastructure DB.

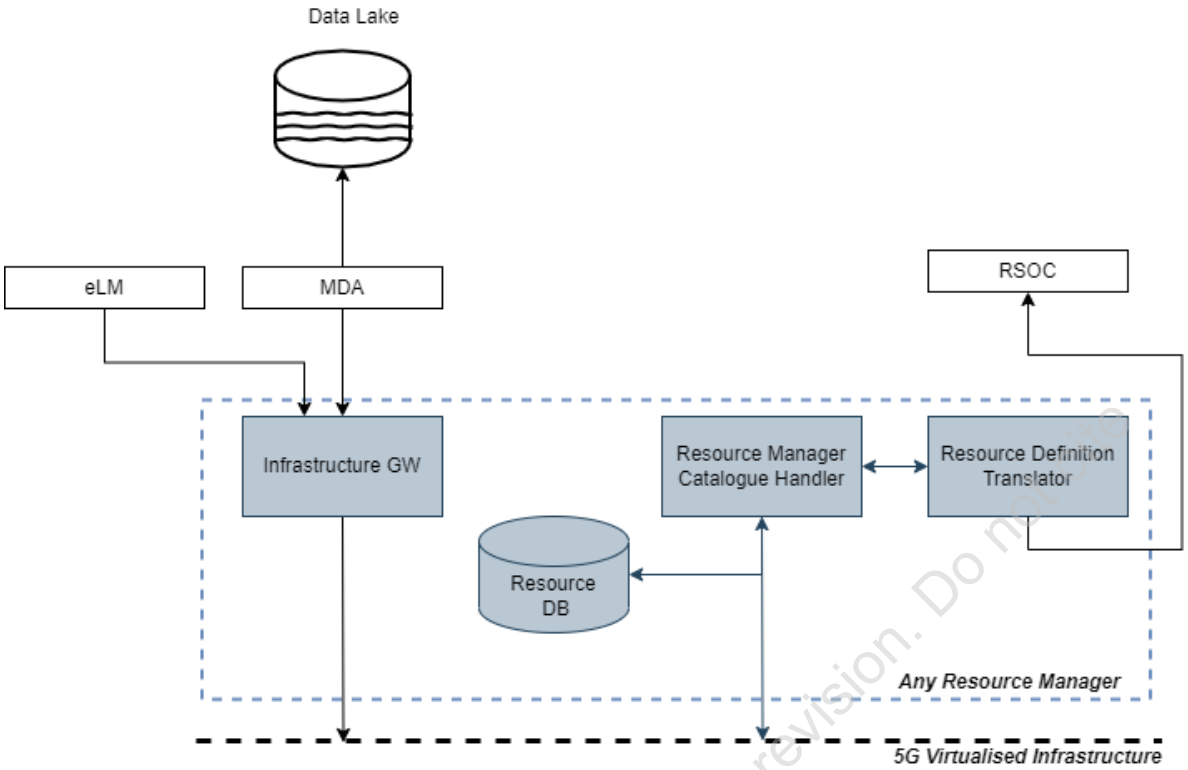

Figure 4-3: Detailed view of the modules composing the any Resource Manager

The *Infrastructure GW* provides the interface and the logic to access the underlying 5G Virtualised infrastructure, avoiding the need for components like MDA and e-Licensing modules to directly access the provider internal infrastructure. The *Resource DB* contains the resource definitions expressed in terms of descriptors and, when required, in terms of references to certain resource packages (e.g., the DB contains a VNF descriptor and a reference to the corresponding VNF package). The resource DB is managed by the *Resource Manager Catalogue Handler (RMCH)* that offers the interface and the logic to access to the resource information. The connection between the RM Catalogue Handler and the 5G Virtualised Infrastructure represents the capability of the xRM to access the internal catalogues of the different elements to add (onboard), update, and remove resources. Finally, the *Resource Definition Translator* provides the logic to translate the resources, as defined in the Resource DB, to the proper models defined by the TM Forum as well as the required interface to store such translations to the RSOC.

Resource Management Service

The resource to be orchestrated by 5GZORRO platform must be already present in the 5G virtualised infrastructure in order to be manipulated by the orchestration stack. With respect to D4.1 [49] and D4.2 [50], the Spectrum has been included among the set of resource managed.

Table 4-1: Definition of Resource Management service (per-domain)

Service name**: Resource Lifecycle Management**		Type**: *Per-domain***
Capabilities	**Support (O|M)**	**Description**
Manage NFVI resources and VNFs/CNFs	**M**	*Manage the edge/cloud NFVI resources and VNFs can be deployed within the domain. NFVI also include container-based virtualization platform and related Network Functions (e.g., Kubernetes and CNFs).*
Manage Radio resources	**M**	*Configure the radio platform within the domain in order to properly create slices at RAN layer.*
Manager Spectrum resources	**M**	*Manage spectrum resources can be configured at radio layer in the target domain*

Manage Network resources	**O**	*Manage the configuration of the network controllers within the domain.*
Notes		

Table 4-2: Definition of Resource Management service interfaces

Operation name: **addVirtualResource**		
Description	Add a virtual resource to the xRM that will also take care to Onboard it into the proper Virtualization Platform.	
Input Parameters	**Type**	**Description**
VirtualResourceType	ENUM	Type of virtual resource to be added in the internal VRM catalogue and onboarded to the proper virtualization platform • VNF • CNF • PNF • EDGE/MEC_RESOURCE • NETWORK_SLICE
description	Object	An object that properly describes the Resource type (e.g., VNFD, PNFD, etc).
Output Parameters	**Type**	**Description**
result	String/UUID	Unique Resource id in the VRM internal store.
Notes		

Operation name: **addRadioResources**		
Description	Adds the RAN resource under the control of the 5GZORRO Radio Resource Management and Control entity (e.g., Radio Controller).	
Input Parameters	**Type**	**Description**
RanResource	List	A list of technological description of each of the RAN element to manage.
Output Parameters	**Type**	**Description**
result	String/UUID	Determines whether the registration of all the RAN elements in the controller was successful or not. In the former case, a resource id is returned, an Error in the latter.
Notes		

Operation name: **addSpectrumResources**		
Description	Adds the Spectrum resource (e.g., Spectoken) resource under the control of the 5GZORRO Spectrum Management entity (e.g., Spectrum Manager).	
Input Parameters	**Type**	**Description**
SpectrumResource	Object	Formal description of a spectrum resources
Output Parameters	**Type**	**Description**
result	String/UUID	Determines whether the registration of all the Spectrum elements in the spectrum manager was successful or not. In the former case, a resource id is returned, an Error in the latter.
Notes		

Operation name: **addNetworkResource**		
Description	Adds the Network resource under the control of the 5GZORRO Network control entity (e.g., SDN Controller).	
Input Parameters	**Type**	**Description**
NetworkResource	List	A list of technological description of each of the Network element to manage (e.g., Virtual or physical devices: Switches, routers, etc).
Output Parameters	**Type**	**Description**
result	String/UUID	Determines whether the registration of all the Network elements in the controller was successful or not. In the former case, a resource id is returned, an Error in the latter.
Notes		

Operation name: **removeResources**		
Description	Adds the Network resource under the control of the 5GZORRO Network control entity (e.g., SDN Controller).	
Input Parameters	**Type**	**Description**
resourceIds	List (String/UUID)	List of unique identifiers of the resources to be removed.
Output Parameters	**Type**	**Description**
result	HTTP Response	A success code is returned. In case of error, a proper error code with a description as payload will be provided.
Notes		

Resource Monitoring Service

xRM offers an interface that allows the monitoring of relevant parameters of the 5G Virtualized infrastructure. In particular, the xRM enables monitoring at Cloud (NFV and Kubernetes), Radio, Spectrum and, optionally, Network (SDN) layer. Such an interface is consumed by the MDA to collect metrics form different targets on demand, aggregate and inject them into the 5GZORRO Platform Data Lake, e.g., in form of data streams.

Table 4-3 Definition of Resource Monitoring service (per-domain)

Service name**: Resource Monitoring Service**		Type**: *Per-domain***
Capabilities	**Support (O|M)**	**Description**
Provide NFVI resource and VNF/CNF performance statistics	**M**	Provide access to monitoring data about the NFVI resource usage and VNF performance/fault statistics 5GZORRO Data Lake.
Provide RAN slice sub-net statistics	**M**	Provide access to information regarding the status of a RAN slice sub-net in 5G ZORRO Data Lake
Provide Spectrum usage statistics	**M**	Provide access to monitoring data concerning Spectrum usage in the 5GZORRO Data Lake.
Provide Network statistics	**O**	Provide access to monitoring data concerning Virtual Network infrastructure in the 5GZORRO Data Lake.
Notes		

Table 4-4 Definition of Resource Monitoring service interfaces

Operation name: **getMetrics**		
Description	Allow the collection of metrics of different types from different resource infrastructures	
Input Parameters	**Type**	**Description**
infraType	ENUM	Type of infrastructure e.g., VIRTUAL_INFRA, RADIO, NETWORK, etc
targetResource	String/UUID	Identifier of the target resource producing the metrics (e.g., VM Identifier)
metricType	ENUM	Type of Metric e.g., CPU, MEMORY, USAGE_TIME, etc.
metricName	String	Name of the metric to be collected as defined into the target infrastructure
Output Parameters	**Type**	**Description**
metricValue	Object	An object containing the metric value along with other elements that depend on infrastructure specifications.
Notes		

Support to Offer Catalogue

The Any Resource Manager also supports the 5GZORRO RSOC by adding resources that are to be shared or traded later in the platform. In this regard, the Any resource manager provides a specific service that allow the translation of a given resource, selected to be traded, from its technical specification to the TM Forum models supported by the 5GZORRO RSOC.

Table 4-5 Definition of Resource exposing service (per-domain)

Service name**: Resource exposing**		Type**: *Per-domain***
Capabilities	**Support (O|M)**	**Description**
Explore available resources	**M**	Provide a list with description of available resources which can be filtered per resource type.
Resource exposing to 3rd parties	**M**	The entity that uses this service can select a resource and mark it a "sealable". This action triggers inside the xRM logic the translation of the resource current information model to the one provided by the TM forum. The resulting information model will be sent to the Offering Catalogue for the further operations required in order to create a new offer. A resource can be removed from the set of sealable ones: the xRM will invoke the removal of the correspondent offers to the Offering Catalogue interface.
Notes		

Table 4-6 Definition of Resource exposing service interfaces

Operation name: **getResources**		
Description	Return a list of available resources.	
Input Parameters	**Type**	**Description**
Id	String/UUID	ID of the resource. It is NOT MANDATORY. If specified, the resource_list will contain 1 (the resource with the given id) or 0 resource (if ID is not present).

	type	ENUM	Resource type. Example: • VNF • CNF • PNF • EDGE/MEC_RESOURCES • NETWORK_SLICE • SPECTRUM • RAN
	status	ENUM	A tuple with a couple of values: (exposed, deployed): • EXPOSED/NOT_EXPOSED the resource is sealable or not
Output Parameters		**Type**	**Description**
	resourceList	List	List of resources found. The list varies based on the values of the input parameters. In the case of no parameters provided, the returned list will contain the complete list of resources available.
Notes			

Operation name: **exposeResources**		
Description	Marks resources (one or more) as sealable (3rd parties exposed through the Offering Catalogue) and triggers the correspondent building of resource descriptors aligned with TM Forum models.	
Input Parameters	**Type**	**Description**
Id	list	List of IDs of the resources to be exposed in the 5GZORRO Offering Catalogue.
Output Parameters	**Type**	**Description**
resourceList	List	List of IDs of the resources exposed.
Notes		

Operation name: **unexposeResources**		
Description	Marks resources (one or more) as not sealable (3rd parties exposed through the Offering Catalogue) and triggers the removal operation from the Offering Catalogue.	
Input Parameters	**Type**	**Description**
Id	list	List of IDs of the resources to be unexposed and removed from the 5GZORRO Offering Catalogue.
Output Parameters	**Type**	**Description**
resourceList	List	List of IDs of the resources unexposed.
Notes		

4.2 Network Slice and Service Orchestrator

The Network Slice and Service Orchestration (NSSO) implements the logic to manage the lifecycle of vertical services and the underlying network slices supporting these services. In the final release, this module acts at the intra-domain layer of the 5GZORRO architecture, relying on the upper layers of the architecture to

provide the multi-domain capabilities. In this sense, the NSSO processes the vertical service and network slice instantiation requests, interacting with different modules to: provision the resources required by the service, configure the monitoring metrics, instantiate the required network services verifying the available licenses.

4.2.1 Design Updates

This module is based on the Vertical Slicer designed in 5G-TRANSFORMER [53], and further developed in 5Growth and other EU funded projects. The main objective is to enable the service design, customization and lifecycle management using high-level business-oriented parameters. In this sense, the NSSO translates the business level requirements into Network Slice (NS) and network service specific deployments. For this, this module contains vertical service blueprint (VSB) and Network Slice Template catalogues and implements the logic which maps vertical service instances to network slices and network services based on the instance specific constraints.

In 5GZORRO this component was enhanced and developed to support the following functionality

1. Support of automatic network connectivity provisioning across domains: the 5GZORRO platform introduces the Network Service Mesh Manager (NSMM), to handle the provisioning of the required connectivity across domains and virtualisation platforms. The Vertical Slicer will be extended to leverage the NSMM in its internal workflows, i.e., the automated translation of vertical service lifecycle management actions into specific requests towards the NSMM.
2. Extension of the supported Vertical Service definition and NS models: The VS definition and NS models will be improved with the introduction of a catalogue to support GSM Generic Slice Templates (GST), and the NS provisioning by means of NEtwork Slice Type (NEST) containing the specific values [56] . This new catalogue will allow a standard way of defining and requesting vertical services and network slices.
3. Automated configuration of the monitoring metrics using the metrics specified in the orchestration related descriptors (i.e., NSDs, VNFDs), with automated annotation of the service product, and transaction metadata through the Monitoring Data Aggregator (MDA) module.
4. Network service lifecycle management with automated e-licensing verification mechanisms.

In Figure 4-4, we illustrate the positioning of this module in the 5GZORRO architecture and the interfaces with the rest of the 5GZORRO platform modules.

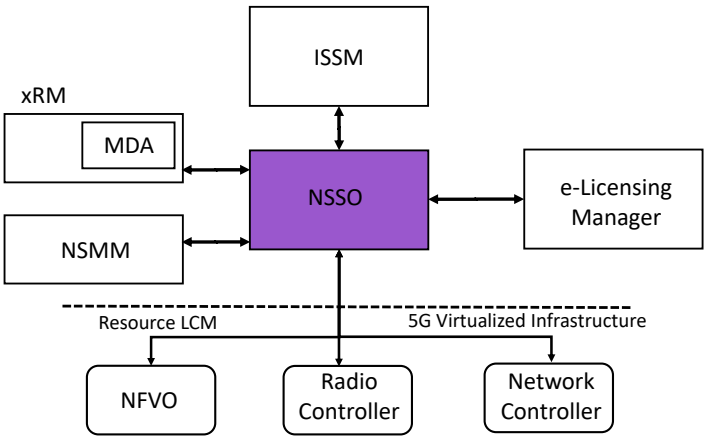

Figure 4-4: Network Slice and Service Orchestrator interfaces

Figure 4- illustrates the internal instantiation workflow of the NSSO, and the interactions with the other 5GZORRO Platform modules.

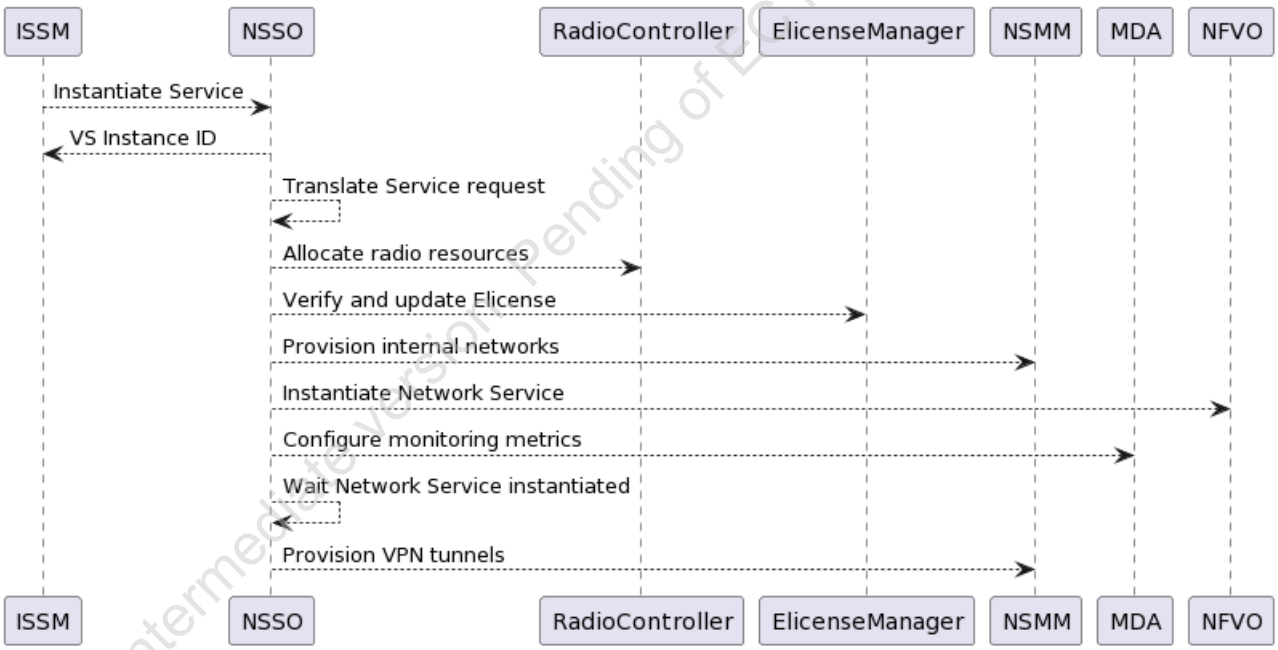

Figure 4-5: NSSO internal instantiation workflow

The following tables describe the operations to be supported by the 5GZORRO Network Slice and Service Orchestration module regarding VSB/VSD onboarding and vertical service lifecycle management, as established in D4.1, and highlighting the changes introduced in the final prototype. The OpenAPI specification is available in [55].

Table 4-7: Definition of VS catalogue management service (cross-domain level)

Service name: **VS Catalogue Management**		Type: ***Per-domain/ Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Onboard VSB	**M**	Create a new Vertical Service Blueprint in the platform. The final prototype allows to specify the specific templates to be used for the network slices in order to support the allocation of resources without a network service.
Onboard VSD	**M**	Create a new Vertical Service Descriptor.
Delete VSB	**M**	Delete a previously onboarded VSB.
Delete VSD	**M**	Delete a previously onboarded VSD.
Query VSB	**M**	Query the available VSBs.
Query VSD	**M**	Query the available VSDs.
Notes		

Table 4-8 Definition of VS LCM service interfaces (cross-domain level)

Service name: **VS Lifecycle Management**		Type: ***Per-domain/ Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Instantiate VS	**M**	Instantiate a vertical service based on a VSD.
Query VS	**M**	Query a specific vertical service instance. This operation was extended to enable retrieving the information regarding the VPN gateway information in order to create VPN connections across services running on different domains.
Terminate VS	**M**	Terminate a vertical service.
Modify VS	**M**	Modify a vertical service, to move from one VSD to another.
Notes		

Table 4-9: Definition of VS catalogue management service interfaces

Operation name: **onboardVSB**		
Description	Onboard a VSB.	
Input Parameters	**Type**	**Description**
VSB	String	The VSB to be onboarded. Extended with the Network Slice Template specific reference.
translationRules	List of rules	The policies to translate the VSB parameters into Network Slices and NFV-NS.
NST/NSDs	NST/NSDs	
Output Parameters	**Type**	**Description**
vsBlueprintId	String	The id assigned to the VSB on the catalogue.
Notes		

Operation name: **queryVSB**	
Description	Retrieve a VSB.

Input Parameters	Type	Description
vsbId	String	The id of the VSB to be deleted.
Output Parameters	**Type**	**Description**
VSB	VSB	The VSB retrieved.
Notes		

Operation name: **deleteVSB**		
Description	Remove a VSB.	
Input Parameters	**Type**	**Description**
vsbId	String	The id of the VSB to be retrieved.
Output Parameters	**Type**	**Description**
none	N/A	N/A
Notes		

Operation name: **onboardVSD**		
Description	Onboard a VSD.	
Input Parameters	**Type**	**Description**
VSD	String	The VSD to be onboarded.
Output Parameters	**Type**	**Description**
vsDescriptorId	String	The id assigned to the VSD on the catalogue.
Notes		

Operation name: **queryVSD**		
Description	Retrieve a VSD.	
Input Parameters	**Type**	**Description**
vsdId	String	The id of the VSD to be deleted.
Output Parameters	**Type**	**Description**
VSD	VSD	The VSD retrieved.
Notes		

Operation name: **instantiateVS**		
Description	Instantiate a VS.	
Input Parameters	**Type**	**Description**
VSD id	String	The VSD id.
VS instance specific parameters	List of parameter elements	Values for the VS parameters which are specific for the instance.
Output Parameters	**Type**	**Description**
Vertical service instance id	String	The id assigned to the vertical service instance.

Notes

Operation name: **terminateVS**		
Description	Finish a VS.	
Input Parameters	**Type**	**Description**
VS instance id	String	The vertical service instance id to be terminated.
Output Parameters	**Type**	**Description**
none	N/A	N/A
Notes		

Operation name: **modifyVS**		
Description	Adjust a VS.	
Input Parameters	**Type**	**Description**
VS instance id	String	The vertical service instance id to be modified.
VSD Id	String	The new VSD id to be used.
Output Parameters	**Type**	**Description**
none	N/A	N/A
Notes		

4.3 Network Service Mesh Manager

The aim of the Network Service Mesh Manager (NSMM) is to provide the functionalities for *enabling* and *securing* the stitching of slices and services across multiple domains. Thus, in order to do so, each stakeholder of the 5GZORRO Platform should deploy an NSMM in its domain.

As already mentioned in the deliverable D4.1 [49], the main consumer of the services exposed by the NSMM on the NBI is the NSSO (Network Slice and Service Orchestrator, see Section 4.2) which has a complete view of the end-to-end slices and services, thus the NSSO knows which services of different domains need to interact to each other and it requests to the NSMM of the involved domains to establish a secure connection between the cross-domain services.

4.3.1 Design Updates

The first functionality offered by the NSMM is to enable the stitching of slices and services across multiple domains and it consists in the creation of the necessary resources on the underlying VIM, which can be OpenStack or Kubernetes, these resources could be just external endpoints such a floating IP address in OpenStack or Ingress resources in the Kubernetes domain. The other functionality, which is to secure communication between cross-domain slices and services, is performed by the NSMM interacting with the ID&P, and the VPNaaS (or Inter-domain Security Service, described in Section 2.4). The ID&P module is used to retrieve, for each endpoint of the communication to be secured, a unique identity (DID) associated with a private/public key pair. Then, the DID and the exposed endpoint are sent to the peer in the other domain, in order to establish the connection. In this way, the other peer of the secure connection can retrieve/verify the public key using the received DID from the ID&P of its domain, before actually connecting to the endpoint,

to be sure to interact with the desired service. The VPNaaS is the module used to perform the actual secure connection between exposed services belonging to different domains.

The high-level architecture of the NSMM with the interactions between different modules is depicted in Figure 4-. The connection database is a dedicated database to store the status of all the secure connections and the related information. The NSMM Logic is in charge of creating and defining all the necessary network resources on the related VIM to allow the instantiation of the VPNaaS and the creation of a secure connection. The Secure Connection Component is in charge of interacting with the ID&P to retrieve keys and DID and to configure and create the actual VPN connection interacting with the VPNaaS modules in the network services.

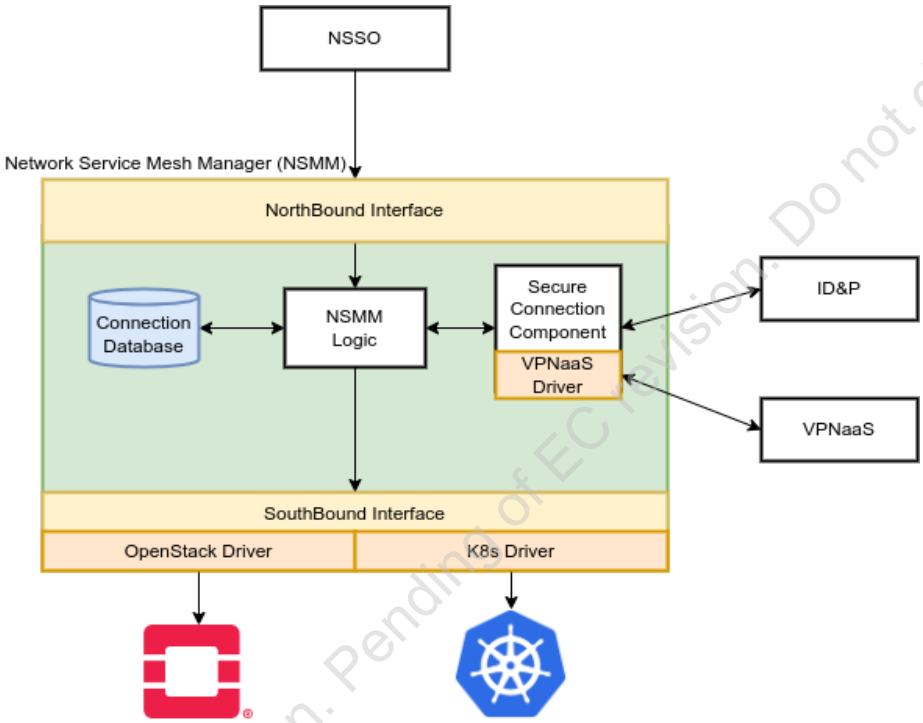

Figure 4-6 High-level NSMM architecture

The NBI of the NSMM exposes a set of REST APIs, described in the tables below, towards the orchestration stack.

Table 4-10: Definition Cross-domain slice stitching service (cross-domain level)

Service name**: Cross-domain Slice Stitching**		Type**: *Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Create slice connectivity resources	**M**	This capability allows creating the resources needed for enabling the secure connection between two different domains. These resources depend on the VIM used to deploy the related services
Create slice connectivity	**M**	This capability allows the establishment of the connectivity between slices.
Destroy slice connectivity	**M**	This capability allows the termination of the connectivity between slices.
Get connectivity status	**M**	This capability allows to retrieve information concerning the current slice connectivity statuses internally maintained by the NSMM.
Notes		

Table 4-11: Definition of Cross-domain slice stitching service interfaces

Operation name: **createSliceConnectivityResources**		
Description	This API performs the creation of local resources to enable the cross-domain end-to-end slice connectivity. It requires the specification of the VIM used for the related services.	
Input Parameters	**Type**	**Description**
slice_id	UUID	Slice instance id local to the domain.
Vim_type	Enum	The VIM type used, OpenStack or Kubernetes.
did	String	Distribute identifier used for retrieving public key of the remote VPN endpoint (client or server).
Output Parameters	**Type**	**Description**
response	Integer	HTTP response code.
Notes		

Operation name: **createSliceConnectivity**		
Description	This API performs the local configuration to enable the cross-domain end-to-end slice connectivity. It requires also the specification of the parameters for the creation of the secure connection, which are used for configuring and creating the VPN connection through the VPNaaS.	
Input Parameters	**Type**	**Description**
slice_id	UUID	Slice instance id local to the domain.
Remote_endpoint	JSON	This parameter specifies the remote peer endpoint of the slice in the other domain.
did	String	Distribute identifier used for retrieving public key of the remote VPN endpoint (client or server).
Output Parameters	**Type**	**Description**
response	Integer	HTTP response code.
Notes		

Operation name: **destroySliceConnectivity**		
Description	This API destroys the local configuration previously set for the stitching of a cross-domain end-to-end slice.	
Input Parameters	**Type**	**Description**
slice_id	UUID	Slice instance ID local to the domain. Since the NSMM keeps internally all the information associated to the slice_id, this parameter is the only required to delete the referenced configuration.
response	Integer	HTTP response code.
Notes		

Operation name: **getE2eSliceConnectivity**

Description		Retrieves all the local information concerning the connectivity of an end-to-end slice.
Input Parameters	**Type**	**Description**
slice_id	UUID	Slice instance ID local to the domain. Since the NSMM keeps internally all the information associated to the slice_id, this parameter is the only required.
Output Parameters	**Type**	**Description**
response	JSON	Dictionary in JSON that contains the full set of local information concerning the connectivity of the end-to-end slice, internally stored by the NSMM.
Notes		

4.4 E-Licensing Manager

5G use cases build on top of disruptive paradigms such as virtualization, softwarization or Anything as a Service (XaaS) as well as of novel technologies which continue to prove capable of bringing a whole new set of business opportunities. For software vendors to be able to adapt to this thriving ecosystem, innovative monitoring approaches (based on the interaction between new players in a 5G ecosystem) are required to support the use of licenses in their products. Thus, enabling a way to materialise the revenues on their development investments, intellectual property rights associated with them, and any other business plan related to the product. In particular, software vendors are those that provide licensed Network Functions (xNFs, that encompasses VNFs, CNFs or network functions composed by several VNFs/CNFs) which are network software functions that can be instantiated and replicated very quickly, thanks to the NFV technology in a multi-domain ecosystem. However, this agility comes along with an increased challenge regarding the license control and management which is only growing in complexity as new players enter the 5G ecosystem.

Vendors enrolled in the 5GZORRO ecosystem can onboard their software resources, exposing capabilities, licensing constraints, duration of xNF use and the business agreements associated through the 5GZORRO Marketplace Portal [42]. The xNF consumer must formally agree in order to use the resource by signing a smart contract that facilitates, verifies and enforces the negotiation of the agreements. Once the sign between the parties is effective, the xNF consumer is in readiness to use the resource in their own domain or in a third-party/external domain. The e-Licensing Manager (eLM) is deployed in the form of an Agent (eLMA) instance on every administrative domain that participates in 5GZORRO with the objective of tracking the usage of the purchased resource in real-time, verifying the compliance of the smart contract and providing prove of usage related to the licensing costs.

4.4.1 Design Updates

Figure 4-2 depicts the design of the monitoring processes (watchers) using existing standardization references such as [47]. To the eLM, the 5GZORRO Marketplace provides the information required to identify the resource/service to be monitored (VNF) along with the license policies (VNF-lp) in the form of a smart contract, *productOffering* and *productOfferingPrices* marketplace models (refer to D3.3) which are stored internally as *eLM Descriptor* and *License* models (Section 5.8). Each watcher in an eLMA instance (managed by the Watcher Manager, which is shown in Figure 4-3) is defined by a VNF and a VNF-lp which both determine the monitoring metric, restrictions and how to link a live instance of a VNF (VNF-i) to a particular watcher. As previously presented in D2.4, watchers provide validation in two stages:

- At instantiation time they check that the attached offering exists in the marketplace of the administrative domain where the xNF is going to be deployed. Every involved *Resource* and *Service Specifications* as well as associated *productOfferingPrices* are fetched from the marketplace, which are used in combination with information obtained directly from the MANO layer to ensure that the instantiation and the agreement are aligned. Licensing expiration checks are performed and lastly, checks against the licensing constraints are evaluated based on the instantiation information and provided records from neighbouring eLMAs.
- For those cases in which the instantiation is permitted, watchers remain actively keeping track of the status and licensing metrics of the xNFs. Additionally, it continues to check for expiration and constraints breaches of the license for the full lifecycle of the instance.

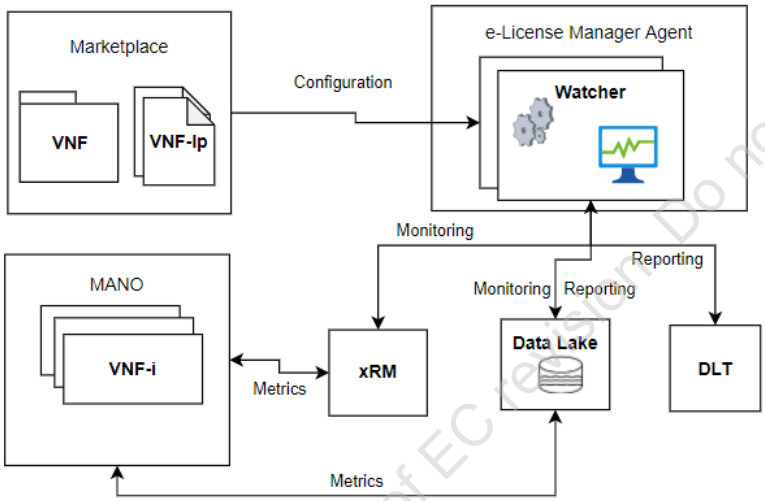

Figure 4-2 e-Licensing Manager watchers

Participating stakeholder will be part of the 5GZORRO ecosystem, trading with the software products in a trusted and secure marketplace with a minimum human intervention. Resulting from this marketplace, 5GZORRO achieves one of its targets which is to enable a way to share assets between different parties. Therefore, the role of service (one or more xNFs) providers may be played by the software vendor directly or by an additional player who composes end to end services as a heterogenous bundle of software from the marketplace and even from their own. Service consumers and providers are assisted by the e-Licensing Manager (eLM) to interact in a transparent, flexible and secure way.

At this point, the e-Licensing Manager is offering the following functionalities to 5GZORRO:

- Enables full automation. There is no need of human intervention, programmatic interfaces are released to interact with the platform.
- NFV Orchestrator agnostic. This provides the capability to deploy the software and proceed with their control independently from the underlying virtualization infrastructure.
- xNF compatibility. Its integration with other components in the 5GZORRO ecosystem allows it to adapt its monitoring workflow for any virtualized NF onboarded to the Platform.
- Infrastructure agnostic. It only needs to run in one location within each administrative domain, therefore there is no need to install dedicated software on the hardware (VIMs) managed by the NFVO.
- Cross-stakeholder e-Licensing Management. Each administrative domain that joins the 5GZORRO ecosystem will be provided with its own instance of the e-Licensing Manager Agent (eLMA) while software vendors will still have a global view of the usage and compliance with the restrictions agreed in the associated smart contract, regardless of where and in how many domains their software is running.

- No need of periodical synchronizations to external endpoints. The usage of the software is tracked in the platform, and it is the platform who generates the bill regarding the usage.
- Metrics used to record the usage of an xNF are customizable by the software vendor to best reflect their business plan. They can be system metrics (e.g., time of use) or application metrics (e.g., number of active users). This allows software vendors to configure the best approach to monetize their product [47] (refer to D3.3):
    - Flat rate is defined by a fixed price for a set of features of the VNFs. Configured via a *ONE TIME productOfferingPrice*.
    - Pay-as-you-Grow is defined by a price that varies based on the usage of the VNFs. Usage can be determined based on various metrics of the VNFs like number of instances or an application-level metric. Configured via a *USAGE productOfferingPrice*.
    - Subscriptions are similar to flat rates but the right to use the purchased VNF needs to be renewed at a known interval. Configured via a *RECURRING productOfferingPrice*.
- Small footprint. Its cloud-native character enables a way to automatically and dynamically balance the resources consumed to correctly offer its functionalities according to the amount of NF instances being monitored.

The KPIs related to the eLM framework and the technical details about how they are overcome are detailed below:

- **Instantiate Network Services with VNFs from diverse providers (KPI target: use eContract to include VNF licensed by at least 3 different providers)**. Smart contracts reflect the legal agreements between unlimited stakeholders, and part of the legal agreements are the licensing terms in case of licensed software. The granularity of the software product can be related to a single network function or can potentially be composed of several software products from different vendors. To achieve this, the 5GZORRO Marketplace allows the configuration of the smart contract and the e-Licensing Manager translates it to monitoring actions at the same level of granularity.
- **Enable the creation of license agreement templates associated to VNF/NS instances (KPI target: create templates attached to eContract detailing name, context, license conditions, negotiation goal and constraints)**. The core technologies for the licensing agreements and control are blockchains and smart contracts, because of the benefits they provide to the framework. Transparency and security are granted in the agreements where providers and consumers sign the terms of the usage and constraints. Licensing terms do not need to follow a fixed subscription model approach, as is typically used in Software-as-a-Service (SaaS) models. In order to deliver a flexible tool for onboarding the network functions, several business models will be supported by the VNF/CNF offer information model detailed in D3.3.
- **Generate vendor independent license token to manage location independent VNFs from 3rd party edge to core datacenter (KPI target: license service creates generic tokens to latter run any vendor VNF across at least 2 network segments)**. The mechanisms for licensing control are metric-based and NFV orchestrator agnostic. Software vendors are permitted to add one or more licensing strategies to their products through a *ProductOfferingPrice* object in the Marketplace, which will provide a token that matches their software product (*ResourceSpecification* or *ServiceSpecification* object in the Marketplace) with a pricing specification that describes how the software shall be monitored and what restrictions to its usage are applicable for the specific token. The location where the software can run is by no means enforced on the specifications either.

eLMA is built as a collection of loosely coupled microservices oriented to be deployed with the minimal footprint in a Cloud-Native environment such as Kubernetes using Helm. Based on the business models previously defined in D2.4, eLMA has been designed to provide the maximum flexibility to the system. For that, a set of functional blocks have been defined (dark blue boxes in Figure 4-3) which are later implemented in one of the three main microservices deployed (light blue boxes in Figure 4-3). The Figure does not show

the database being used (MongoDB with a MongoExpress UI) and the message broker used to exchange events (RabbitMQ) since those services are shared with other components of the 5GZORRO Platform.

1. The translator functional block makes the map between the business agreements and the configuration of the processes that the eLMA watchers should execute. This includes fetching a metric, checking for a specific restriction on their usage etc.
2. The DB Manager ensures that the information stored is in the correct format, contains the required details and validates them.
3. The Watcher LCM detects that there is new information coming from a new registration of a license and proceeds to request the generation of the new watchers if required. It is a scheduled process that executes the watcher's processes at the configured frequency and informs back to the eLMA Rest with the obtained results and status. It also triggers events when an external component needs to be informed.
4. The Watcher Generator is the one that deals with the actual instantiation of a new process (watchers) and its configuration using the relevant configuration based on what the Translator extracted from the Marketplace Information Models.
5. The Infrastructure Manager offers a collection of libraries that allow the actual watcher to interact with the MANO or CNI where the licensed-xNF is deployed.
6. The Events handler assist others with asynchronous requests between the internal components of eLM.
7. The Notification Manager offers an interface with 5GZORRO components that need feedback from eLMA in some way.

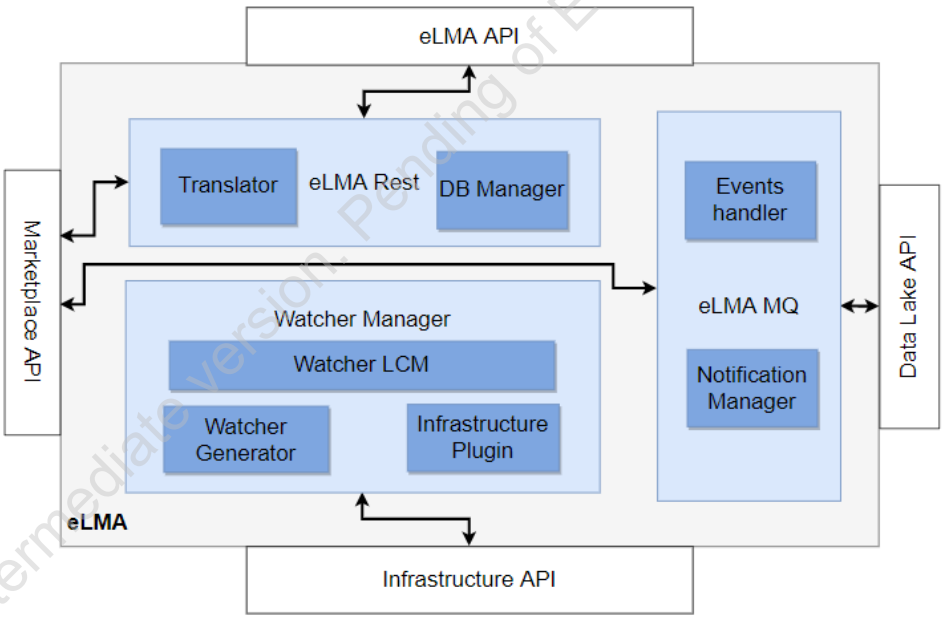

Figure 4-3: e-Licensing Manager Agent components

Complementary to Figure 4-3, Figure 4-4 shows what components of the 5GZORRO Platform are being used on each one of the APIs depicted in the former Figure. Notice that even though eLMA consumes external APIs (defined in this same document in their respective sections) on different internal components, the only exposed API is managed by the eLMA Rest microservice.

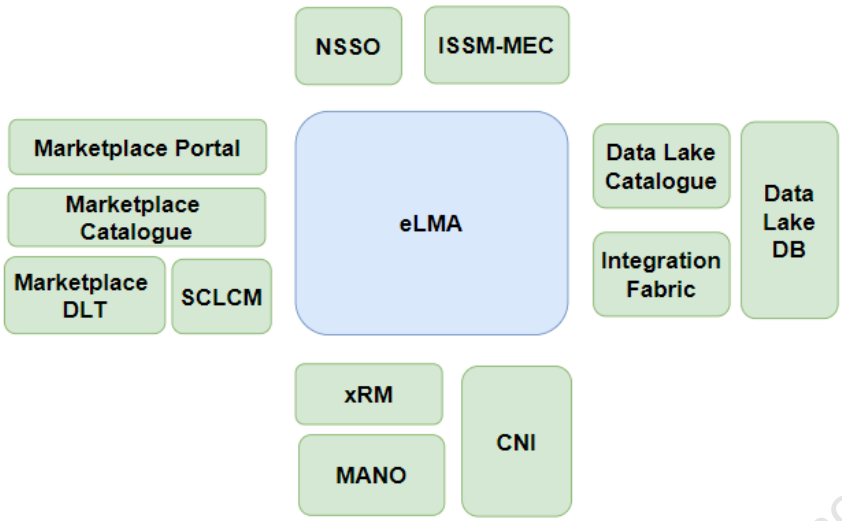

Figure 4-4: e-Licensing Manager Agent interfaces

To support these capabilities, Table 4-12, Table 4-13and Table 4-14 introduce more details about the necessary operations to cover the e-Licensing Management service.

Table 4-12: Definition of e-Licensing Management service (per-domain)

Service name: **e-Licensing Management**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Check licensing	**M**	Capability of the technical orchestrator (NSSO or ISSM-MEC) to notify the e-Licensing Manager about a licensing control that needs to be checked in a specific software resource which is about to be deployed.
Get licensing resources	**M**	Retrieve the licensing terms for the service from the Marketplace
Get application metrics	**M**	Licensing terms may rely on application metrics which are provided in the data lake through the MDA.
Licensing watchers LCM	**M**	Create, remove, update the licensing watchers to observe the xNF licensing events in the virtualized infrastructure
Get instance status	**M**	Watchers need to check the status and information of running xNFs to ensure that there is not a breach on the licensing terms.
Persist an action to DLT	**M**	Create the action record request in the DLT.
Notes		
none		

Table 4-13 Definition of e-Licensing Management service (cross-domain)

Service name: **e-Licensing Management**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Check licensing synchronization	**M**	Capability to communicate with analogous instances of e-Licensing Management Service in a different administrative domain to ensure that the licensing constraints are verified in an aggregated way.
Licensing notification	**M**	Manage ACK/ERROR notifications of the licencing actions and other relevant events to the stakeholders.
Notes		
none		

Table 4-14: Definition of e-Licensing Manager service interfaces

Operation name: **checkLicensing**		
Description	Notification to the e-Licensing manager to make it aware that a new product offering license must be verified.	
Input Parameters	**Type**	**Description**
productDID	String	An id of the offering in the 5GZORRO platform. It shall contain valid license specifications that match with the other input parameters.
nsDescriptorId	String	Software-vendor-assigned id which needs a license verification
nsInstanceId	String	MANO-assigned id from an instance of a xNF that needs a license verification.
Output Parameters	**Type**	**Description**
operationId	String	eLMA-assigned id of the checkLicensing operation required further requests
statusCode	Integer	HTTP status code representing the result of the operation.
responseType	String	Contains REGISTRATION_FINISHED when responding to a checkLicensing requests.
Info	List (String)	Debug information that complements the response when statusCode != 200
Notes		

Operation name: **getLicensingResources**		
Description	Obtain all resources associated to a given offering which are required to verify the licensing terms.	
Input Parameters	**Type**	**Description**
productDID	String	An id of the offering in the 5GZORRO platform. It shall contain valid license specifications that match with the other input parameters.
Output Parameters	**Type**	**Description**
productOfferings	List	Resources from the Marketplace about a transaction
productOfferingPrices	List	Resources from the Marketplace about the licensing configuration for a transaction
Notes		

Operation name: **getApplicationMetrics**		
Description	Obtain all application metrics from the data lake to keep track of the license monitoring metrics and restrictions.	
Input Parameters	**Type**	**Description**
nsInstanceID	String	Identification of the application exposing metrics
Output Parameters	**Type**	**Description**
Metrics	List	Collection of metric values with the format provided by MDA. Keys of interest for eLMA are *instanceid, metricname, metricvalue, productid*
Notes		

Operation name: **licensingWatcherLCM**		
Description	Based on the eLMA IM it will create or update the corresponding watcher processes and return back watcher's findings in the same objects	
Input Parameters	**Type**	**Description**
Descriptor	Object	eLMA descriptor object (Section 5.8)
License	Object	eLMA license object (Section 5.8)
Output Parameters	**Type**	**Description**
Descriptor	Object	eLMA descriptor object (Section 5.8)
License	Object	eLMA license object (Section 5.8)
Notes		

Operation name: **updateInstanceStatus**		
Description	Save the latest status and default metrics from an instance of a licensed xNF.	
Input Parameters	**Type**	**Description**
instanceId	String	Identification of the instance in the MANO/CNI layer
Output Parameters	**Type**	**Description**
Instance	Object	eLMA instance object (Section 5.8)
Notes		

Operation name: **persistAction**		
Description	Create the action record request in the DLT	
Input Parameters	**Type**	**Description**
Action	Object	eLMA action object (Section 5.8)
Output Parameters	**Type**	**Description**
statusCode	Integer	HTTP status code resulting from the request
Info	String	Debug information for the cases of errors or request rejected by the DLT
Notes		

Operation name: **checkLicensingSync**		
Description	Inform about the status of the restrictions for a given license to a neighbouring eLMA instance that is attending a checkLicensing request	
Input Parameters	**Type**	**Description**
offeringPriceId	String	Related id to search for among all eLMA license objects registered within this domain.
restriction	String	Name of one of the restrictions which current status within this domain is required.
Output Parameters	**Type**	**Description**
value	String	Value of the metric associated to the restriction at this moment in this domain
Notes		

Operation name: **licensingNotification**		
Description	Attend and perform notifications to the stakeholders that the eLMA is connected to.	
Input Parameters	**Type**	**Description**
offeringPriceId	String	Related id to search for among all eLMA license objects registered within this domain.
restriction	String	Name of one of the restrictions which current status within this domain is required.
Output Parameters	**Type**	**Description**
action	Object	eLMA action object (Section 5.8) which contains the trigger event and information
Notes		

5 Updated Information Elements

5.1 5G-enabled Trust and Reputation Management Framework Information Model

Due to the fact that no trust information models are available in state of the art, this section proposes an original UML design for the trust and reputation management framework. In this vein, the information model for the 5G-TRMF is fundamentally built up from the entities and characteristics of a trust model. Thus, Figure 5-1 depicts some of the most relevant parameters related to trust information model, grouped in the tables below. First and foremost, Table 5-1 introduces a subset of generic characteristics that may be associated with a trust instance regardless of its enforcement environment. The second table, Table 5-2, represents the trusted information associated with the service or resource provider, i.e., the entity with which we want to establish a relationship of trust. Some of the parameters presented in this table are acquired through the resource offer information model [46], [47] and service offer information model [48] addressed in Deliverable 3.1. By means of these parameters, the trust model will determine the trust level and its score on a stakeholder resource or service offered in the marketplace. Lastly, Table 5-3 describes trustworthy data of the source entity, i.e., the stakeholder who is interested in establishing a connection. Note that, the information models are used internally by the stakeholders that make up the 5GZORRO ecosystem. Furthermore, Table 5-1, Table 5-2, and Table 5-3 are being updated from the previous versions presented in D4.1 [49] and D4.2 [50], since new characteristics were required during the development and testing phases.

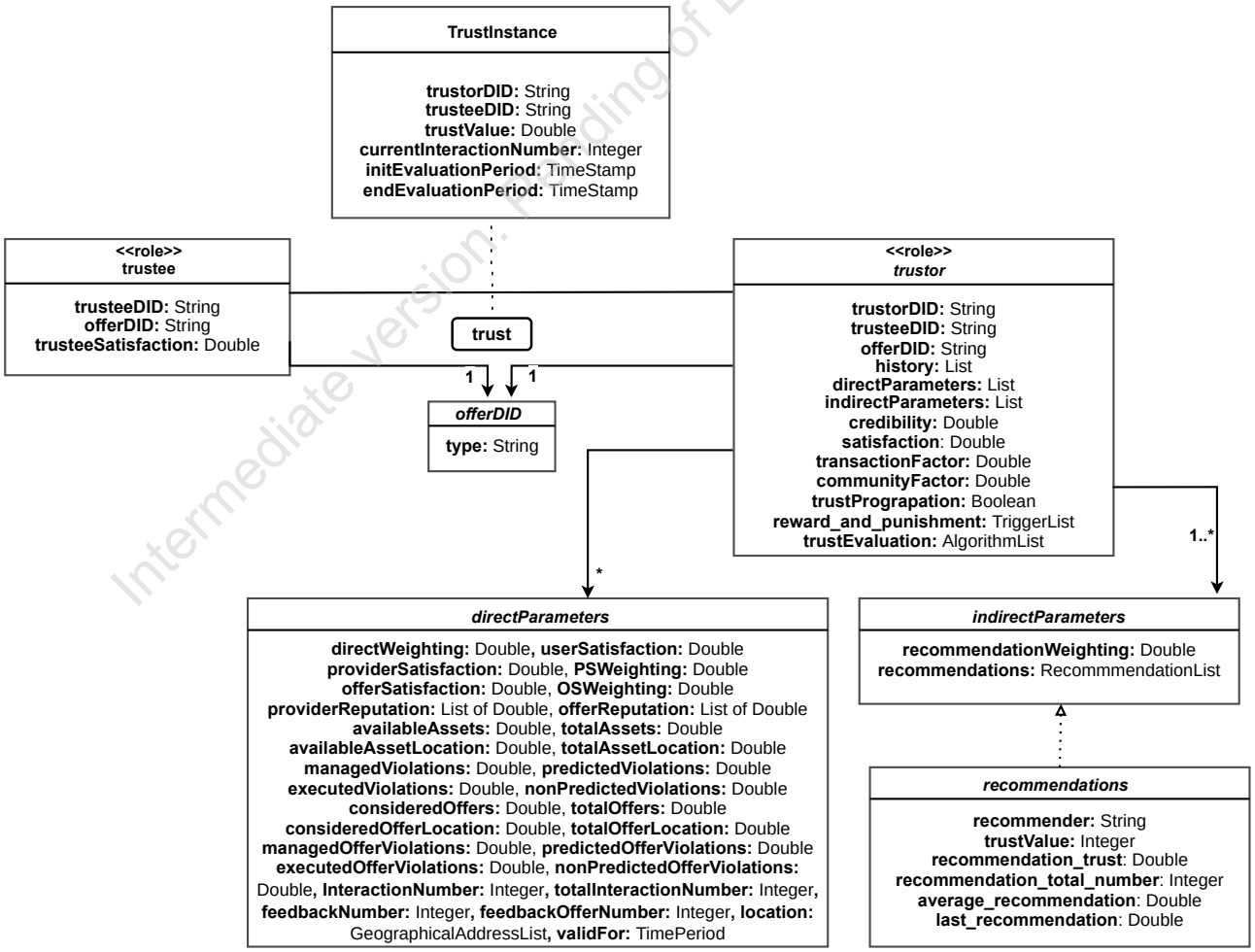

Figure 5-1 : UML diagram of Trust and Reputation Management Framework

Table 5-1: Trust and Reputation Management Framework Instance Information Model

Parameter	Type	Description
trustorDID	String	Unique identifier for a resource or service consumer.
trusteeDID	String	Unique identifier for a resource or service provider.
trustValue	Double	Current trust value assigned.
currentInteractionNumber	Int	Number of interactions between the trustor and the trustee.
initEvaluationPeriod	TimeStamp	The time when trust value was generated.
endEvaluationPeriod	TimeStamp	The time when trust value will be over and has to be reassigned if required.

Table 5-2: Trustee Entity Information Model

Parameter	Type	Description
trusteeDID	String	Unique identifier for a resource or service provider.
offerDID	String	Unique identifier for a particular product offer of the provider.
type	String	Kind of offer (RAN, spectrum, VNF/CNF, slice, or edge)
trusteeSatisfaction	Double	Truestee's satisfaction after x interactions with other providers.

Table 5-3: Trustor Entity Information Model

Parameter	Type	Description
trustorDID	String	Unique identifier for a resource or service consumer.
trusteeDID	String	Unique identifier for a resource or service provider.
offerDID	String	Unique identifier for a particular product offer.
type	String	Kind of offer (RAN, spectrum, VNF/CNF, slice, or Edge)
history	List (double)	Set of trust evaluations about an entity.
directParameters	List of key-value features	Dictionary with direct trust data to calculate trust level.
directWeighting	Double	Direct weighting parameter.
userSatisfaction	Double	Internal assessment of the service or resource provided by a stakeholder (trustor).
***provider**Satisfaction*	Double	Trustor satisfaction in a third-party provider (trustee).
PSWeighting	Double	Weighting factor ∈ [0,1], PS + OS = 1
offerSatisfaction	Double	Trustor satisfaction in a particular kind of offer of a third-party provider.
OSWeighting	Double	Weighting factor ∈ [0,1], PS + OS = 1
providerReputation	List (double)	Set of previous trust evaluations about a provider.
offerReputation	List (double)	Set of previous trust evaluations about a specific kind of offer of a provider.

availableAssets	Integer	The available assets (services and resources) of the trusteeDID when the trustor is determining the reputation.
totalAssets	Integer	The total assets of the trusteeDID when the trustor is determining the reputation (active and inactive).
availableAssetLocation	Integer	The available assets of the trusteeDID, at a specific location, when the trustor is determining the reputation.
totalAssetLocation	Integer	The total assets of the trusteeDID, at a specific location, when the trustor is determining the reputation (active and inactive).
managedViolations	Integer	The total number of predicted SLA violations that were finally managed successful, associated with the trusteeDID.
predictedViolations	Integer	The total number of predicted SLA violations associated with the trusteeDID.
executedViolations	Integer	The total number of predicted SLA violations that were finally managed unsuccessful (violation), associated with the trusteeDID.
nonpredictedViolations	Integer	The number of SLA violations that were not predicted and turned out to be SLA violations, associated with the trusteeDID.
consideredOffers	Integer	The number of offers considered by the Smart Resource and Service Discovery (SRSD), for a particular type of offer, from the trusteeDID when trustor is determining the reputation.
consideredOfferLocation	Integer	The available number of a particular offer type from the trusteeDID when trustor is determining the reputation.
totalOfferLocation	Integer	The number of offers considered by the Smart Resource and Service Discovery (SRSD) for a particular type of offer from the trusteeDID, at a specific location, when trustor is determining the reputation.
managedOfferViolations	Integer	The available number of a particular offer type from the trusteeDID, at a specific location, when trustor is determining the reputation.
predictedOfferViolations	Integer	The total offer number (for a particular kind of offer) of predicted SLA violations that were finally managed successful, associated with the trusteeDID.
executedOfferViolations	Integer	The total offer number (for a particular kind of offer) of predicted SLA violations associated with the trusteeDID.
nonpredictedOfferViolations	Integer	The total offer number (for a particular kind of offer) of predicted SLA violations that were finally managed unsuccessful (violation), associated with the trusteeDID.
interactionNumber	Integer	The total offer number (for a particular kind of offer) of SLA violations that were not predicted and turned out to be SLA violations, associated with the trusteeDID.
totalInteractionNumber	Integer	Number of interactions carried out by the trusteeDID with the other domains.

feedbackNumber	Integer	Number of feedbacks made by other providers about the trusteeDID.
feedbackOfferNumber	Integer	Trustor satisfaction in a third-party provider (trustee).
location	List of GeographicalAddress objects [45]	It constitutes a group of GeographicalAddress.
validFor	TimePeriod	The period for which this resource or service is valid.
indirectParameters	List of key-value features	Dictionary with indirect trust data to calculate trust level.
recommendationWeighting	Double	Recommender's weighting parameter(s).
recommendations	List of recommendations	Set of recommendation about a third entity from one or more external entities.
recommender	String	Unique identifier for a recommender
trust_value	Int	Final trust score ranged from 0.0 to 1.0.
recommendation_trust	Double	Trust level of the trustor in the recommendation.
recommendation_total_number	Int	Number of recommendations of a given trustee.
average_recommendation	Double	Average of all recommendations.
last_recommendation	Double	The last recommendation.
credibility	Double	Factor that determines how accurate the recommendations are.
satisfaction	Double	Trustor satisfaction in a specific trustee.
transactionFactor	Double	Intra or inter-domain trust score and parameterTuple propagation (0 means intra, 1 means inter)
communityFactor	Double	It indicates the triggers to recompute trust score.
trustPropagation	Double	It identifies different evaluation algorithms.
reward_and_punishment	List of triggers	Increase or decrease score based on current events such as security, time-decay, etc.
trust_evaluation	List of algorithms	It identifies different evaluation algorithms such as PeerTrust reputation model.

5.2 Trusted Execution Environment Security Management Information Model

Instead of developing a solution from scratch using proprietary hardware drivers for TEEs enabled microprocessors and implementing the attestation, key management and provisioning systems, our focus is to deploy the application-level security module as a service, enabling its integration with DevOps frameworks, such as Kubernetes, as do the other software modules in 5GZORRO.

Secure Linux Containers (SCONE) [9] is a framework built on top of Intel's TEE solution, in the context of other H2020 projects that not only abstracts specific implementation details of the secure enclave, but also provides encryption at rest, in transit and during runtime without requiring source code changes supporting most modern program languages. SCONE also has built-in attestation and key provisioning modules, allowing the application developers to focus on the orchestration and configuration of the security management solution and not on the security-solution implementation.

SCONE contains 4 components:

1. **CAS (Configuration and Attestation Service)**: a remote service deployed in a container provided by SCONE that generates and stores the application secrets. These secrets are provided to the application once attestation is successful, considering the applications key's access policy. The keys are used to decrypt the binary inside the secure container, decrypt files stored in the filesystem and securely retrieve environmental variables.
2. **LAS (Local Attestation Service)**: a service that runs locally, alongside the enclave, deployed in a container provided by SCONE that creates the quote to be verified by CAS.
3. **Session**: Instance of a security policy that describes all the security-relevant details of a SCONE application (docker image to be used, command to be executed, unique enclave signature). The security policy contained in a session can be updated to fit the needs of the application during runtime.
4. **Docker container**: the trusted application intended to run in a secure enclave.

Interaction with SCONE is performed using the SCONE CLI, a stateful CLI that preserves state between invocations, such as attestation information and identity keys. To run a microservice (a session) inside a remote secure enclave using SCONE one must:

1. Start a docker image of a remote CAS. The remote CAS should then be attested and provisioned;
2. Start a docker image of a LAS. Afterwards, the LAS would generate a quote that will be verified by the remote CAS;
3. Create and post application sessions, to be run on the secure enclave

After starting the remote CAS container, attestation must be performed to build a trust relationship between the user and the remote CAS, with validation of the CAS information by the user. If the attestation process is successful, the SCONE CLI stores all the information required to safely communicate with the CAS.

The following command attests a remote CAS that is running with the address *cas_address*, with the *cas_hash*, the expected CAS public key hash, and, *cas_sw_hash*, the expected CAS software public key hash. If the CAS software is signed by a custom signer, instead of the default provided by SCONE, the signer public key (*signer_pub_key*), the Independent Software Vendor Product ID (*isvprodid*) and the Independent Software Vendor Security Version Number (*isvsn*) should be also provided. More details about these parameters can be found in Table 5.4.

```
scone cas attest <cas_address> -c <cas_hash> -s <cas_sw_hash> --mrsigner <signer_pub_key>
--isvprodid <isvprodid> --isvsn <isvsn>
```

After attestation, the CAS needs to be provisioned and configured, which can be done with the command below.

```
scone cas provision <cas_address> -c <cas_hash> --config-file <config_file> --token <token>
[with-attestation]
```

Where *cas_address* represents the CAS address, *cas_hash* the expected CAS public key hash, *config_file* the path to the configuration file where the information of the CAS will be updated and stored once configuration is successful and, *token,* the provisioning token to allow the CAS to verify its owner.

Alternatively, the CAS can be attested, provisioned and configured by using the *with-attestation* flag during provisioning, which attests the CAS and then verifies if the CAS is running in a secure enclave, and therefore it if it is safe to transfer the confidential information from the user to the CAS. If this flag is used, all the parameters used for attestation, detailed in Table 5.4, can also be used. These parameters are summarized in Table 5.5.

After the CAS is running, the LAS docker image must be started, establishing a secure communication with the CAS. After both the CAS and LAS have been set up and are running, a session must be created and posted to the CAS, using the command:

```
scone session create --cas <cas_address> <policy_file>
```

As before, the *cas_address* is the address of the remote CAS to use. The *policy_file* is the path to the file that contains the session descriptions that entail all security-relevant details of a SCONE application. These parameters can be consulted in Table 5.6.

Table 5-4: SCONE CAS CLI parameters for attestation

Parameters	Type	Description
cas_address	String	remote CAS address to attest.
cas_hash	String	Expected CAS public key hash.
software_hash	String	Expected CAS software public key hash.
signer_pub_key	String	Alternative MRSIGNER public key used to verify the CAS software signature instead of the default SCONE key.
isvprodid	Integer	Independent Software Vendor Product ID (ISVPRODID) to be verified.
isvsn	Integer	Independent Software Vendor Security Version Number (ISVSVN) to be verified.

Table 5-5: SCONE CAS CLI parameters for provisioning

Parameters	Type	Description
cas_address	String	remote CAS address.
cas_hash	String	Expected CAS public key hash.
config_file	String	Path to the config file containing the server-side CAS configuration.
token	String	Public key of the alternative CAS signer to be used for attestation.

Table 5-6: SCONE CAS CLI parameters for creating new sessions

Parameters	Type	Description
cas_address	String	The address of the remote CAS to use.
policy_file	String	The path to the file containing the session description.

5.3 Security Analysis Service Information Model

The information model of the security analysis service module depends on the available information of Network Services i.e., the 5G core network or mobile edge infrastructures requested by the 5GZORRO resource consumers. Hence, it is not fixed but dynamically adjusted based on the information that the module receives. Moreover, it is based on the Zeek network security monitor types [37]. Nevertheless, irrespectively from the information availability this module also creates log files based on exchanged packets. Such information is provided in the following table:

Table 5-7: Information model of the security analysis service

Parameter	Type	Description
ts	Time	The timestamp in which the packet is received.
uid	String	Unique identifier of the connection that is used for packet exchange.

srcIP	Addr	The source IP address from which the packet is transmitted.
srcPort	Port	The source port from which the packet is transmitted.
dstIP	Addr	The destination IP address in which the packet is received.
dstPort	Port	The destination port in which the packet is received.
l2proto	ENUM	The lower layer protocol (e.g., Ethernet, WiFi) used for packet exchange.
l4proto	ENUM	The transport network protocol (e.g., TCP, UDP) used for packet exchange.
l7proto	ENUM	The application network protocol (e.g., TCP, UDP) used for packet exchange.
service	String	The network service (e.g., DNS, DHCP, HTTP) that used to support the exchanged packet.
txBytes	Byte Array	The number of transmitted bytes from all the packets.
rxBytes	Integer	The number of received bytes from all the packets.
pktType	String	The type of command that is included inside the exchanged packet.
pktLength	Integer	The length of the actual data that is included inside the exchanged packet.
pktData	String	The main part of the packet containing the actual data that are exchanged.
statusCode	Integer	The status code of the exchanged packet indicating success, acknowledgment reception, error or malformed structure.
fuid	String	The file identifier that may be present inside the exchanged packet.
fname	String	The file name that may be present inside the exchanged packet.
errorDescr	String	The description of possible error or malformed structure in the exchanged packet.
certId	String	The id of security certificate that is associated with the exchanged packet.
certIssuer	String	The issuer of the security certificate that is associated with the exchanged packet.

5.4 VPN-as-a-Service Information Model

The VPN-as-a-Service model is basically composed by the configuration parameters required to setup the VPN service and access to it. Some of the parameters, like public keys and service IPs and ports will be public, while private keys and allowed IPs will be used in the authentication process and are private. Next, the current information models identified will be specified. Note that some parameters may vary based on the actual VPN solution utilized for the VPN-as-a-Service.

Table 5-8: VPN server configuration information model

Parameter	Type	Description
PrivateKey	String	VPN server private key for service authentication.
PublicKey	String	VPN server public key for service authentication.
ListenPort	String	Network port where the VPN service is exposed.
ServerIP	Addr	Public IP of the VPN service.
ClientsList	List	List of authorized clients, including the public information of each client (*PublicKey, DID* and *AllowedIPs*).

Table 5-9: VPN client configuration information model

Parameter	Type	Description
PrivateKey	String	VPN client private key for authentication.
PublicKey	String	VPN client public key for authentication.
EndPoint	Addr	IP and port where the VPN server to connect is located.
AllowedIPs	Addr	IP (or range) that the peer can have when connecting.

5.5 Any Resource Manager Information Model

xRM maintains a limited set of information related to the resources belonging to the underling virtualized infrastructure, since the different resource specification are stored in the catalogues of the respective orchestrators, controllers and managers. Internally xRM maintains a table that contains just a couple of entries, described in Table 5-10.

Table 5-10: Any Resource Manager – Resource information model

Parameter	Type	Description
ResourceID	UUID/String	Unique id of the resource indexed form the RM. The id is locally unique.
ResourceSpecID	UUID/String	ID of Resource Specification (TM Forum) object result of the translation process to support the creation product offers. The presence of such ID implies that the object resources has been translated and stored in the Resource and Service Offer Catalogue

With respect to the resource that can be translated by the xRM, each of them implements its own information model, aligned with a given standard.

Network services, Virtual and Physical networks functions, and their respective descriptors are aligned to the standard ETSI SOL 006 [38], which follows the models previously defined in ETSI GS NFV-IFA 11 [39] (VNF(D), NS(D)) and ETSI GS NFV-IFA 14 [40] (PNF(D)). MEC Applications follow ETSI GS MEC 010-2 [41]

Network slice model has been already provided in D2.2 and it is aligned to GSMA General Slice Template (GST) [42].

For what concerns the Radio resources, there are some standardized information models to abstract them as virtual resources. For instance, 3GPP has an exhaustive information model to manage 5G NR resources [43], which does not include non-3GPP technologies as Wi-Fi or spectrum resources. A simplified set of information models, defined for the purpose of 5GZORRO, are described in the following tables.

Table 5-11 Radio Spectrum Resource Information Model

Parameter	Type	Description
slotID	UUID/String	Unique ID of the spectrum slot.
regulator	UUID/String	Unique ID of the National Regulator of the spectrum.
operator	UUID/String	Unique ID of the licensee of the spectrum, typically an MNO.
place	Object	TMForum's Geographic Address object, which includes the GPS coordinates of the covered area

duplexMode	String	The operation mode to be used in the spectrum resource can be TDD or FDD.
technology	String	Determines the technology to use with the spectrum resource
startDlFrequency	Numeric	The start Downlink frequency (MHz) of the spectrum resource.
endDlFrequency	Numeric	The end Downlink frequency (MHz) of the spectrum resource.
startUlFrequency	Numeric	The start Uplink frequency (MHz) of the spectrum resource.
endUlFrequency	Numeric	The end Uplink frequency (MHz) of the spectrum resource.

Table 5-12 RAN Resource Information Model

Parameter	Type	Description
ranResource	UUID/String	Unique ID of the RAN resource.
ranType	String	The type of RAN resource. Possible values may include cell, access point or backhaul link, and more.
place	Object	TMForum's Geographic Address object, which includes the location of the RAN resource in GPS coordinates
technology	String	The wireless technology of the RAN resource (WIFI5, WIFI6, LTE, NR).
duplexMode	String	Duplex operation mode can be either TDD or FDD
operationBand	List (Numeric)	List the supported operation bands (Cellular) or channel number (WiFi).
centralDLFrequency	List (Numeric)	The central downlink frequency of the radio resource
centralUlFrequency	List (Numeric)	The central uplink frequency of the radio resource
bandwidth	List (Numeric)	Lists the supported system bandwidths of each operationBand.
txPower	Numeric	The maximum transmission power in dBm.

5.6 Network Slice and Service Orchestration Information Model

The offers available on the 5GZORRO Marketplace will be mapped and translated to information elements supported by the Network Slice and Service Orchestration in order to be deployed. As described in Section 4.2. This latter module shall support two main specification models: (i) Vertical Service Blueprints/ Descriptors (VSBs/VSDs) and (ii) NEtwork Slice Type (NEST).

VSBs are high level templates which allow to describe services without requiring an in-depth knowledge of how the service is deployed. VSB serve as a formal and structured way to specify the service a vertical aim to deploy, and that can be used to determine the end-to-end network slice of the service. It is mainly composed by:

- Atomic functional components: represent the main blocks composing service. Usually, atomic components represent the Network Functions (both physical and virtual), but they can also represent more complex structures such as NFV-Network Services or even Vertical-subservices composing the end-to-end vertical service

- End points: End points are used to express how the different atomic components are inter-connected and therefore able to interact, and allow to specify properties which can be used to specify service SLA related constraints.
- Connectivity services: connectivity services model the relationship between endpoints and therefore determine how the atomic components are connected to each other.

- Parameters: Establish the input parameters that allow to customize the service and the required SLA.

The Table 5-13, extracted from [52], describes the structure of a VSB.

Table 5-13 VSB Information model

Parameter	Type	Description
blueprintId	String	Unique Identifier for the VSB.
version	String	A version number.
name	String	Name for the VSB.
description	String	Short description of the VSB.
parameters	List	List of parameters that describe the service constraints the vertical has to fill (i.e., valorize) when filling the VSB to produce a new VSD. The list provides for each parameter its name, type, description, and the field of applicability.
atomicComponents	List	List of atomic functional components (i.e., network functions and virtual applications in general) needed to implement the VSB.
endPoints	List	Specification of connection endpoints. They can be internal or external.
connectivityServices	List	List of virtual links and their relevant end points. Virtual links describe how the atomic functional components are connected.
serviceSequence	List	Description of how traffic flows among atomic components, supporting also multicast scenarios.
configurableParameters	List	Parameters that can be configured at instantiation time by the user for a specific instance of service derived from the given blueprint.

The VSDs allow to customize the VSB to the specific needs of the vertical service, by providing specific values to the parameters. Table 5-14, extracted from [52], describes the structure of a generic VSD.

Table 5-14 VSD information Model

Parameter	Type	Description
vsdId	String	Unique identifier for a VSD.
name	String	Name provided by the vertical for this VSD.
description	String	Short description of the VSD.
version	String	A version number.
blueprintId	String	The identifier of the blueprint from which this VSD was derived.
Sst	Enumerate: eMBB, URLLC, mIoT	Slice Service Type, as defined by 3GPP. Allowed values are therefore: eMBB, URLLC, mIoT.

serviceConstraints	List	List of service-related constraints that have to be fulfilled by vertical instances created starting from the given descriptor (e.g., geographical constraints, sharing rules, etc.).
qosConstraints	List	List of QoS related constraints that have to be fulfilled by vertical instances created starting from the given descriptor. This attribute contains the parameter types and values as filled by the vertical according to the parametrization of the related VSB.

In addition to the service definition based on VSBs/VSDs, the Network Slice and Service Orchestration module shall also support definitions based on the GST/NEST approach proposed by GSMA [42]. The information model and the relevance within 5GZORRO of these two latter elements has already been reported in [43] .

The high-level service specifications shall be translated into resource-oriented specifications to be requested to other components of the 5GZORRO platform. For instance, a possible information model for the network slices can be adopted from the 3GPP specification TS 28.541 [45].

5.7 Network Service Mesh Manager Information Model

In this section is defined the information model of the internal Connectivity Store of the NSSM (see Figure 4-)

Table 5-15 Connection Element (CE) information model

Parameter	Type	Description
id	String	Unique identifier of the connection. The uniqueness property should be maintained on the local domain only and the id could be not guaranteed on the different domains involved, if any.
Endpoints	List	List of endpoints object. See Table 5-16.

Table 5-16 Endpoint Element (EE) information model

Parameter	Type	Description
id	String	Unique identifier of the endpoint. The uniqueness property should be maintained on the local domain only and the id could be not guaranteed on the different domains involved, if any.
domain_id	String	Identifier of the domain the endpoint belongs to.
VRT_platform_info	object	Set of parameters specific for the Virtualization platform.
VPN_info	object	Set of parameters characterizing the VPN side (client or server).

Table 5-17 Virtualization platform information model

Parameter	Type	Description
type	ENUM	Virtualization platform type: • NFV • CN (cloud-native)
name	string	Virtualization platform name: e.g., k8s, Openstack, vmware, etc.
Management_ip	string	Address for the management of the platform. Needed for configuring the networking.
Entity_type	ENUM	Virtual entity managed: VNF, CNF, etc
Attachment_ip	String	Address used for the stitching.

Table 5-18 VPN configuration information model

Parameter	Type	Description
role	ENUM	Role of the endpoint in the VPN • CLIENT • SERVER
Local_ip	String	Ip of the endpoint in the VPN subnet.
Allowed_ips	List	List of addresses allowed to use the VPN (in case or Role=SERVER).
Exposed_subnets	List	List of exposed subnets through the VPN connection
Remote_server_ip	String	IP of VPN server (in the case of Role=CLIENT).
Remote_public_key	String	Public key of the remote VPN endpoint.
Public_key	String	Endpoint public key.

5.8 e-Licensing Management Information Model

In this section is defined the information model of the internal elements in the e-Licensing Manager which was described in section 4.4.

The eLMA is responsible of the real-time control of the usage of the xNFs in each domain, monitoring the operational usage of the software components inside the domain and storing this usage in the DLT through action objects.

Table 5-19: eLMA license registration Information Model

Parameter	Type	Description
productDID	String	Unique identifier for a resource or service consumer.
timestamps	object	Contains datetimes objects for the keys: createdAt, updatedAt, timeStart, timeEnd
status	Enum	Defines the state of the registration operation: PROCESSING, COMPLETED, FAILED.
offerings	List	List of ProductOffering objects obtained through productDID
licenses	List	List of eLMA License objects
input	Object	Received payload on the checkLicensing endpoint
output	Object	Response provided to the requester

Table 5-20: eLMA Licence Information Model

Parameter	Type	Description
timestamps	object	Contains datetimes objects for the keys: createdAt, updatedAt, timeStart, timeEnd
offeringPriceId	object	ProductOfferingPrice Object obtained from the marketplace
history	List	List of references to this license in previous registrations. Each entry contains a registrationId and offeringId key
kind	Enum	Watcher kind that is monitoring this license: TIME_OF_USE, RECURRING, ONE TIME, N_OF_USER
Descriptor	UUID	Reference to an eLMA Descriptor object

Table 5-21: eLMA Descriptor Information Model

Parameter	Type	Description
nfvLevel	Enum	Type of xNF related to this descriptor added as a resource or service Specification model of the marketplace: VNF, CNF, NS
vendorId	String	Identification used in the resource or service Specification model of the marketplace
onboardedId	String	Identification used by the MANO or CNI layer
instances	Object	Collection of eLMA Instance objects organized by Id
history	List	List of references to this Descriptor in previous registrations. Each entry contains registrationId, offeringId and offeringPriceId keys

Table 5-22: eLMA Instance Information Model

Parameter	Type	Description
Status	Enum	Status reported by the MANO or CNI layer
timestamps	object	Contains datetimes objects for the keys: createdAt, updatedAt, timeStart, timeEnd
registrationId	String	Reference to the eLMA registration object that provided this instance
actions	Object	Collection of eLMA actions objects organized by Id

Table 5-23: eLMA Action Information Model

Parameter	Type	Description
Status	Enum	Status reported by the MANO or CNI layer
timestamps	object	Contains datetimes objects for the keys: createdAt, updatedAt, timeStart, timeEnd
instanceId	String	Identification of the NF instance that originated this action
offeringPriceId	String	Identification of the productOfferingPrice that originated this action
offeringId	String	Identification of the productOffering that originated this action
triggerKind	Enum	Reason for the delivery of the action; ERROR, EXPIRATION, SCHEDULED
Status	Enum	Current state of the action; RUNNING, SENT
Metric	Object	Metric value obtained according to the productOfferingPrice containing value and unit keys

6 Conclusions

This deliverable provides the final design of part of the architecture and 5GZORRO core platform. This report covers security and trust orchestration, intelligent and automated slice & service management, and the MANO and slicing tools enhancements, being the final outcome of the design for zero-touch service management with security and trust solutions performed in the project.

The architecture presented in this document is based on deliverables D2.2, D2.4, and D4.1, where the 5GZORRO high-level reference architecture was introduced. This report has carried out an iteration on the basic architecture presented in D4.1 to improve the implementation details and the functionalities associated with 5GZORRO services. To succeed in our commitment, multiple sets of interfaces, information models, and 5GZORRO specific enhancements have been reported for each of the above capabilities. Thus, a summary of objectives and sub-objectives met by the specific contribution of the presented design are provided in Table 6-1 in terms of applicable design artefacts.

The design artefacts described in this deliverable serve as input for the final implementation work that will be carried reported in deliverable D4.3, as well as for the final use case validation activities in WP5.

Table 6-1: D4.4 contribution to 5GZORRO objectives and KPIs.

OBJECTIVE	Target KPIs	Applicable Design Artefact
OBJ-1. Define a system level architecture combining zero-touch automation solutions and distributed ledger technologies to enable a secure, flexible and multi-stakeholder combination and composition of resources and services in 5G networks.	• *Support actual distributed multi-party service and business configurations (KPI target: more than 3 providers/operators of virtualized resources or services for spectrum, radio/edge/core compute & network).*	not applicable (n/a) because not referring to any component described
	• *Inject and process operational service data (configurations and runtime monitoring and logging) into a multi-party 5G Operational Data Lake (KPI target: at least 10 heterogeneous and diverse operational data sets streamed into 5G Operational Data Lake from various data sources, at least one per provider/operator).*	See Sec. 2.1 for 5G-enabled Trust and Reputation Management Framework, Sec. 2.3 for Security Analysis Service and Sec. 4.1 for Any Resource Manager
	• *Expose open APIs to application layer for processing operational data for analytical processes, which discover and "inventorize" various types of resources (KPI target: all external 5GZORRO APIs are exposed via open and public specifications).*	n/a
	• *Automate the overall service lifecycle management with seamless use of heterogeneous virtualization platforms (i.e., VMs and containers, interconnected with various levels and forms of service meshes) across different providers (KPI target: completion of end-to-end provisioning in less than 5 mins, service deletion in less than 1 min).*	See Sec. 3.2 for ISSM, Sec. 3.3 for Intelligent Network Slice and Service optimizer, Sec. 4.1 for Any Resource Manager, and Sec. 4.3 for Network Service Mesh Manager
	• *Support a real-time market for dynamic spectrum allocation allowing business agents to trade on spectrum allocations in space and time (KPI target: Time from transaction to spectrum availability in less than 10 minutes; support of 5GNR, LTE and WiFi technologies).*	n/a

OBJECTIVE	Target KPIs	Applicable Design Artefact
OBJ-2. Design and prototype a security and trust framework, integrated with 5G service management platforms, to demonstrate Zero-Day trust establishment in distributed multi-stakeholder environments and automated security management to ensure trusted and secure execution of offloaded workloads across domains in 5G networks	• *Provide mechanisms for zero touch trust automation in multi-domain scenarios on top of a 5G service management framework (KPI target: to cover up to 4 different stakeholders as part of the automated trust establishment process and to enable its automatic renegotiation when a stakeholder is joining or leaving the trust link).*	See Sec. 2.1 for 5G-enabled Trust and Reputation Management Framework.
	• *Enhance a 5G service management framework enabling the detection of security vulnerabilities and compromises and the provision of a set of potential countermeasures to mitigate them using a zero-touch approach (KPI target: identifying 6 different types of common attacks to software infrastructures and provide a complete set of countermeasures -filter traffic, divert it to a honeynet, send an alert to the system admin, etc.- for each of them).*	See Sec. 2.3 for Security Analysis Service.
	• *Support the integration of zero trust hardware platforms (TEE - Trusted Execution Environments) as a root of trust for the monitoring of information and the establishment of end-to-end secure communications enabling critical workloads to go across different tenants and different stakeholders (KPI target: research on the integration evolution of three TEE platforms --one provided by a project partner-- and two other commercial ones to support a fast and secure establishment of end-to-end cross-slice communications for critical workloads).*	See Sec. 2.2 for Trusted Execution Environment Security Management.
OBJ-3. Define a Smart Contract ecosystem anchored on a native distributed ledger to allow commercial and technical data provided by 3rd-party users to be standardised and mapped into Smart Contracts, which can be initiated "at will" between multiple untrusted parties.	• *Ability for untrusted parties to negotiate, set-up and operate a new technical/commercial relationship via a Smart Contract for 3rd-party resource leasing/allocation with associated SLA (KPI target: Smart Contract for 3 or more untrusted parties).*	See Sec. 2.1 for 5G-enabled Trust and Reputation Management Framework, Sec. 3.5 for Intelligent 3rd Party Resource Planner and Sec. 4.2 for Network Slice and Service Orchestration.
	• *Availability of an Oracle data layer to enable external data sources, processing and results to be requested by SLA smart contracts (KPI target: Oracle data layer accessed by 3 or more parties).*	Part of the Smart Contract DLT capabilities
	• *Enable off-chain processing of transactions through payment channels using smart contract in order to enable faster and cheaper transactions compared to on-chain (KPI target: Twice the number of transactions performed over on-chain).*	Part of the Smart Contract DLT capabilities

OBJECTIVE	Target KPIs	Applicable Design Artefact
OBJ-4. Define solutions for secure, automated and intelligent resource discovery, brokerage and selection, operation with SLA to facilitate workload offloading to 3rd-party resources supporting pervasive computing across multiple 5G domains.	• *Automatically discover and "inventorize" various types of resources (i.e., compute, storage, network at core, edge, far-edge), spectrum and services capabilities from different domains and service providers (KPI target: distribution of resource updates and discovery in less than 10 mins).*	n/a
	• *Implement/correlate technical service configurations and SLA monitoring interactions between multiple parties (KPI target: SLA measurements and validation from at least 3 operators involved in a multi-party service chain).*	n/a
	• *Support intent-based API to guide the AI-driven resource discovery system (KPI target: open 5GZORRO API specification for resource discovery).*	See Sec. 3.5 for Intelligent 3rd Party Resource Planner
OBJ-5. Define and prototype a secure shared spectrum market to enable real-time trading of spectrum allocations between parties that do not have a pre-established trust relationship.	• *Time to process and enforce new spectrum transactions (i.e., from the moment the transaction is settled until the spectrum becomes available) (KPI target: complete new spectrum transactions in less than 10 minutes).*	n/a
	• *Number of transactions per second handled by the market, which will determine the volume of spectrum transactions processed by the market (KPI target: 20 transactions/second).*	n/a
	• *The authenticity of the market agents, preventing double spending that would allow an agent to trade spectrum rights that it does not own (no explicit KPI target: verification of the built-in property of Blockchains).*	n/a
	• *Linkability between market agents and their associated radio access points, which will allow to provide the appropriate spectrum rights to each access point (KPI target: <10M cell towers should be linkable by the system, which is a reasonable EU nation-wide deployment).*	n/a
	• *Ability to enforce the settled spectrum rights and obligations, which will build on lightweight Trusted Execution Environments (TEE) embedded in the radio access points to ensure that the reported spectrum measurements are faithful, and the spectrum allocations settled in the market are enforced (KPI target: Be able to detect spoofing attacks where a base station uses an allocation not authorized by the market).*	See Sec. 2.2 for Trusted Execution Environment Security Management and See Sec. 2.3 for Security Analysis Service.
	• *Agnostic support of various radio technologies, to ensure that the market will work regardless of the considered radio technology (KPI target: 5GNR, LTE and WiFi will be supported).*	See Sec. 4.1 for Any Resource Manager.

OBJECTIVE	Target KPIs	Applicable Design Artefact
OBJ-6. Realize a cloud-friendly network software licensing framework for location independent network appliances execution.	• *Enable the creation of license agreement templates associated to VNF/NS instances (KPI target: create templates attached to eContract detailing name, context, license conditions, negotiation goal and constraints).*	See Sec. 4.4 for e-Licensing Management
	• *Generate vendor independent license token to manage location independent VNFs from 3rd party edge to core datacenter (KPI target: license service creates generic tokens to latter run any vendor VNF across at least 2 network segments).*	See Sec. 4.4 for e-Licensing Management
	• *Instantiate Network Services with VNFs from diverse providers (KPI target: use eContract to include VNF licensed by at least 3 different providers).*	See Sec. 4.4 for e-Licensing Management and Sec. 5.8 for Network Slice and Service Orchestration Information Model
OBJ-7. Validate the 5GZORRO zero-touch automation, security and trust in relevant use cases for the implementation of Smart Contracts for Ubiquitous Computing/Connectivity, Dynamic Spectrum Allocation, and Pervasive virtual CDN services over 3rd-party edge resources.	*No specific target to be covered by architecture design*	n/a
OBJ-8. Ensure the long-term success of the project through standardization and dissemination in scientific, industrial, and commercial fora, and by contributing to relevant open source communities & SDOs also exploring synergies with other EU initiatives and projects.	*No specific target to be covered by architecture design*	5GZORRO architecture include and is aligned with many SDO design and specifications in all its elements as reported in Section 2, 3, 4, and 5.

7 References

[1] ETSI ZSM ISG information. URL: https://portal.etsi.org/zsm Accessed 27 April 2022.

[2] Stafford, V.A.: Zero trust architecture. NIST Special Publication. 800, 207 (2020)

[3] Kaloxylos, Alexandros, Gavras, Anastasius, Camps Mur, Daniel, Ghoraishi, Mir, & Hrasnica, Halid. (2020). 5G PPP Whitepaper, AI and ML – Enablers for Beyond 5G Networks. Zenodo. https://doi.org/10.5281/zenodo.4299895

[4] Azure Cloud – SGX powered Servers. URL: https://azure.microsoft.com/en-us/blog/dcsv2series-vm-now-generally-available-from-azure-confidential-computing/ Accessed 27 April 2022.

[5] Intel SGX Powered CPU. URL: https://ark.intel.com/content/www/br/pt/ark/products/193743/intel-xeon-e-2288g-processor-16m-cache-3-70-ghz.html. Accessed 27 April 2022.

[6] Secure Enclaves for Reactive Cloud Applications (SERACA) project. URL: https://www.serecaproject.eu/. Accessed 27 April 2022.

[7] Software Guard eXtensions (SGX). https://kernel.org/doc/html/x86/sgx.hmlt#. Accessed 28 April 2022.

[8] SecureCloud. URL: https://www.securecloudproject.eu/. Accessed 27 April 2022.

[9] SCONE. URL: https://sconedocs.github.io/aboutScone/. Accessed 27 April 2022.

[10] Design-IT: How to TAP traffic in a virtual environment. URL: https://www.garlandtechnology.com/blog/design-it-how-to-tap-traffic-in-a-virtual-environment. Accessed 25 April 2022.

[11] Wagner, C., Dulaunoy, A., Wagener, G., Iklody, A.: MISP: The design and implementation of a collaborative threat intelligence sharing platform. In ACM on Workshop on Information Sharing and Collaborative Security. 49-56 (2016)

[12] Open Source Mano ETSI. https://osm.etsi.org/

[13] Common Event Format, ArcSight, Inc. URL: https://kc.mcafee.com/resources/sites/MCAFEE/content/live/CORP_KNOWLEDGEBASE/78000/KB78712/en_US/CEF_White_Paper_20100722.pdf. Accessed 27 April 2022.

[14] Ghafir, I., Prenosil, V., Svoboda, J., Hammoudeh, M.: A survey on network security monitoring systems. In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops. 77-82 (2016)

[15] Shah, N., Willick, D., Mago, V.: A framework for social media data analytics using Elasticsearch and Kibana. Wireless Networks. 1-9 (2018)

[16] Argo Project. https://argoproj.github.io. Accessed 27 April 2022.

[17] Kubernetes. https://kubernetes.io/docs/concepts/extend-kubernetes/operator/. Accessed 27 April 2022.

[18] 5G-MEDIA Serverless orchestration, infrastructure and VIM Driver implementation for OSM. https://github.com/5g-media/faas-vim-plugin. Accessed 27 April 2022.

[19] Etcd. https://etcd.io/. Accessed 27 April 2022.

[20] MySQL. https://www.mysql.com/products/community/. Accessed 27 April 2022.

[21] OptaPlanner. https://www.optaplanner.org/. Accessed 27 April 2022.

[22] Gurobi Optimization. http://www.gurobi.com/. Accessed 27 April 2022.

[23] Red Hat Operator Framework. https://www.redhat.com/en/blog/introducing-operator-framework-building-apps-kubernetes. Accessed 25 April 2022.

[24] Red Hat Open Cluster Management Kubernetes. https://github.com/open-cluster-management. Accessed 27 April 2022.

[25] Red Hat Advanced Cluster Management. https://www.redhat.com/en/technologies/management/advanced-cluster-management. Accessed 27 April 2022.

[26] IBM ILOG CPLEX Optimization. http://www.cplex.com/. Accessed 28 April 2022.

[27] PuLP. https://www.pulpproject.org/. Accessed 28 April 2022.

[28] GNU Linear Programming Kit. http://www.gnu.org/software/glpk/glpk.html. Accessed 28 April 2022.

[29] Coin-or branch cut. https://github.com/coin-or/Cbc. Accessed 28 April 2022.

[30] Mosek. https://www.mosek.com/. Accessed 28 April 2022.

[31] FICO Xpress Solver. https://www.fico.com/es/products/fico-xpress-solver. Accessed 28 April 2022.

[32] Choco-solver. https://choco-solver.org/. Accessed 28 April 2022.

[33] Mixed Integer Programming Class Library CPP. http://mipcl-cpp.appspot.com/. Accessed 28 April 2022.

[34] Scpipt. https://www.scipopt.org/. Accessed 28 April 2022.

[35] Free5GC. https://www.free5gc.org/. Accessed 28 April 2022.

[36] 5GCity, Final 5GCity Orchestrator Release (D4.4), October 2019.

[37] Zeek network security monitor types. URL: https://docs.zeek.org/en/current/script-reference/types.html. Accessed 27 April 2022.

[38] ETSI GS NFV-SOL 006 V3.3.1 (2020-08): Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; NFV descriptors based on YANG Specification

[39] ETSI GS NFV-IFA 011 V4.1.1 (2020-11): "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; VNF Descriptor and Packaging Specification"

[40] ETSI GS NFV-IFA 014 V3.3.1 (2019-09): "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Network Service Templates Specification"

[41] ETSI GS MEC 010-2 V2.1.1 (2019-11): Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management

[42] GSMA NG.116 - Generic Network Slice Template, V 3.0, 22 May 2020. URL: https://www.gsma.com/newsroom/wp-content/uploads/NG.116-v3.0.pdf. Accessed 25 April 2022.

[43] 3GPP SA5, “Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3,” 3GPP TR 28.541, version 17.6.0, March 2022

[44] 5GZORRO Consortium, Deliverable D2.2 – “Design of the 5GZORRO Platform for Security & Trust”, November 2020

[45] TM Forum Open-API Schema Repository, "GeographicAddress". URL: https://github.com/tmforum-rand/schemas/blob/candidates/Common/GeographicAddress.schema.json. Accessed 25 April 2022.

[46] ETSI GS MEC 010-2 V1.1.1, "Mobile Edge Computing (MEC); Mobile Edge Management; Part 2: Application lifecycle, rules and requirements management" (2017-07)

[47] Resource Catalog Management API REST Specification" (2020-11), TM Forum Specification, TMF634, Release 17.0.1, December 2017.

[48] Product Catalog Management API REST Specification, TM Forum Specification, TMF620, Release 19.0.0, July 2019.

[49] D4.1: Design of Zero Touch Service Management with Security & Trust Solutions. https://www.5gzorro.eu/wp-content/uploads/2021/10/5GZORRO-D4.1-EC-approved.pdf. Accessed 27 April 2022.

[50] D4.2: Intermediate prototype of Zero Touch Service Mgmt with Security and Trust. https://www.5gzorro.eu/wp-content/uploads/2021/08/D4.2_v1.0-FINAL-QA-TM-SS.pdf. Accessed 27 April 2022.

[51] 3GPP TS 28.541, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16)", v16.2.0, September 2019

[52] SliceNet, Cross-Plane Slice and Service Orchestrator (D7.1), May 2020. URL: https://bscw.5g-ppp.eu/pub/bscw.cgi/d361865-3/*/*/*/DOI-SLICENET-D7.1.html. Accessed 27 April 2022.

[53] 5G-Transformer, Deliverable D3.1 – "Definition of vertical service descriptors and SO NBI", Marc 2018

[54] ETSI GS NFV-IFA 031 V3.3.1: Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Requirements and interfaces specification for management of NFV-MANO

[55] Nextworks Slicer OpenApi specification v2-0. URL: https://github.com/nextworks-it/slicer/blob/master/API/sebastian-openapi-v2-0.yaml. Accessed 27 April 2022.

[56] Generic Network Slice Template, Version 3.0 22 May 2020, https://www.gsma.com/newsroom/wp-content/uploads/NG.116-v3.0-1.pdf. Accessed 27 January 2021.

[57] Standardized VNF License Management Framework White Paper https://wiki.mef.net/display/CESG/Standardized+VNF+License+Management+Framework+-+White+Paper. Accessed 27 April 2022.

[58] D. Breitgand, Alexios Lekidis, Rasoul Behravesh, Avi Weit, Pietro Giardina, Vasileios Theodorou, Cristina E. Costa, Katherine Barabash, "Dynamic Slice Scaling Mechanisms for 5G Multi-domain Environments," 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), 2021, pp. 56-62, doi: 10.1109/NetSoft51509.2021.9492716.

8 Abbreviations and Definitions

8.1 Definitions

No definition introduced in this deliverable.

8.2 Abbreviations

5G IA	5G Infrastructure Association
AIOps	Artificial Intelligence for IT operations
CNF	Cloud Native Function
DID	Distributed Identifier
DIF	Decentralised Identity Foundation
DLT	Distributed Ledger Technology
DPKI	Decentralised Public Key Infrastructure
EC	European Commission
FaaS	Function as a Service
ISSM	Intelligent Slice and Service Manager
K8s	Kubernetes
LCM	LifeCycle Management
MANO	Management and Orchestration
MEC	Mobile Edge Computing
NBI	Northbound Interface
NFV	Networks Function Virtualization
NFVI	Networks Function Virtualization Infrastructure
NFVO	Networks Function Virtualization Orchestrator
NPM	Node Package Manager
NS	Network Service or Network Slice depending on the context
NSM	Network Service Mesh
POP	Product-Offering Price
RAN	Radio Access Network
SC	Smart Contract
SDO	Standards Development Organization
SM	Service Mesh
VC	Verifiable Claim
VDU	Virtual Deployment Unit
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
W3C	World Wide Web Consortium
WG	Working group
WP	Work Package
ZSM	Zero Touch Service Management

<END OF DOCUMENT>