5GZORRO

Grant Agreement 871533

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

D4.1: Design of Zero Touch Service Management with Security & 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: **V1.0**

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

Deliverable title:
D4.1: Design of Zero Touch Service Management with Security & Trust Solutions

Start date of the project: **01/11/2019 (duration 30 months)**	Contractual delivery date: **31.10.2020**	Actual delivery date: **31.01.2021**

Editor(s)
Gregorio Martínez Pérez, José María Jorquera Valero, Pedro Miguel Sánchez Sánchez (UMU)

List of Contributors

Participant	Short Name	Contributor
Universidad de Murcia	UMU	J. M. Jorquera Valero, P. M. Sánchez Sánchez, M. Gil Pérez, G. Martínez Pérez
Ubiwhere	UW	P. Diogo, P. Martins
Atos Spain	ATOS	F. Bravo Díaz, A. Ramos
Altice Labs	ALB	J. Bonnet, M. Mesquita
Fundaciò i2CAT	I2CAT	J. Fernández, C. Herranz Claveras, A. Fernández-Fernández, M. S. Siddiqui, L. A. Ochoa
Telefónica Investigación y Desarrollo	TID	C. Rodríguez Cerro, D. R. López
IBM Israel Science and Technology	IBM	D. Breitgand, K. Barabash
Nextworks	NXW	P. G. Giardina, J. Brenes, E. Bucchianeri, G. Carrozzo
Intracom	ICOM	A. Lekidis, V. Theodorou
Fondazione Bruno Kessler	FBK	R. Behravesh

List of Reviewers

Participant	Short Name	Contributor
Fondazione Bruno Kessler	FBK	R. Behravesh
Nextworks	NXW	J. Brenes, P. G. Giardina, G. Carrozzo
Fundaciò i2CAT	I2CAT	M. S. Siddiqui

Change History

Version	Date	Partners	Description/Comments
0.0	25-09-2020	UMU	Table of Contents and editing assignments
0.1	01-12-2020	UMU, i2CAT, NXW, ICOM, ATOS,	Background sections; initial technical content
0.2	08-12-2020	TID, ALB, i2CAT, NXW, UMU, ICOM, FBK	Revise/edit starting technical contents by other partners
0.3	15-12-2020	ALB, NXW, ICOM, UMU, i2CAT	Initial contributions of software module component specification (interfaces and information models)

0.4	04-01-2021	UMU, NXW, i2CAT, UW, ALB, IBM	Second contributions of software module component specification (interfaces and information models). IBM provides initial contribution of ISSM design
0.5	12-01-2021	NXW, i2CAT, UMU	Refinement of the current content to improve the argument and close the forgotten details
0.6	14-01-2021	UW, i2CAT, ALB, ICOM, UMU, NXW, ATOS, TID	Refinements and formatting of all sections
0.7	15-01-2021	NXW, ICOM, UMU	Response to open comments and refinement of sections
0.8	20-01-2021	IBM, UW, TID	Jan 18 IBM contribution Section 3 second contribution. UW contribution Section 1.1.1 first contribution. TID provided a general revision
0.9	25-01-2020	NXW, FBK	QA review
0.10	27-01-2020	ATOS, UMU, i2CAT, NXW, ICOM, UW	Refinement round after QA reviews
0.11 - QA	29-01-2021	UMU, NXW	Final QA after TM review
1.0	31-01-2021	I2CAT	Final submission version
1.1	22-09-2021	UMU	Updated the trust management model architecture and added text on ML/DL algorithms and PeerTrust model

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 design of the 5GZORRO platform modules dedicated to automated service management with security and trust, by detailing the high-level design concepts presented in deliverable D2.2 *(Design of the 5GZORRO Platform for Security & Trust*)

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), slicing, security, trust, etc. which are also included in the descriptions and architectures presented as part of this design document.

This deliverable contains 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 design details related to the KPIs, and the module APIs/Interfaces. Moreover, the information models are presented, one for each module.

Changes from v1.0 (Jan-21) to v1.1 (Sept-21)

A revision of this design document is issued with version 1.1 which contains additional details on the proposed trust management framework both in terms of architecture which is updated and of techniques used to compute statistically trust levels.

This design incorporates references to specific 5GZORRO ideas presented in the 5G PPP Whitepaper, AI and ML – Enablers for Beyond 5G Networks (Zenodo. https://doi.org/10.5281/zenodo.4299895), specifically in the context of AI/ML algorithms for the Trust Management Framework and their related security aspects.

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 infrastructure, 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, slices, and any other necessary element.

This document details the architecture of a trustworthy and intelligent network slice management mechanism for building secure cross-domain slices and services. Taking Figure 1-1 as 5GZORRO reference architecture, this report encloses security and trust across multiple domains and cross-domain intelligent network slice & service management.

1.1 Document scope and objectives

Aligned with the previous ideas, this document seeks to provide the design details for the key contributions of 5GZORRO framework regarding automated service management including security and trust. The

document also covers the design of the software modules required to fulfil the desired objectives for service management automation with security and trust.

1.1.1 Deliverable Scope with respect to the 5GZORRO Architecture specification

Having in mind the 5GZORRO high-level reference architecture depicted below and documented in greater detail in D2.2, this deliverable focuses on the components represented in the bottom half of the diagram, namely the **Security and Trust Functions** and the **Zero-Touch Management and Orchestration Functions.**

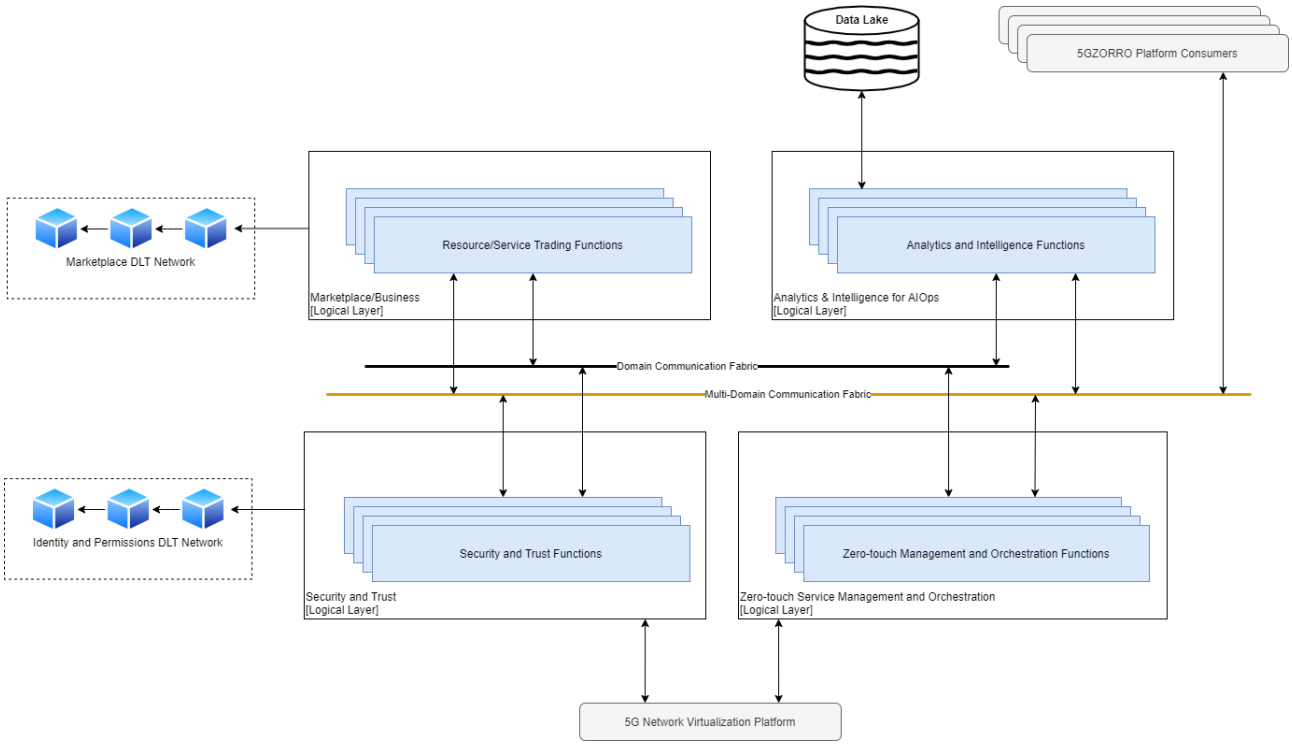

Figure 1-1 : 5GZORRO High Level reference architecture

As mentioned in section 1.1, these particular components are 5GZORRO key building blocks when it comes to establishing a zero-trust and secure Telco marketplace for multilateral business agreements across different stakeholders who securely connect with one another across multiple domains. Thus, this deliverable takes into account the initial high-level architecture considerations and assumptions, and focuses on the actual implementation, detailing how the different components interface with one another (secure intra-domain and cross-domain communication fabric) to enable a secure and trustworthy marketplace with zero-trust service management and orchestration.

The deliverable is mainly focused on the design of the 5GZORRO core platform components which can implement the three capabilities mentioned above, in order to derive an intelligent zero-touch solution to orchestrate the trustworthy establishment of secure communications in distributed multi-party 5G scenarios.

1.1.2 Related 5GZORRO Project Objectives

The main objectives of this deliverable can be summarised in the following:

- Research, design and an initial implementation of zero-touch service in order to accomplish security & trust requirements for distributed multi-party 5G scenarios through innovative mechanisms.

- Design and an initial implementation of mechanisms to harmonise cross-domain security and trust relationships, including trust establishment and secure communications among different stakeholders.
- Design and an initial implementation of mechanisms to promote trusted execution of offloaded workloads across multiple domains and dynamic vulnerability assessment, using internal resource and service metrics and ML/DL techniques to detect the attacks and mitigate them.
- Investigation and an initial implementation of enhancements of network slicing and NFV-MANO to provide end to end intelligent and secured data-driven automated management solutions for services and network slices.

1.2 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.
- Section 3 describes the 5GZORRO domain intelligent services by means of an automated slice management approach that utilises security and trust service to build cross-domain slice and services in distributed multi-party 5G scenarios.
- Section 4 covers the development of NFV-MANO and network slicing enhancements to implement the 5GZORRO extensions.
- 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.
- 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 component is divided into various modules according to their scope (see Figure 2-1). Concretely, these modules are:

- ***Trust Management Framework***, 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.
- ***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.
- ***Inter-domain security establishment at the communication level***, which manages the establishment of secure and trusted connections between different domains in the 5GZORRO environment, guaranteeing privacy and integrity properties without sacrificing performance.

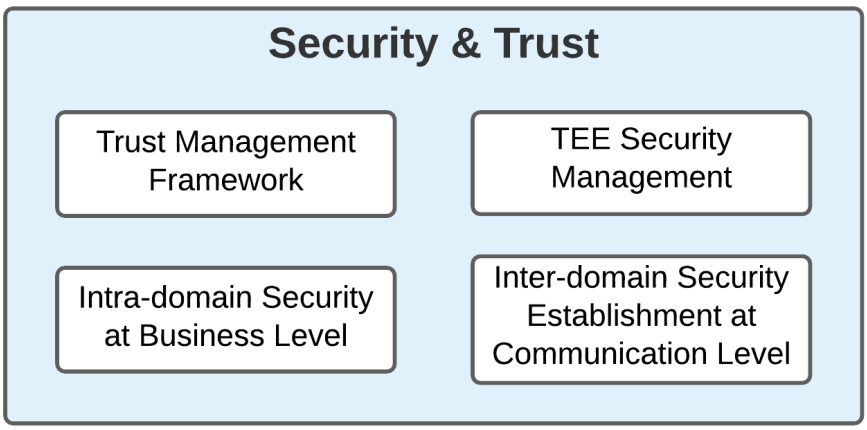

Figure 2-1: Security and Trust Orchestration modules

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 in its functioning. Next section depicts in depth the enhancements and design details of the Security and Trust Orchestration modules designed and developed in 5GZORRO Project.

2.1 Trust Management Framework

This module covers the design, implementation, and validation of the trust 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.

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 trust manager include: deciding the information sources to be used by the algorithms when gathering trust information, trust assessment, and trust assessment report. The constant update of trust values, and improvements for the current models and the mechanisms to assess trust, are also in the scope of the trust management framework. 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 we commented before, this module has a decentralized nature, unlike most of the previous 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 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, this 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, this 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 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 management framework. Furthermore, Figure 2-2 also depicts the main modules and interfaces that trust management framework should contemplate 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.

The first phase is about the gathering of *trust statements* 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 management framework should start by figuring out what are the available sources. Considering the 5GZORRO platform, the support for the statements will be provided by Distributed Ledger Technologies (DLTs), with the outcomes of Smart Contracts (SCs) based on Service Level Agreements

(SLAs) considered as the generators of the statements conveying trust information. There are three main sources for trust statement recollection, depicted at the left side in Figure 2-2, as input to the Trust Management module, from which to derive and infer features: the shared Datalake platform, the Monitoring analytics, and the Security management service. Note that, the trust 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 trust management framework. For instance, a stakeholder is able to enable or disable a continuous data collection for a specific third-party. Once information sources have been determined, the trust management framework will generate from this information a set of trust scores.

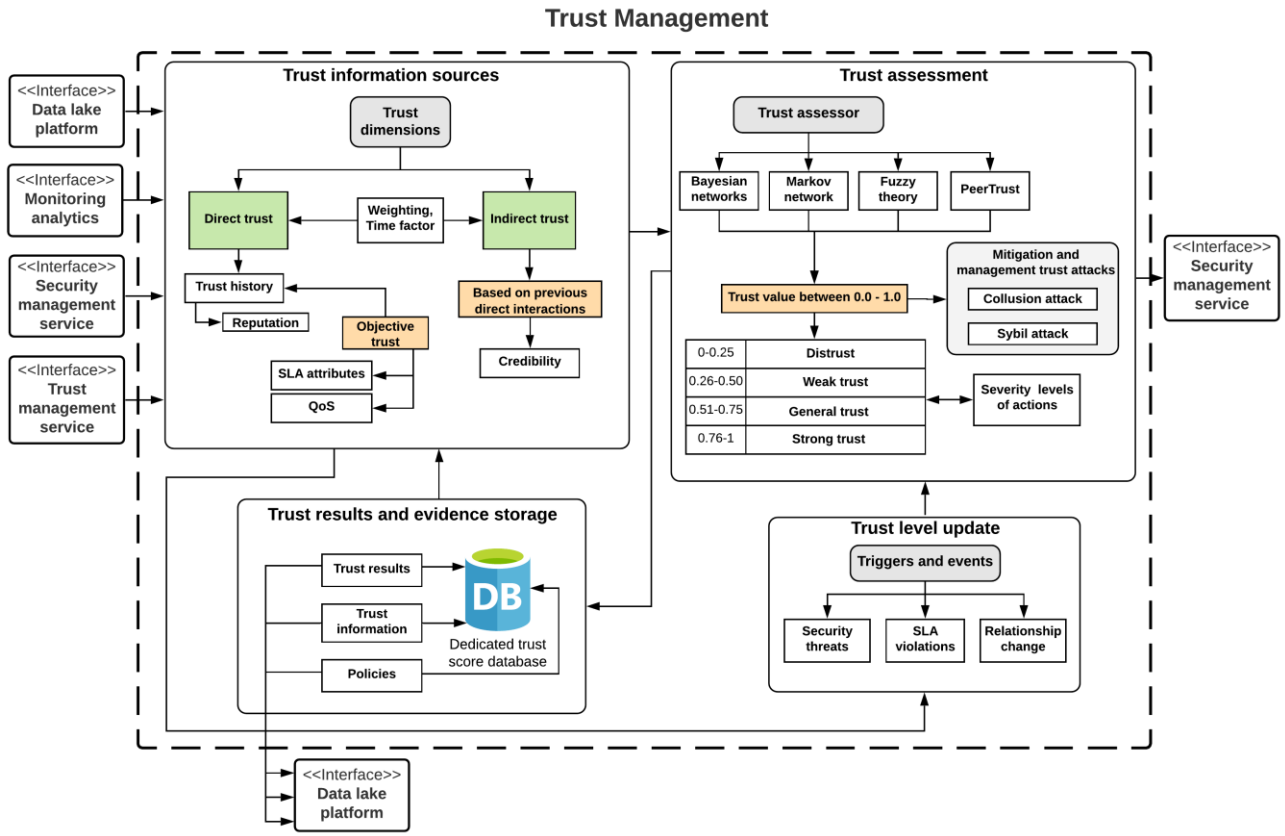

Figure 2-2: Trust management module architecture

As can be observed in Figure 2-2, the 5GZORRO trust management framework 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.

After getting statements based on direct and indirect trust, the next step is to evaluate a trust score for the entity. In this regard, the *trust assessment* module plays an essential role. 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 by us 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. The Figure 2-2 depicts some of the most well-known and accurate intelligent techniques used in the literature such as Bayesian networks, Markov networks, and Fuzzy theory. Moreover, Figure 2-2 also 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 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 takes into account 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 assessment* module should withstand common trust attacks such as collusion attack (unrealistic or dishonest recommendations) and Sybil attack (multiple identities, associated with the same entity, increasing/decreasing reputation).

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 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 trust framework 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*. Due to the fact that 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 modules. 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 trust management framework may utilise in order 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.

Last but not least, the trust management framework contains the *trust level update module.* 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 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.

To support this, the trust management framework service will provide the necessary APIs to enact all trust actions. Table 2-1 depicts a general view of the main capabilities supported by the trust management service, as well as its level of support (i.e., M-mandatory, O-Optional). Table 2-2 provides more detailed information on the above capabilities introduced in Table 2-1.

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

Service name**: Trust management framework**		Type**: *Per-domain & Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Start Data Collection	**M**	This capability enables/disables continuous data collection about a stakeholder.
Gather Information	**M**	This capability extracts available information from an interested party from data lake platform.
Compute Trust Level	**M**	This capability calculates trust level of the stakeholder to some internal resource using the previously acquired parameters.
Store Trust Level	**M**	This capability enables to record the previously calculated trust level and the utilized information in the available storage sources.
Query Trust Level	**M**	This capability enables to request the current trust level of a particular resource. If there is no value calculated, it triggers the assessment process.
Notes		

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

Operation name: **startDataCollection**		
Description	This method is responsible for starting or stopping a continuous collection data (e.g., Monitoring Analytic, Security Management Service, etc.) from Data Lake platform.	
Input Parameters	**Type**	**Description**
stakeholder_did	String	Stakeholder's DID on which it launches the process of continuous information recollection.
trust_parameters	Dictionary	Empty template with the data to be collected in the Data Lake
*specific_events**	List of objects	Set of triggers that enable an event driven pre-processing to take place directly in the Data Lake.
Output Parameters	**Type**	**Description**
data_collection_status	Boolean	It returns true or false if the data collection cycle is active or not.
Notes:		
More detailed information about the dictionary can be found in section 5.1 (trust management framework information model). *Specific_events** is an optional parameter.		

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.
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 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: **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**
stakeholder_did	String	Stakeholder's DID from whom to retrieve trust info.
Output Parameters	**Type**	**Description**
result	Double	Trust level previously computed.
Notes		

2.2 Trusted Execution Environment Security Management

The multi-tenant and multi-stakeholder nature of 5GZORRO creates a scenario where no inherent trust mechanisms exist. However, while a stakeholder or service provider can protect its machines against a malicious tenant using mainstreamed solutions such as virtualisation and containerization, fewer solutions exist to protect a tenant service or application running in a computing node against a malicious stakeholder with root access.

This module focuses on the development of such functionalities by integrating 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.

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. In particular, WP2 along with WP4 ought to tackle quantified targets according to TEEs, which are partially associated with this deliverable, as well as D4.2 and D4.3.

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, 5GZORRO carries out a research in order to discover commercial TEE platforms that covers requirements associated with its ecosystem. During initial phase of design, 5GZORRO has studied various TEE platforms, identifying those which can provide a clear added value to the project. As stated in the DoA, three different TEE platforms would have to be identified – 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 SW API exposing, at a higher level, the capabilities offered by existing TEE platforms such as Intel SGX interfaces (x86) and that of GlobalPlatform's TEE Client API (which relies on OP-TEE's libraries, focused on exposing ARM's TrustZone capabilities). This abstract API intends to merge the functionalities provided by lower-level libraries, so that we have a common and single API for a subset of architectures.
2. X86 hardware TEE Platform with SGX capabilities – this platform will be provided by an external Cloud Provider, Azure [4], allowing 5GZORRO to access a pool of compute resources whose underlying CPUs have built-in native SGX support. Ubiwhere, 5GZORRO partner, will facilitate access to such environment. Specifically, it is a 3.7GHz Intel XEON E-2288G with Software Guard Extensions (Intel SGX) and Trusted Execution Technology support [5].
3. Building on top of the aforementioned x86 hardware TEE Platform, Ubiwhere will deploy 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 [8]). More details on SCONE's functionalities that will be integrated in 5GZORRO as a way to interface with other components 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 these with the current MANO tools for 5G networks.

In this sense, 5GZORRO develops as a novelty a "TEE-as-a-Service" platform, which enables the integration of TEE solutions with different elements of the telco infrastructure environment. The main objective of this solution is to provide high execution security and trustworthiness into the system deployment lifecycle. Besides, this platform can be leased through the 5GZORRO Marketplace to third party tenants, together with processing offloading capabilities as an additional security guarantee.

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

- Containerized VIM resources and services. Here, the objective is to achieve VIM trusted execution by deploying these as containerized systems isolated from the host system using TEE solutions. This isolation is designed to protect sensitive data and processes running critical services. Then, the entire component lifecycle has to be covered by the TEE platform, including the services for secure boot process, secure connection and data management, and kernel memory and code integrity check to avoid code injection and overflow attacks.
- Secure SLA Breach Prediction (as a Secure Oracle). Since SCONE also includes attestation mechanisms, it can ensure that sensitive operations, such as the SLA 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. Future work will be on ensuring that the 5GZORRO module responsible for this task (more can be found in Deliverable 3.1) will be compiled and packaged as a secure container, using SCONE. 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 directly 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) will expose the methods allowing for such a thing, which will be something 5GZORRO will focus on.

Besides, the "TEE-as-a-Service" platform also offers capabilities to enable the execution of other simpler components and code, such as tasks manipulating critical data, improving the flexibility of the objects that can be deployed leveraging the TEE properties.

In the scope of 5GZORRO, SCONE modules can be integrated in the orchestration services allowing the deployment of critical services on SGX-enabled nodes present in the marketplace. The orchestrating services can then deploy a custom application in a secure enclave, ensuring end-to-end encryption and secure provisioning of the application, its data, and keys.

The following tables introduce the necessary operations to be supported by the TEE Security Management service.

Table 2-3: Definition of Trusted Execution Environment Security Management service (domain level)

Service name**: TEE Capabilities Management**		Type**: *Per-domain***
Capabilities	**Support (O|M)**	**Description**
Get TEEs	**M**	This capability enables to acquire the current TEEs available in which code and service execution can be carried out.
Create TEE connection	**M**	This capability initializes and configures the connection with the available or selected TEE.
Execute command in TEE	**M**	This capability allows to execute commands in the TEE once a connection has been established.
Delete TEE connection	**M**	This capability deletes the connection with the TEE, deleting all the data left in the environment.

Notes
None.

Table 2-4: Definition of Trusted Execution Environment Security Management service interfaces

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 Intra-domain Security at the business level

The intra-domain security is the module that aims to design, develop, and validate the security services in charge of detecting possible vulnerabilities and attacks, and apply the required countermeasures in order to mitigate these adverse events. The scope of this module covers an intra-domain perspective where each stakeholder deploys the services enabled by this module in order 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.

In particular, this module leverages the usage of internal resource and service metrics (such as network communications, resource usage, etc.) and ML techniques to 1) detect attacks and 2) provide mitigation procedures for them before they affect significantly the stakeholder infrastructure. Initially, for the detection part, this module identifies effectively operational and cyber-security threats based on evidence that originates from network communications. Additionally, operational threats related to failures or malfunctions, as well as cyber-security threats related to malicious activity or abuse of the 5GZORRO platform components. Secondly, the mitigation procedures are based on the integration of tools that ensure the protection against unauthorized and malicious entities (e.g., through the configuration of firewall policies) as well as the isolation of infrastructure or 5GZORRO platform components that are compromised (e.g., moving them to a specific VLAN). As a result, mitigation should ensure the continuous operation and reliability of the 5G infrastructure as well as the 5GZORRO platform components.

Furthermore, this module also includes a risk management methodology development inspired by ISO/ICE 27005 standard, covering those of its steps that are considered a universal approach to risk management – context establishment, risk identification, risk analysis, and risk evaluation (assessment).

The intra-domain security module monitors 1) the 5GZORRO platform modules, resources and services of the domain it is deployed on and 2) the network infrastructure of the 5GZORRO platform consumers i.e., the 5G core network or mobile edge infrastructures. To enable 2), access to the network packets exchanged at the different segments involved in the network slice and their corresponding planes (control, user, management). The high-level architectural diagram of this module is shown in Figure 2-3.

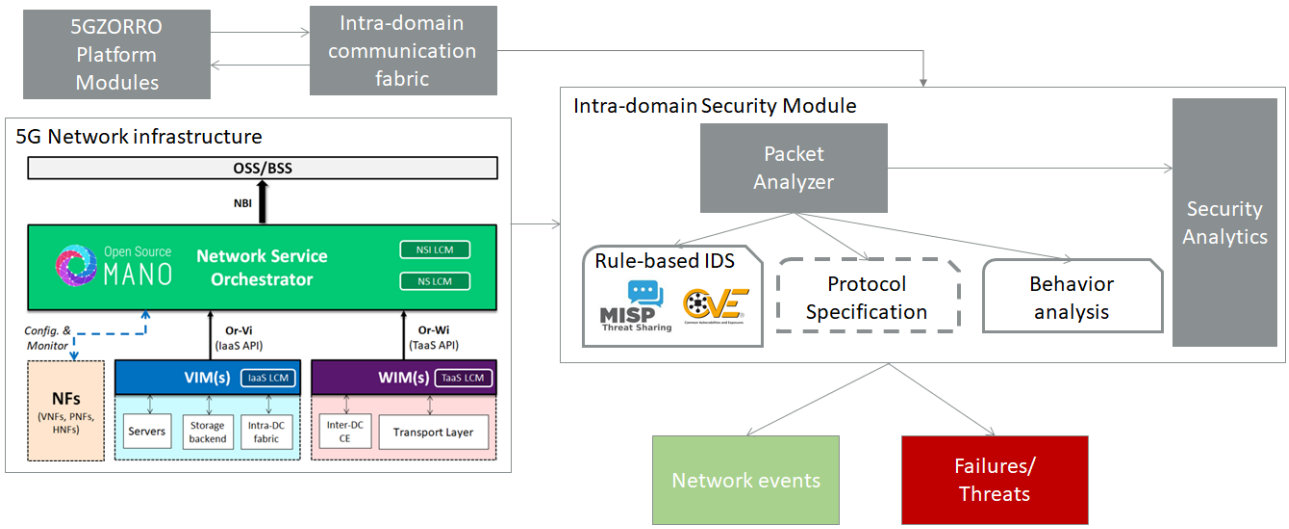

Figure 2-3: Intra-domain security module architecture

For the network monitoring part of the intra-domain security module, the **Packet Analyzer** is using a virtual SPAN (i.e., port mirroring) or TAP configuration or network probes on network elements as switches, gateways and routers to collect network traffic data and control commands for each domain. Specifically, it is monitoring a switch, router or a gateway to obtain the 5G Network infrastructure traffic and has a virtual TAP configuration [9] to capture a copy of the data flowing between the 5GZORRO Platform modules of a certain domain. This happens for example through the presence of a virtual switch on the virtual links between the domain modules. Since traffic data and control commands can be sent either in standardized or proprietary formats, it also includes dissector modules for interpreting them. These network protocols are the ones used in 1) the 5G Network infrastructure (e.g., Core, Edge and Radio Access Network) as well as 2) the Intra-domain communication fabric.

The detection part is focused on both known and unknown (zero-day) threats, by employing three main threat detection mechanisms:

- **Rule-based Intrusion Detection System (IDS)**: follows a set of predefined rules and signatures for detecting known threats. The required inputs for this mechanism are obtained by Threat Intelligence platforms, as the Malware Information Sharing Platform (MISP) [9] or vulnerability databases as MITRE CVE [11].
- **Protocol specification**: follows the specifications of the network protocols that are used to communicate between the intra-domain modules and detects anomalies when unusual message sequences, unsupported commands/codes, as well as error or unsupported messages are spotted. Since many protocols lack a dedicated specification or their specification is only proprietary, this module may require reverse engineering actions for interpreting commands that are inside the exchanged packets.
- **Behaviour analysis**: uses ML techniques to learn the intra-domain behaviour by analysing network data and forming a baseline upon which the detection of anomalies is feasible.

The **Security Analytics** module uses the collected data from the Packet Analyzer, in order to perform data analytics, provide operational information (visualization, alarms, etc.), and produce Indicators of Compromise (IoCs) useful for the investigation of failures or threats.

Overall, the Intra-domain Security module receives network traffic data as input and produces network events and alerts about potential failures or security threats using the three detection mechanisms. The events and alerts are in the form of the Common Event Format (CEF) [12].

The current version of the Intra-domain Security module extends the functionality and protocol support of the Zeek network security monitor [12], integrated with a 5GZORRO-tailored Kibana [14] dashboard for the Security Analytics module. The choice of Zeek was based on its ability to perform knowledge- and behaviour-based intrusion detection, instead of the rule-based detection strategies that Suricata and Snort apply [12]. The complete integration is illustrated in Figure 2-4.

Figure 2-4: 5GZORRO Intra-domain security module implementation

Related to the Key Performance Indicators that have been identified in Deliverable 2.1 and Deliverable 2.2, the intra-domain security module has to verify a target value regarding common attacks and countermeasures. Concretely, this module should validate its performance by means of identifying 6 different types of common attack types to software infrastructures and provide a complete set of countermeasures for them. This target can be achieved through the functionality supplied by the Packet Analyzer and the Security Analytics modules which 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.

In this vein, the fulfilment of this KPI not only provides the 5GZORRO platform with a continuous monitoring service that enables us to secure our platform but also guarantees a set of basic security services on an external domain or platform when we are in a multi-domain slicing context.

More details about the intra-domain security service capabilities at domain level are outlined in Table 2-5 and Table 2-6, respectively.

Table 2-5: Definition of Intra-domain Security service (per-domain level)

Service name: **Intra-domain security**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Start network infrastructure monitoring	**M**	This capability enables monitoring of the network traffic for different 5G infrastructure segments (i.e., core, RAN, MEC) through a virtual SPAN port configuration.
Stop network infrastructure monitoring	**M**	This capability disables monitoring of the network traffic for different 5G infrastructure segments (i.e., core, RAN, MEC).
Start platform monitoring	**M**	This capability enables monitoring of the network traffic that is exchanged by the 5GZORRO modules through the Communication fabric.
Stop platform monitoring	**M**	This capability disables monitoring of the network traffic that is exchanged by the 5GZORRO modules through the Communication fabric.
Get monitored interfaces	**M**	This capability allows to obtain the list of active monitoring interfaces.
Get monitoring data	**M**	This capability allows to obtain diagnostics, logs and Packet Capture (PCAP) files from monitored interfaces.

Set monitoring submodules	**M**	This capability sets the submodules of the Intra-domain security service that are used for detection of anomalies.
Notes		

Table 2-6: Definition of Intra-domain Security service interfaces

Operation name: **startNetworkInfrastructureMonitoring**		
Description	This method starts the network monitoring of the 5G network infrastructure	
Input Parameters	**Type**	**Description**
interfaceName	String	The name of the network interface that will be monitored (default is all available 5G network infrastructure interfaces)
config	String	Sets if monitoring is real-time or through stored network data (e.g., logs, PCAP files)
Output Parameters	**Type**	**Description**
status	String	The result (i.e., OK or failure) of starting the Intra-domain security service
eventID	String	The identifier of a network diagnostic event that is generated from the Intra-domain security service
alertID	Integer	The identifier of an alert network diagnostic event that is generated from the Intra-domain security service
alertType	String	The type of alert (i.e., failure, threat, warning, information) that is generated from the Intra-domain security service
Notes		

Operation name: **stopNetworkInfrastructureMonitoring**		
Description	This method stops the network monitoring of the 5G network infrastructure	
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 Intra-domain security service
Notes		

Operation name: **startPlatformMonitoring**		
Description	This method starts the monitoring of network traffic from 5GZORRO platform modules through the Communication fabric	
Input Parameters	**Type**	**Description**
interfaceName	String	The name of the interface that will be monitored (default is all available platform interfaces)
config	String	Sets if monitoring is real-time or through stored network data (e.g., logs, PCAP files)
Output Parameters	**Type**	**Description**
status	String	The result (i.e., OK or error) of starting the Intra-domain security service
eventID	String	The identifier of a network diagnostic event that is generated from the Intra-domain security service

	alertID	Integer	The identifier of an alert network diagnostic event that is generated from the Intra-domain security service
	alertType	String	The type of alert (i.e., failure, threat, warning, informational) that is generated from the Intra-domain security service
Notes			

Operation name: **stopPlatformMonitoring**		
Description	This method stops the monitoring of network traffic from 5GZORRO platform modules through the Communication fabric	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
status	String	The result (i.e., OK or failure) of stopping the Intra-domain security service
Notes		

Operation name: **getMonitoredInterfaces**		
Description	This method retrieves the list of all the active monitored interfaces by the Intra-domain security service	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
IfList	String array	The returned list of monitored interfaces
Notes		

Operation name: **getMonitoringData**		
Description	This method retrieves the diagnostic logs and Packet Captures (PCAPs) that are captured by the Intra-domain security module	
Input Parameters	**Type**	**Description**
startTime	String	The start time of log collection
endTime	String	The end time of log collection
Output Parameters	**Type**	**Description**
logName	String	The collected log files for the timeframe [startTime, endTime]
pcapName	String	The collected PCAP files for the timeframe [startTime, endTime]
Notes		

Operation name: **setMonitoringSubmodules**		
Description	This method selects the submodules of the Intra-domain security service that will be enabled for the detection of anomalies	
Input Parameters	**Type**	**Description**
moduleID	String	The submodule identifiers of the Intra-domain security service (i.e., 1-> Rule-based, 2-> Protocol-based, 3-> Behaviour-based)

	endTime	String	The end time of log collection
Output Parameters		**Type**	**Description**
	status	String	The updated list of selected modules used for the detection
Notes			

2.4 Inter-domain Security at the communication level

This module aims to design, implement, and validate the software components required 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 present in 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.

Figure 2-5 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, Operator A detects that it is not able to cover the performance indicated in its SLA, and therefore, it selects a resource and signs a Smart Contract with Operator B. After that, both operators request the public key of each other to start the process for establishing a secure and private communication channel. Since 5GZORRO leverages DIDs and 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, Operator A starts the VPN configuration process, and in particular, the initial process called shared key generation. By means of Operator B's public key acquired from its Verifiable Credential (IdM DLT network), 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.

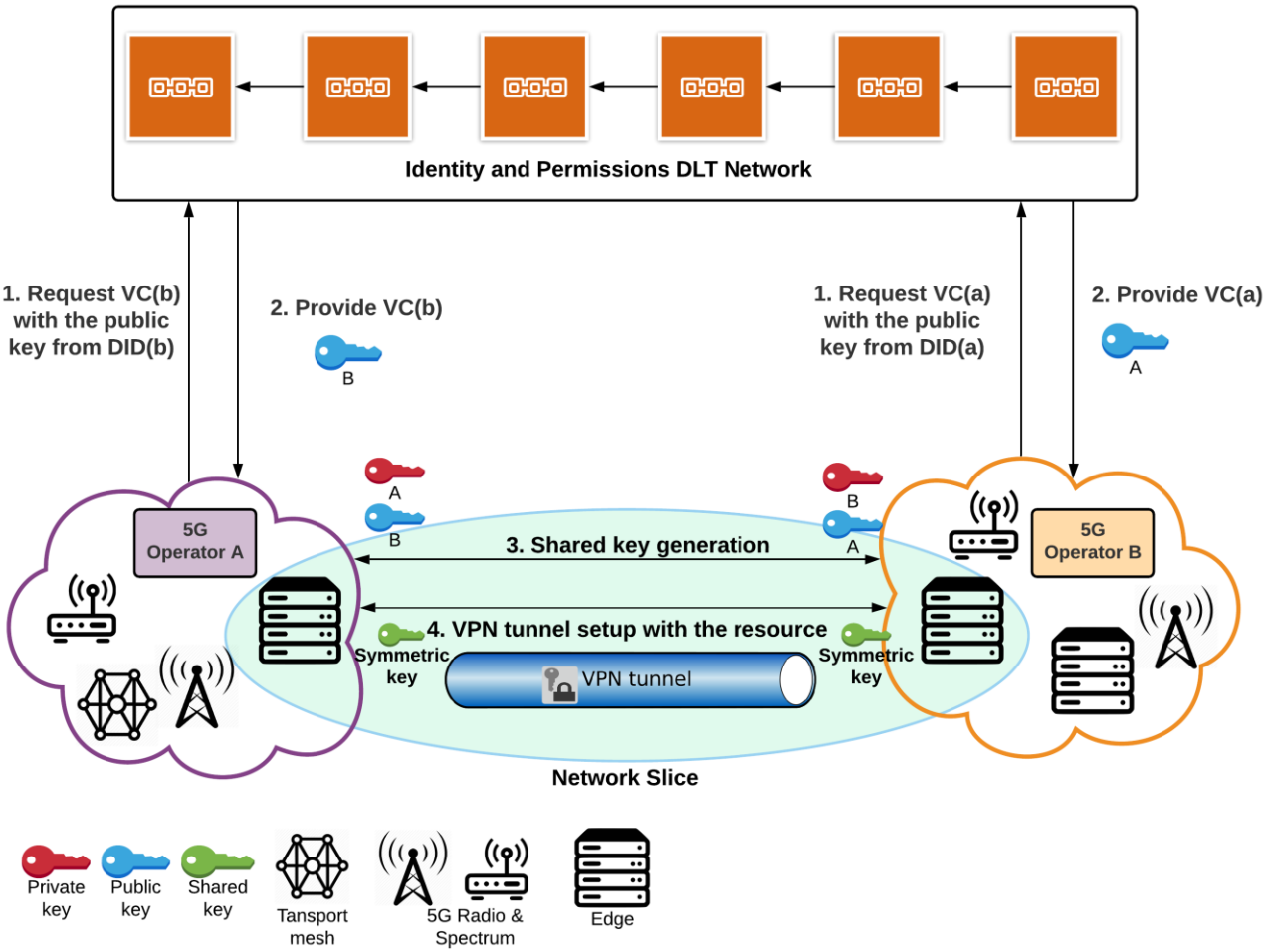

Figure 2-5: Inter-domain security module architecture

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.

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 inter-domain security module will provide the necessary APIs to deploy a Virtual Private Network which utilises DIDs as identification mechanism. In the case of the Table 2-7, this depicts a general view of the main capabilities supported by the inter-domain security module, as well as its level of support (i.e., M-mandatory, O-Optional). The Table 2-8 provides a greater level of detail on the capabilities previously presented in the Table 2-7.

Table 2-7: Definition of Inter-domain Security Establishment service (per-domain/cross-domain level)

Service name: **Inter-domain security establishment**		Type: ***Per-domain & Cross-domain***
Capabilities	**Support (O|M)**	**Description**

Get Server 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 2-8: Definition of Inter-domain Security Establishment service interfaces

Operation name: **getServerConfig**		
Description	This method allows downloading configuration options from server (client side).	
Input Parameters	**Type**	**Description**
ip_address_server	String	It is an end-point where VPN server will be installed.
port_server	Integer	Specific port where VPN server will be available.
Output Parameters	**Type**	**Description**
did	String	Public identifier associated with server stakeholder in the DLT, which will be used to gather server's public key.
config	Dictionary	Dictionary with paramount configuration parameters to consider before establishing a connection.
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.
did	String	Public identifier which will be utilised to gather client's public key.
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

5GZORRO Intelligent and Automated Slice and Service Manager (ISSM) module focuses on automation of managing secure cross-domain slices and services within them. The focus is on how to implement intelligent and automated slice and service management in distributed multi-party 5G scenarios through a policy-based protection approach, whose mechanisms will make it possible to control and preserve data confidentiality, integrity and availability features in information sharing operations.

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.

ISSM is responsible for the optimization of resource allocation to inter-domain slices subject to SLAs and cost-efficiency targets.

ISSM 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. 4.2).
- ***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.

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).

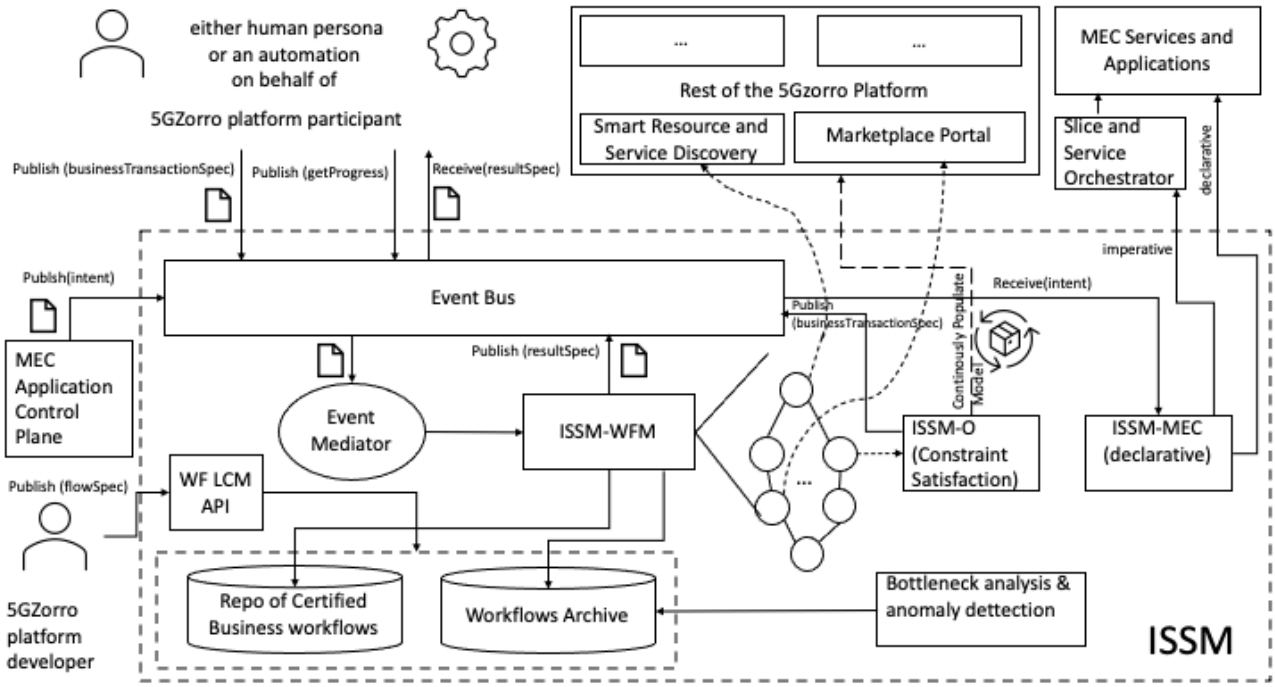

Figure 3-1: High Level Architecture of ISSM

Software Architecture

In Figure 3-2, we describe a software architecture that we would like to 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.

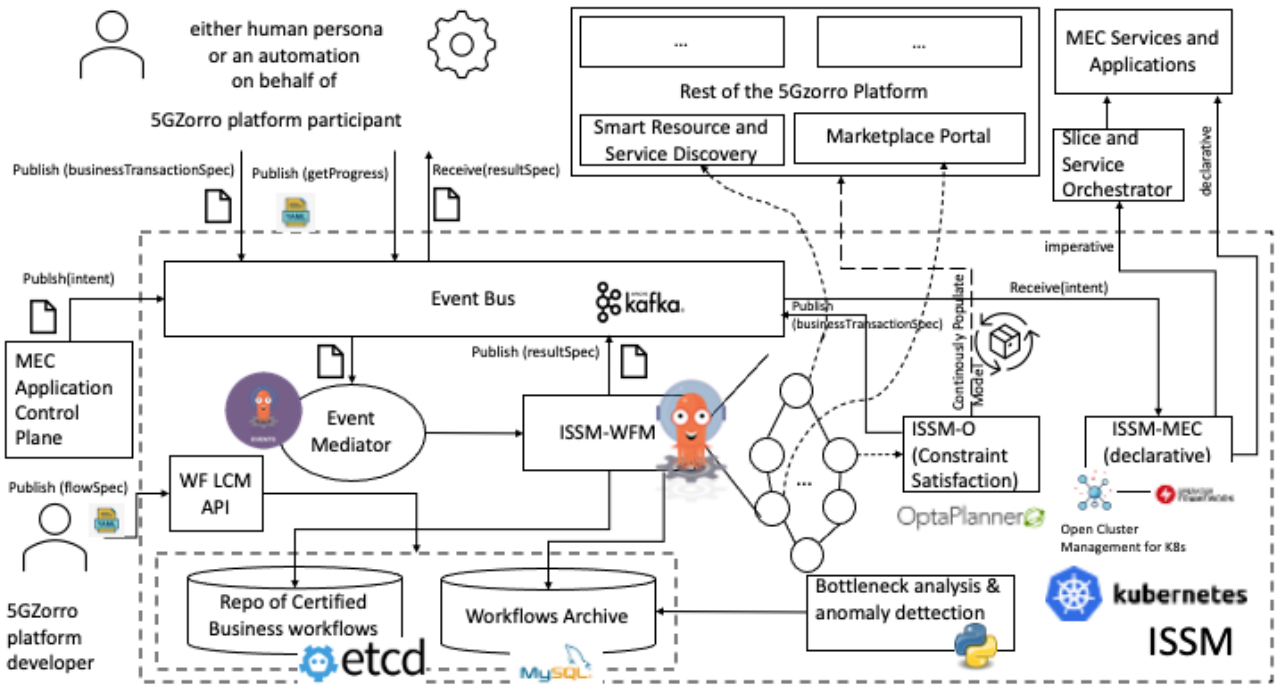

Figure 3-2: ISSM Software Architecture

At the base of the ISSM system there are a number of 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.

Kubernetes (k8s) is a container orchestration platform created by Google and open-sourced after several years of extensive in-house usage and testing. K8s quickly became a de facto industry standard for orchestrating containerized applications and is embraced and governed by the Cloud Native Compute Foundation (CNCF). Recently, it began to gain acceptance as a platform for creating universal control plane and orchestration software, in large because of its declarative approach to specifying the intended deployments and to its openness and extensibility. Ubiquitous acceptance of k8s both as a cloud native development platform and as a cloud orchestration software, increases the appeal of the technology and establishes it as a common denominator both for cloud developers and for cloud providers in terms of the required skills, expectations, and toolchains. In 5GZORRO, we are going to extend k8s in several ways, one of them is to become a basis of 5GZORRO Intelligent Service and Slice Manager described in this section.

Argo [15]: a CNCF project that comprises three sub-projects: Argo Workflows and Pipelines, Argo Events and Argo CI/CD. These projects follow Operator design pattern, native to Kubernetes [16]. An Operator comprises a Controller and a Custom Resource Definition (CRD). CRD is a schema describing new API that we want to add to K8s to extend it to orchestrate our custom resource as if it was a K8s native resource, such as Pod, Deployment, ReplicaSet, etc. Roughly CRD can be compared to class while Custom Resource (CR) is an instance of this class created on demand by applying YAML definition to K8s API Server. A Controller watching CRs of the CRD receives all events pertaining to CRs. A reconciliation function, the main business logic of a controller, continuously reconciles an observed state of a resource to its desired state as represented by CR. This management style is termed declarative, because CR definition represents a declarative intent rather than imperative instructions. Declarative management is cloud native. For Argo a workflow definition CR (instance of Workflow CRD) defined in Argo dialect of YAML represents a desired state of a Workflow (i.e.,

all the dependencies in a task DAG that should be resolved in topological order. Argo Workflows controller reconciles this desired state with the observed state (actually resolved dependencies). Argo Events follows the same pattern. It defines a Sensor CRD to trigger any K8s resource (including Argo Workflow) when an event dependency evaluates to true. Argo events can receive events from more than 20 sources and in particular from Kafka to trigger workflows and other resources. Argo is part of Red Hat OpenShift. In 5GZORRO, we are going to re-use and adapt Argo based event-driven workflow mechanisms developed by IBM in the 5G-MEDIA project to facilitate non-standard orchestration flows [17].

Etcd [18]: is a highly reliable key/value store that is being used by K8s to store all operational data including CRDs and CRs. When a new workflow CRD is created for a new business flow, its YAML definition is stored in the etcd key/value store of K8s. Likewise, if a new event sensor CRD is created, it's being stored in etcd of our orchestration control plane, which is K8s itself.

MySQL [19]: is used as archiving database by Argo. It is not a good practice to overload etcd of a cluster with objects. Therefore, for the sake of scalability the flow information is archived in a distinct database. This database is inspected by optional Bottleneck Analysis and Anomaly Detection module (e.g., in our implementation we are planning to develop a Python based analytics).

OptaPlanner [20]: embeddable open-source constraint optimization engine running on K8s. OptapPlanner is supplied with Red Hat OpenShift. OptaPlanner is based on annotations to Java classes. These annotations are automatically translated into a mathematical problem. The problem is being solved using time constrained informed search optimizing a fitting score of a solution. We discuss Intelligent Network Slice and Service Optimizer component (ISSM-O) in greater detail in the next section.

Red Hat Operator Framework [21]: The Operator Framework is an open-source project that provides developer and runtime Kubernetes tools, enabling developers to accelerate the development of an Operator. The Operator Framework includes:

- Operator SDK: Enables developers to build Operators based on their expertise without requiring knowledge of Kubernetes API complexities.
- Operator Lifecycle Management: Oversees installation, updates, and management of the lifecycle of all of the Operators (and their associated services) running across a Kubernetes cluster.
- Operator Metering (joining in the coming months): Enables usage reporting for Operators that provide specialized services.

Red Hat Open Cluster Management for Kubernetes [22]: Red Hat Open Cluster Management (RHT OCM) is a rich collection of open-source projects that help managing multiple Kubernetes clusters. Multiplicity of Kubernetes cluster is a natural requirement arising in 5G MEC. To meet latency requirements of the 5G standards, multiple far edge data centers running Kubernetes are expected as NFVI. This is true even for the case of a single MNO. A set of the far edge DCs that host the 5G-MEC platform are managed from a central office. OCM is a scalable technology to manage distributed Kubernetes clusters. This open-source project powers Red Hat Advanced Cloud Management for Kubernetes commercial product [23] .

3.1 ISSM-WFM

5GZORRO Platform Participant persona publishes business transaction specification (a document) on a well-known topic of Event Bus. Typical business transaction might take considerable time because it spans multiple components in the 5GZORRO platform as illustrated in D2.2 flows. Therefore, business transactions are executed asynchronously. Business transaction spec is communicated to ISSM-WFM via Event Mediator. ISSM-WFM triggers an appropriate workflow from the workflow repository. Workflow is a document that defines the 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 (cross-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

Figure 3-3 depicts OSSM-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 our initial implementation, 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.

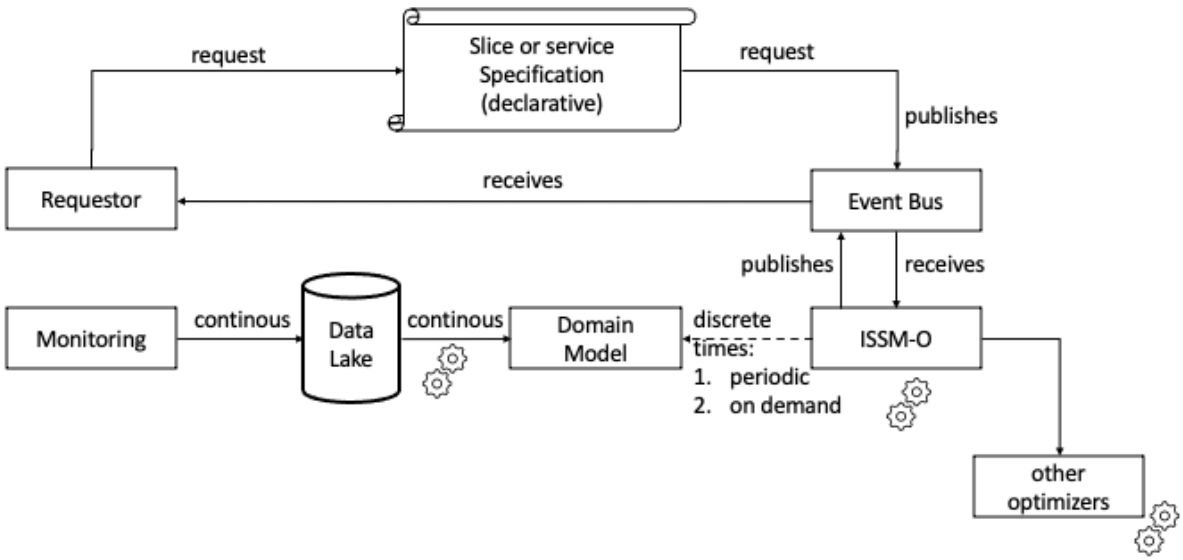

Figure 3-3: ISSM-O High Level Design

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 or Gurobi. In other cases, methods like relaxation and rounding, column generation, etc. can be deployed.

After evaluating a number of possibilities for open-source tools, we have selected OptaPlanner as an open-source tool that can serve as a basis for our design. OptaPlanner is open-source, cloud-native tool that – while not being a commercial grade solver – often shows very good results at the optimization challenges. It is Kubernetes friendly and is being marketed by Red Hat as part of its OpenShift commercial ecosystem.

The main appealing feature of OptaPlanner is that it is essentially a pluggable framework, which is friendly to developer. A standard development cycle in OptaPlanner includes programming an information model as Java classes with annotations. These classes are then translated by OptaPlanner into a mathematical model. OptaPlanner performs guided search, where meta-heuristics, such as Tabu search, hill climbing, simulated annealing, Late Acceptance, etc. can be tuned and configured by a developer. Likewise, a developer can specify own heuristic. Moreover, they can be combined in a flow in a series of programmatically controlled "moves". In principle, even connecting a best of breed solver, such as CPLEX or Gurobi is possible in the future. Constraints can be weighted differently, and soft and hard scores can be controlled.

It should be noticed that while OptaPlanner can be harmonized with ML and AI tools into an optimization pipeline, it is not a tool for AI or ML. We envision, usage of these tools for forecasting and classification (the topics tackled separately) and OptaPlanner is going to be used for optimization.

OptaPlanner is embeddable solver service exporting Web API and we are leveraging this in our design. 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 Optimizer 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 (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 plan to implement an experimental cloud-native MEC platform and to have ISSM-MEC to dispatch declarative deployment requests to it as depicted in Figure 3-1 and Figure 3-2.

Figure 3-4 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. To begin with, we plan to support two actuators: Network Slice and Service Orchestrator (NSSO) described in more details in section 4.2 and the experimental cloud native MEC Platform (CNMP) described in section 3.4. In the future, we envision extensibility to support additional actuators.

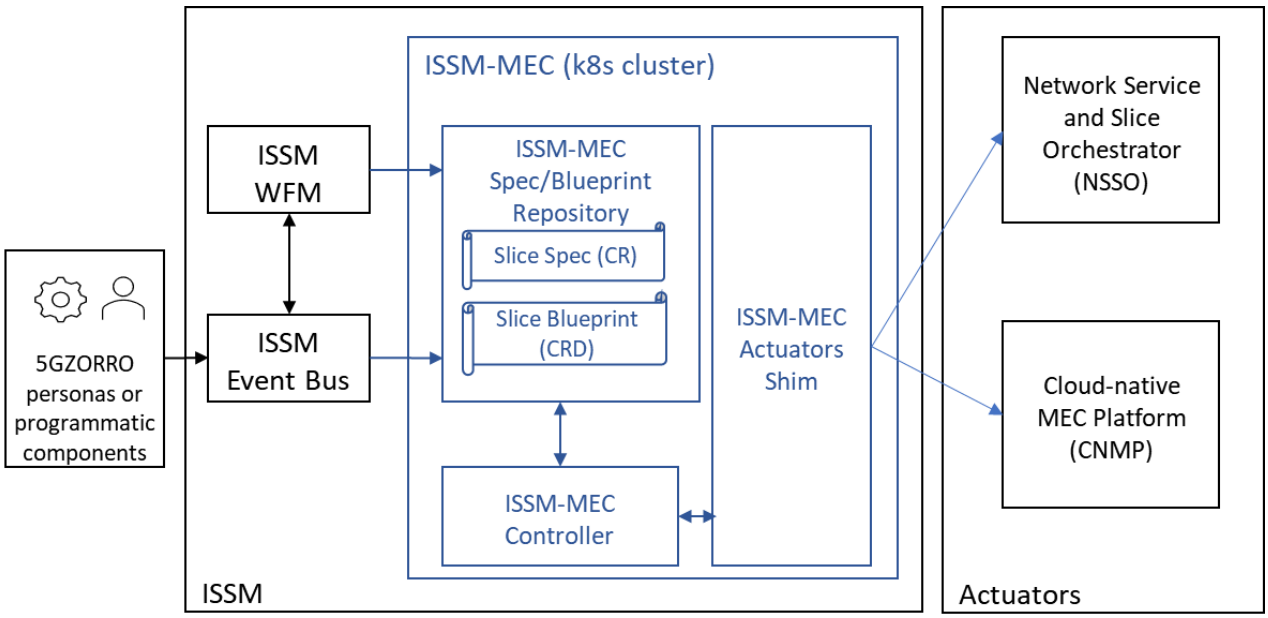

Figure 3-4: ISSM-MEC architecture and its interaction with the rest of ISSM and the external actuators

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 external to ISSM and will be created to represent the emerging cloud native MEC environment, to be managed/federated under 5GZORRO. This new architectural component is 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.

We plan to create CNMP following the 3GPP standards for slicing and the ETSI standard for MEC. This early experimentation will help us to assess the readiness of k8s to support these standards on the one hand, and to observe the readiness of the existing/emerging standards to serve the industry needs. If/when we will encounter gaps, we plan to contribute to the open-source software and/or to the standards.

Figure 3-5 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. As control plane is concerned, we plan to rely on Open Cluster Management for Kubernetes and extend it with capability for slice lifecycle management. Open Cluster Management for Kubernetes is composed of the multi-cluster controller, to be integrated into ISSM-MEC and of the multiple agents installed in the remote managed clusters. All the managed clusters will be enhanced with slice lifecycle management capability. Data plane enhancements to k8s are planned to control the attachment of the k8s slices to the infrastructure network slices or network segments and to establish and control data flows between managed clusters if required. Low level design of these enhancement is not yet completed while we are considering multiple options, e.g., OVN/OVS based CNI with tunnelling support and Submariner multiclause networking solution (https://submariner.io/).

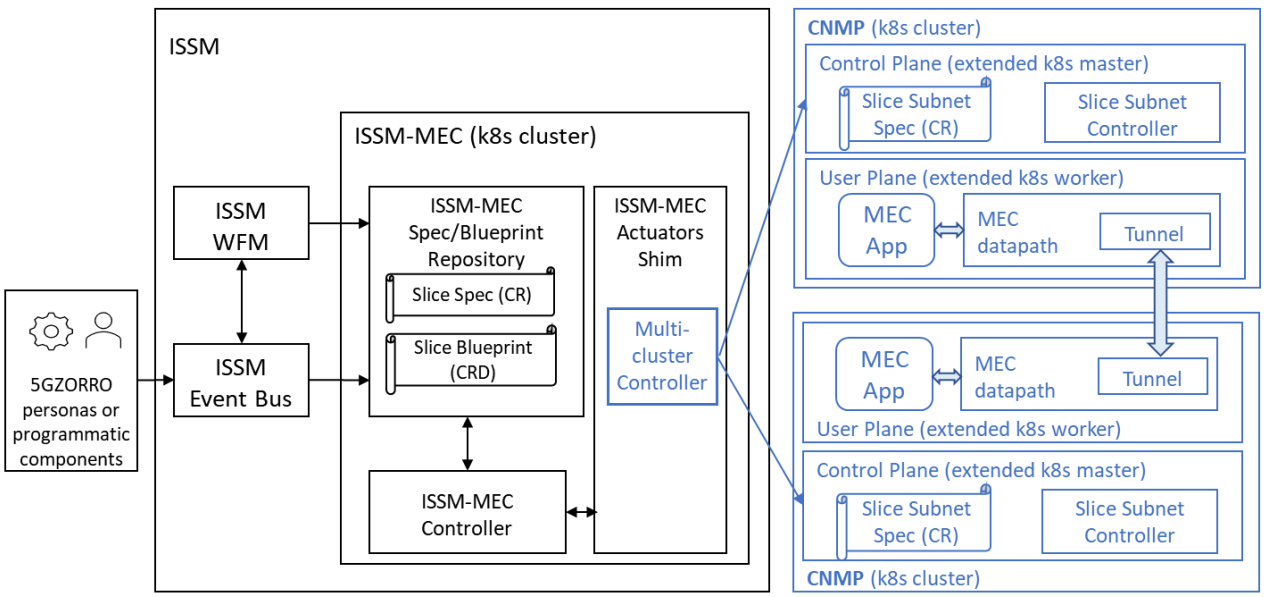

Figure 3-5: Cloud-native MEC platform

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

4.1 Virtual Resource Manager

The Virtual Resource Manager (VRM) is a module in the 5GZORRO platform that directly interacts with the underlying 5G Virtualized platform. Due to its positioning in the 5GZORRO architecture, the VRM offers the upper layers a set of services related to the resources monitoring and management that also includes a direct support to the 5GZORRO Offering Catalogue. Despite the fact that VRM can be involved in the orchestration and monitoring of resources belonging to services and slices deployed across different domains, all the services it exposes are considered to be per-domain.

The VRM module in the 5GZORRO architecture is the entity controlling the deployment of all the resources available in the Marketplace. The diversity of resources available in the 5GZORRO platform, which includes computational, storage, network, and RAN resources, makes it essential to handle them differently. More specifically, the 5GZORRO platform differentiates RAN resources (base stations, backhaul links, and access points) and virtual resources. More importantly, the efficient utilization of resources by the resource provider and the consumer must be guaranteed. This is achieved using a service in charge of triggering the generation of resource monitoring data. The VRM module encompasses these three functional elements in the 5GZORRO platform, naming the Service and Resource Monitoring (SRM, for what concerns the Resource monitoring), the Virtual Resource Management and Control (VRMC), and Radio Resource Management and Control (RRMC), already defined and discussed in D2.2.

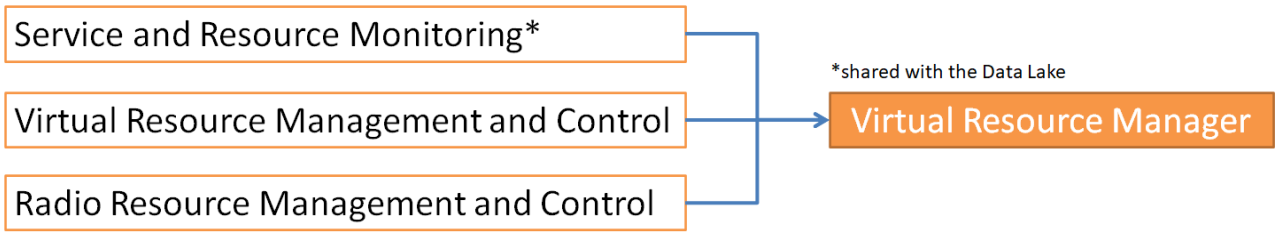

Figure 4-1: Virtual Resource Manager comprising Service & Resource Monitoring service, Virtual and Radio Resource Managers

The RRMC module is responsible for managing the RAN elements in the platform. The management can be either passive or active. This module is compatible with the most wide-spread wireless technologies, including LTE, 5G, and Wi-Fi. The RRMC will configure the necessary RAN elements required by a given service, and it supports a range of manufacturers or RAN equipment vendors. Each vendor has its private Operation and Maintenance Service (OMS) protocol. Therefore, RRMC should implement an adaptation layer for each specific vendor. The RRMC has a modular design and enables adding new vendors as new adaptation layers are added.

The Virtual Resource Manager has the following KPIs:

- 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*)
- Mobility support, meaning that a user will be handed over to the best cell as it moves and goes beyond the coverage area of its serving cell. Only applies for cellular connections (LTE or 5GNR). *Possible KPI target: Number of Radio Link Failure (RLF) below a threshold*
- The RRMC is capable of optimising the allocation of the RAN resources without degrading the QoS of any of the services. For instance, a given Wi-Fi service agreed to provide 30% of airtime, but only 20%

of resources are actually used. The RRMC can optimise the resource allocation and freed the idle 10% of resources for other services. *Possible target KPI: QoS satisfaction rate larger than a threshold (e.g., > 99%), or ratio between delivered service over offered service > 0.99*

- 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)*

Resource Management Service

In order to orchestrate the virtual resources, they should be already present inside the 5G Virtual infrastructure since the orchestration stack, which is in charge of the lifecycle management of such resources, expected to have them ready to be manipulated. The VRM offers a set of services in this sense.

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.*
Manage Network resources	**M**	*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 VRM 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 • MEC_APP • 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.	
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: **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

VRM offers a set of features that allows the monitoring of relevant parameters of the 5G Virtualized infrastructure. In particular, the VRM enables monitoring at Cloud (NFV and Kubernetes), Radio and Network (SDN) layer. For what concerns the specific case of NFV, ETSI defines in [39] a long list of parameters that can be monitored to measure MANO performances. For the rest of the infrastructure, the data model will be defined and refined step by step, according to the requirement of the 5GZORRO Platform.

The values collected by the VRM should be able to be requested on demand through an adequate interface which must also offer the possibility to create proper data streams with the scope of injecting the monitored

parameter into the 5GZORRO Platform Data Lake. In this regard, the infrastructure provider must guarantee the origin of the data hence, the VRM must sign the set of measures before sending the to the Data Lake.

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**	Collect and inject monitoring data about the NFVI resource usage and VNF performance/fault statistics 5GZORRO Data Lake.
Provide RAN slice sub-net statistics	**M**	Provide regular information regarding the status of a RAN slice sub-net. This information is stored locally and injected in the Data Lake.
Provide Network statistics	**M**	Collect and injects monitoring data concerning Virtual Network infrastructure in the 5GZORRO Data Lake.
Notes		

Table 4-4 Definition of Resource Monitoring service interfaces

Operation name: **setMonitoringProcess**		
Description	Configure the VRM monitoring service to collect and push to the Data Lake certain set of parameters, with a certain period. The call will return an ID to identify the Monitoring Process set.	
Input Parameters	**Type**	**Description**
period	Int	Period (in second) for collection data from the underlying monitoring platform(s).
signed	Boolean	Specify if data collected should be signed by the provider before sending them to the Data Lake.
sendTo	string	Endpoint exposed by the consumer of the data (i.e., Data Lake).
dataSet	Object/dictionary	Complex object (e.g., Dictionary or Map) which represent the set of data to be collected.
Output Parameters	**Type**	**Description**
monitoringProcessId	String/UUID	ID that uniquely identifies (in the domain) the monitoring process in progress.
Notes		

Operation name: **getMonitoringProcessList**		
Description	Returns the list of the Monitoring Processes in progress (IDs).	
Input Parameters	**Type**	**Description**
none	N/A	N/A
Output Parameters	**Type**	**Description**
monitoringProcesses	list	List of monitoring processes IDs.
Notes		

Operation name: **teminateMonitoringProcess**		
Description	Terminates a monitoring process in progress.	
Input Parameters	**Type**	**Description**
monitoring_process_id	String/UUID	ID that uniquely identifies (in the domain) the monitoring process in progress.
Output Parameters	**Type**	**Description**
response	HTTP response	HTTP Code and boy that provides the result of the Termination operation.
Notes		

Support to Offering Catalogue

The Virtual Resource Manager also supports the 5GZORRO Catalogue by adding virtual or radio resources that are to be shared or traded later in the platform. The Virtual Resource Manager is responsible for maintaining an inventory of the resources available, which implies that the Virtual Resource Manager can update the details of a given resource and, also, remove a shared resource from the Catalogue.

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 VRM 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 VRM 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 • MEC_APP • NETWORK_SLICE • RADIO_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 Orchestration

The Network Slice and Service Orchestration is responsible for the automated lifecycle management of the Network Slice (NS) Instances supporting vertical services. This module acts both at the inter-domain and the intra-domain layer of the 5GZORRO architecture. At the inter-domain layer, this module manages the lifecycle of the end-to-end NS mapped to the service, and the split of this end-to-end slice into NS and Network slice subnets (NSS) which will be provisioned by the different domains. At the intra-domain layer, this triggers the lifecycle management actions of the NS and NSS to be provisioned completely intra-domain, interacting with the Virtual Resource Manager for the provisioning of the resources.

This module will be based on the Vertical Slicer designed in 5G-TRANSFORMER (called 5GT-VS) [38], and currently being developed in 5Growth. The 5GT-VS allows the translation of the business level requirements, such as the number of users, the number of devices connected, etc., into NS. Therefore, vertical service

instances are mapped to a number of network slice instances. Depending on the specific constraints, the NS can be created on demand or re-used from the available slices, depending on the specific constraints. Vertical services are defined using "templates" called *Vertical Service Blueprints (VSBs)*, which are kept as part of the 5GT-VS Vertical Slicer catalogue. This template is customised using *Vertical Service Descriptors* (VSDs), which are stored in the internal catalogues of this module and allow setting specific values for the blueprint inputs. A service can then be requested using a reference VSD, which the Vertical Slicer automatically translates into a set of NSs and NSSs. Then, the 5GT-VS, automatically translates the vertical service lifecycle management actions into network slice level actions.

Within 5GZORRO the development will mainly target three different enhancements:

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 [41] . This new catalogue will allow a standard way of defining and requesting vertical services and network slices.
3. Enhanced support for multi-domain scenarios: The multi-domain capabilities of the Network Slice and Service Orchestration will include multi-domain capabilities. Currently, the 5GT-VS supports the interaction with multiple domains at the NS level by using a driver-based approach. In other words, to request NSs and NSSs to different domains using a specific domain driver, usually relying on REST requests. This approach in 5GZORRO is to adopt the principles of the service-based architecture, and therefore the target in this sense is set into updating the software architecture to leverage the inter-domain Communication Fabric (described in [31]) for the inter-domain NS lifecycle management.
4. Edge service orchestrator: The mobile edge platforms that manage at certain locations of the 5G infrastructure (e.g., base stations) provide performance and latency improvements in network slices. The NFV MANO should coordinate with such platforms, in order to obtain their list of offered resources and services. Within 5GZORRO a plugin will be developed to make this information available to both NFV MANO as well as the resource and service catalogue of the 5GT-VS. Then, the 5GT-VS will 1) act as a service orchestrator for the extension of network slices at the edge level and 2) support *Vertical Service Blueprints* with edge enhancements.

Finally, the use of the Slice Manager developed in 5GCity [24] is also envisioned at the NS level. Following the modular principle of the 5GZORRO architecture, this integration enforces the expected flexibility for the lifecycle management of NSIs by supporting several components acting as Network Slice Management Function (NSMF). In addition, the use of the 5GCity Slice Manager will enable the integration with i2CAT's RAN Controller. This RAN Controller acts as a Network Slice Management Function (NSMF) that provides slicing support and neutral host capabilities (i.e., sharing of common infrastructure resources among different slice tenants) over different Radio Access Technologies (e.g., WiFi, LTE). To achieve this integration with the Vertical Slicer, the Slice Manager will be extended to support a NEST-based NS provisioning.

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. The starting point for the OpenAPI specification is available in [40]. The specification for the operations related to GST/NSTs will be specified in the next release of this deliverable.

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.
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.
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.
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 Networks Service Mesh Manager (NSMM) is to provide the functionalities required for the stitching of slice and service across multiple domains. The virtualization platforms the services and slices rely on can be heterogeneous (e.g., VM, Containers) with the aim of providing a true end-to-end multidomain slicing and service context, agnostic with respect to the underlying virtualization technologies. A crucial aspect of the stitching process is the secure cross-domain inter-connection, provided by the Inter-domain Security Service described in Section 2.4, as part of the 5GZORRO Security and Trust Orchestration stack. Due to this, a direct interaction between the two modules is mandatory.

The main consumer of the services exposed by the NSMM is the NSSO (Network Slice and Service Orchestrator, see Section 4.2) that maintains a complete view of the end-to-end slices and services. This interaction implicitly shows three important aspects that characterize the design and the consequent development of the NSMM:

1. Any possible action it can perform is always part of an orchestration workflow
2. Since the NSSO takes care of the end-to-end service/slice, it can interact with the NSMM of each domain
3. Intra-domain heterogeneous slice/service stitching actions (e.g., to create a slice with a VNF and a CNF) are always managed locally

The immediate consequence is that NSMM is not required to provide any cross-domain service: any configuration required to establish a cross-domain connectivity is performed on the NSMM on the local domain. In Figure 4-2 shows a preliminary scheme of the NSSM architecture.

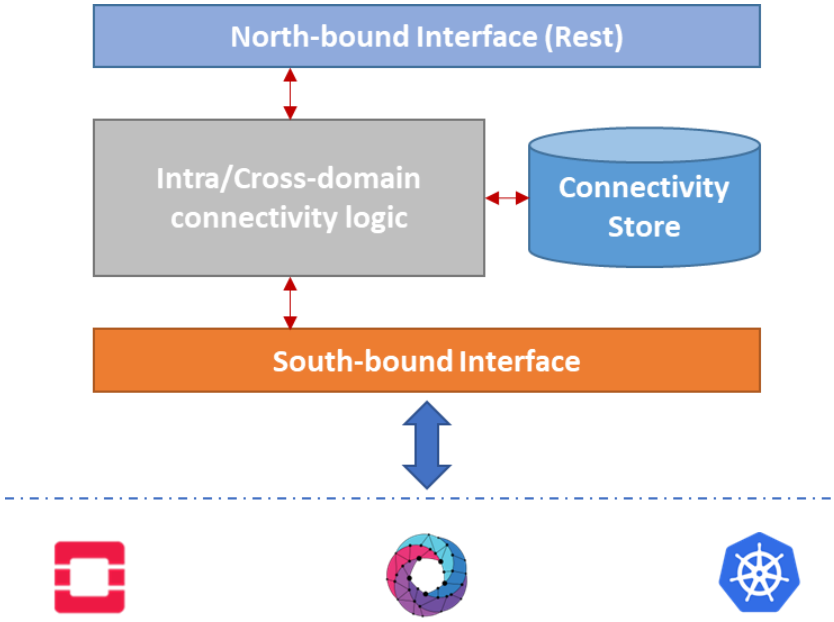

Figure 4-2 Simple NSMM architecture

The interface exposes a set of REST APIs, described in the tables below, towards the orchestration stack. The requests are forwarded to the internal NSMM connectivity logic that, based on the type of the request and the set of parameters specified, distinguishes the following cases:

- Intra-domain service/slice stitching between heterogeneous virtualization platforms (hybrid stitching)
- Cross-domain service/slice stitching (hybrid or not)

In this last case, the NSMM should take care of establishing the VPN-based secure connection between the involved domains. In this sense, each NSMM instance running in the domains involved in the e2e slice deployment should configure the slice network endpoint locally by building and applying proper configurations to set it as a VPN client or a VPN server.

The established connections are stored in the Connectivity Stores that keep track of the intra-domain connections and, partially, the cross-domain ones. Since the NSMM acts locally to a domain, and the VPN follows a client-server paradigm, the Connectivity Store only contains the connectivity information with the adjacent domains. In other words, each local instance of NSMM only maintains information on the connection segments towards adjacent domains. This way to store the information is discussed in Section 5.7 where it defines the information model of the Connectivity Store.

The objective of NSMM is the following:

> *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).*

Table 4-10: Definition of Intra and Cross-domain slice stitching service (per-domain/cross-domain level)

Service name**: Intra and Cross-domain Slice Stitching**		Type**: *Per-domain/ Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Create slice connectivity	**M**	This capability allows the establishment of the connectivity between slices. Based on the set of parameters, the NSMM recognizes whether the VPN (cross-domain connectivity) is required.
Destroy slice connectivity	**M**	This capability allows the termination of the connectivity between slices. Its scope is to clean the network stack form old slice stitching configurations and prepare the environment for new ones.
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 Intra and Cross-domain slice stitching service interfaces

Operation name: **createSliceConnectivity**		
Description	This API performs the local configuration to enable the intra/cross-domain end-to-end slice connectivity. It requires also the specification of the parameters for the creation of the VPN which composition depends on the type of endpoint (see below). NSMM internal logic will transform the input parameters into a set of calls to the APIs exposed by the interconnectivity modules (see Section 2.4).	
Input Parameters	**Type**	**Description**
slice_id	UUID	Slice instance id local to the domain.
endpoint_type	Enum	This parameter specifies the role of the endpoint of a slice in the context of a VPN: SERVER, CLIENT, LOCAL. In case of LOCAL endpoint, the connectivity between the various entity happens in the same domain and no VPN is required.
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. NSMM internal logic will transform the input parameters into a set of calls to the APIs exposed by the interconnectivity modules (see Section 2.4).	
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 Management

In this era of software disruption, caused by the softwarisation of hardware, virtualisation and Anything as a Service (XaaS) approaches, new mechanisms are required from software vendors that need to materialise the revenues on their development investments and intellectual property rights associated with them, applying licensing costs to their products according to their business plans using an automated implementation. Regarding the telecom sector, Network Functions (xNFs, that encompasses VNFs, CNFs or network functions composed by several VNFs/CNFs) are network software functions that can be instantiated and replicated very quickly, thanks to the NFV technology in a multi-domain ecosystem. But this agility increases the challenge of the license control and management.

The licensing framework encloses a set of tools to bring service providers and software vendors the capabilities to interact in a transparent, flexible and secure way. Both stakeholders will be part of the 5GZORRO ecosystem, trading with the software products with a minimum human intervention, where the software vendors play the role of xNF providers and service providers as xNF consumers.

Vendors enrolled in the platform can onboard their software resources, exposing capabilities, licensing constraints and the agreements associated. 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 licensing framework tracks the usage of the resource in real-time, verifying the compliance of the smart contract and acting in the licensing costs.

Since one of the targets of 5GZORRO is to share assets between different parties, there is a need of enable the mechanisms to involve a fair software purchase system that is enabled by the eLicensing framework. The main features of this framework are:

- No block of software usage waiting for license key verification. The software is deployed in the operator premises or in third-party/external premises, and the usage will be controlled regarding the licensing agreements reflected in the contract between the parties.
- 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.
- The eLicensing Manager Agent installed inside the domain will serve to all vendors. With this agent, all software vendors can control the usage of their software products and support all their licensing schemas.
- Enables full automation, there is no need of human intervention. Programmatic interfaces are released to interact with the platform.
- NFV Orchestator agnostic. This provides the capability to deploy the software and proceed with their control independently from the underlying virtualization infrastructure.

The KPIs related to the eLicensing 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.
- **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 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 deliverable 3.1.
- **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. There are two components designed to manage the location independent xNF usage: (I) The eLicensing Context and Evaluation Manager (LCEM) act as the brain of the licensing management, has a full view of the status of the usage of the xNFs declared in every contract and (II) The eLicensing Manager Agent (ELMA), that is the licensing agent that is local to each domain.

(I) The eLicensing Manager Agent is deployed over each domain, and its internal functional blocks are depicted in Figure 4-3: eLicensing Manager Agent functional blocks. The ELMA verifies the usage of the xNFs with licensing agreements that are running.

In order to support the business models previously defined in D2.2, the e-Licensing Manager Agent retrieves the business licensing agreements from the marketplace, using the agreements API. The translator functional block makes the map between the business agreements to the technical actions that are controlled, that is, if for example certain xNF follows a subscription model, the action that needs to be observed it is the time. The result of the translator is used for the next block, the watcher generator. The watcher generator configures entities called watchers before the xNF deployment. Watchers are dedicated pieces of code that observe the usage of an xNF depending on their scope. They real-time control the xNFs for operational (scaling up/down decisions from the NFVO, time of usage) or inside their business logic (as number of users connected to a database or storage used in a steaming application).

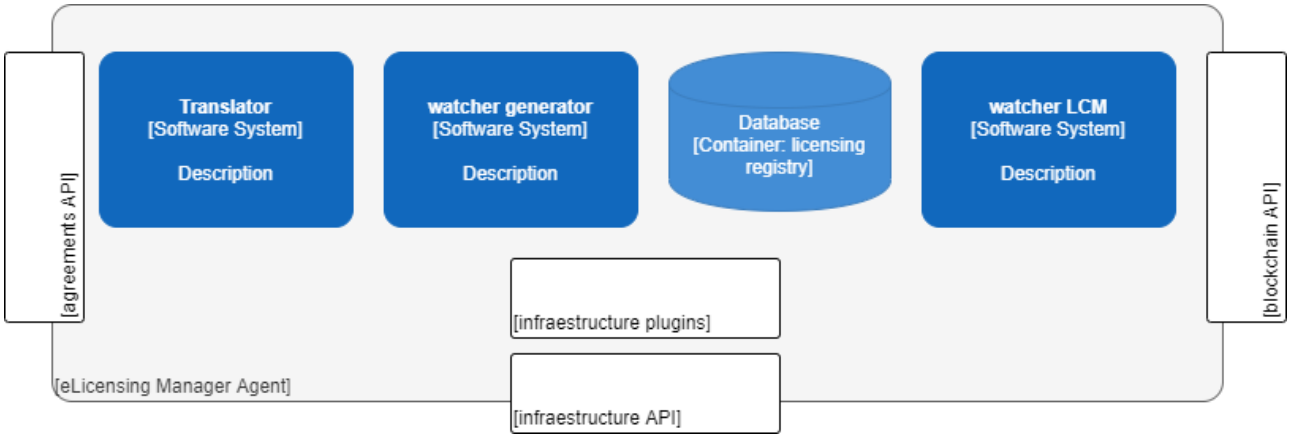

Figure 4-3: eLicensing Manager Agent functional blocks

The watcher LCM retrieves the results from the watchers and request the gathering of the actions in the DLT using the blockchain API.

(II) The eLicensing Context and Evaluation Manager is the component of the framework for supporting the xNF scaling and replication in a multi-domain fashion. This centralized component is responsible to keep the context of the licensing usage for each xNF.

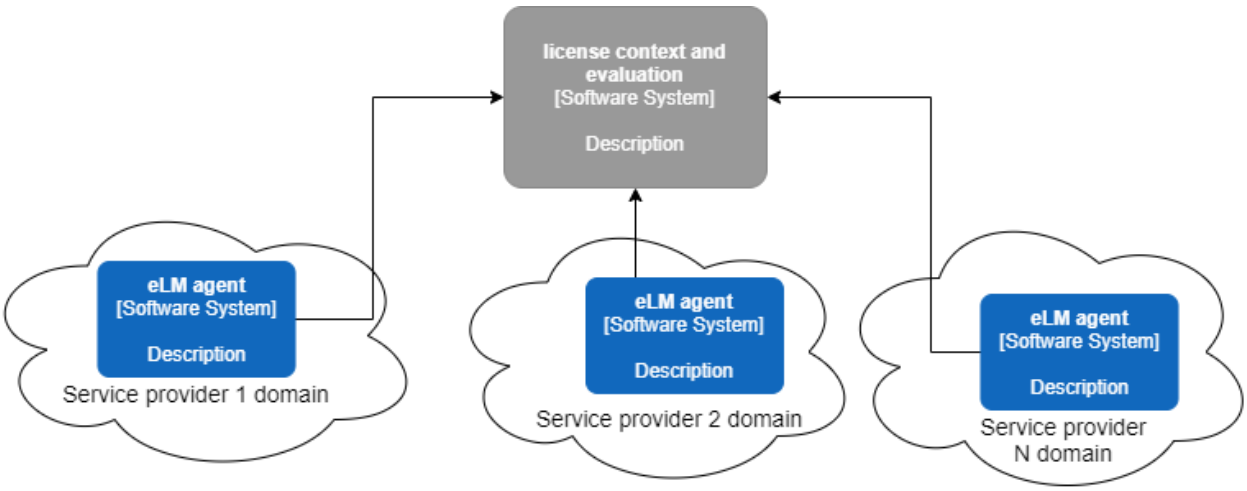

Figure 4-4: eLicensing Context and Evaluation Manager

It has a global view of the status of the usage of the xNFs declared in every contract, independently of the location, domain owner or underlying infrastructure technology. It is responsible to synchronize the information (such as agreements related to the product, watchers generated) gathered in the ELMAs to other ELMAs in case of scaling or replicating xNFs to other domains.

It is also equipped with a notification system to inform the involved stakeholders in case of contract expiration or breach of the agreed contracts regarding the actions gathered in the DLT and the licensing agreements reflected in the signed smart contract.

To support these capabilities, Table 4-12 and Table 4-13 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/cross-domain level)

Service name**: e-Licensing Management**		Type**: *Per-domain/ Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Read resource agreements	**M**	Retrieve the licensing terms for the product from the Marketplace
Generate licensing watchers	**M**	Create, remove, update the licensing watchers to observe the NF licensing events
Trigger action record	**M**	Create the action record request in the DLT.
Notes		

Table 4-13: Definition of e-Licensing Management service interfaces

Operation name: **getAgreements**		
Description	Get the agreements from the marketplace of a certain product.	
Input Parameters	**Type**	**Description**
productDID	String	An identification code of the product in the 5GZORRO platform.
Output Parameters	**Type**	**Description**
List< AgreementDetail>	List of agreements	List of resource agreements related of each component of the service.
Notes		

Operation name: **getManoIds**		
Description	Retrieve from the Vertical Slicer the IDs to identify the xNF in the MANO.	
Input Parameters	**Type**	**Description**
productDID	String	Product identification.
Output Parameters	**Type**	**Description**
List<Vnf-id>	List of String	List of MANO resource identifier for each xNF.
Notes		

Operation name: **checkLicensing**		
Description	Notification to the e-Licensing manager to make it aware that a new product is ready for the deployment.	
Input Parameters	**Type**	**Description**
productDID	String	An identification code or number of the product in the 5GZORRO platform.
Output Parameters	**Type**	**Description**
ACK	Integer	Indicates that the licensing check has been performed.
Notes		

Operation name: **persistAction**		
Description	Create the action record request in the DLT.	
Input Parameters	**Type**	**Description**
AgreementDetail	Resource OfferPrice	Resource agreement related.
Action	ENUM	Action related. TIME, N_INSTANCES, N_GB, N_USERS, etc.
Vnf-id	String	MANO resource identifier for each NF.
Output Parameters	**Type**	**Description**
error	Boolean	Indicates if the action is correctly recorded.
Notes		

5 Information Elements

5.1 Trust Management Framework Information Model

Due to the fact that no trust information models is available in state of the art, this section proposes an original UML design for the trust management framework. In this vein, the information model for the trust management framework 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 [33], [34] and service offer information model [35] 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.

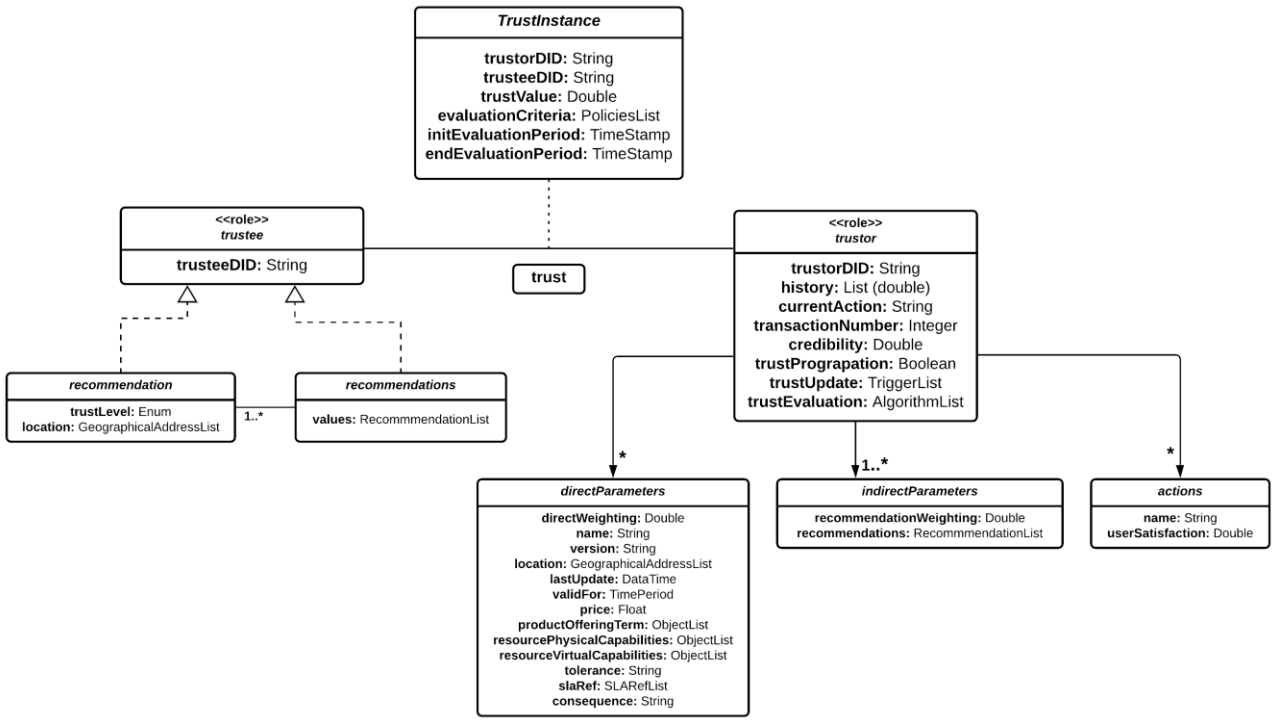

Figure 5-1 : UML diagram of Trust Management Framework

Table 5-1: Trust 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.

evaluationCriteria	List of intra or inter-domain policies	Criterion selected by trust model to assign the values to 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.
recommendation	List of objects	A recommendation about a third-party.
trustLevel	Enum	Possible trust levels such as low, medium, and high trust.
location	List of GeographicalAddress objects [32]	It constitutes a group of GeographicalAddress
recommendations	List of recommendations	Set of recommendations about a third entity from one or more external entities.

Table 5-3: Trustor Entity Information Model

Parameter	Type	Description
trustorDID	String	Unique identifier for a resource or service consumer.
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.
name	String	Name of the resource or service.
version	String	The version of resource or service offer.
location	List of GeographicalAddress objects [32]	It constitutes a group of GeographicalAddress.
lastUpdate	DataTime	Date and time of the last update of this resource or service.
validFor	TimePeriod	The period for which this resource or service is valid.
price	Float	A positive value determining the amount of money.
productOfferingTerm	List of objects	Terms of the offer (e.g., duration, conditions).
resourcePhysicalCapabilities	List of objects	A list of operation band values.
resourceVirtualCapabilities	List of objects	A list of operation band values.
tolerance	String	Allowable variation of the metric.
slaRef	List of SLARef	Service level agreement associated reference.
consequence	String	Action to take as a result of SLA Violation.
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.
currentAction	String	Name of the action/task involving a trust assessment.

actions		List of objects	
	name	String	Name of the action/task involved in a trust assessment.
	userSatisfaction	Double	Internal assessment of the service provided.
transactionNumber		Integer	Number of transactions carried out by the third-party with the other domains.
credibility		Double	Factor that determines how accurate the recommendations are.
trustPropagation		Boolean	Intra or inter-domain trust score and parameterTuple propagation (0 means intra, 1 means inter)
trustUpdate		List of objects	It indicates the triggers to recompute trust score.
trustEvaluation		List of objects	It identifies different evaluation algorithms.

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) [8] 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 Intra-domain Security Information Model

The information model of the intra-domain security module depends on the available information of 1) the 5GZORRO modules, resources, and services as well as 2) the 5G network infrastructure segments as the mobile core, RAN, transport and edge. 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 [25]. 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 5-4: Information model of the intra-domain security module

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 Inter-domain Security Information Model

The inter-domain security information 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 in the DID DLT, 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 inter-domain security service.

Table 5-5: 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* and *AllowedIPs*).

Table 5-6: 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 Virtual Resource Manager Information Model

In this section we describe the information models implemented by the Virtual Resource Manager (VRM). In particular, one of them represents how the information of the managed resources are stored inside the VRM internal catalogue while, the other ones, model resources themselves (e.g., VNF).

Table 5-7: Virtual Resource Manager – Resource information model

Parameter	Type	Description
ResourceID	UUID/String	Unique id of the resource indexed form the VRM. The id is locally unique.
ResourceType	ENUM	Type of resource: • VNF • CNF • PNF • MEC_APP • NETWORK_SLICE • RADIO_SPECTRUM • RAN
Status	Tuple (ENUM)	A tuple with a couple of values: (exposed, deployed): • EXPOSED/NOT_EXPOSED the resource is sealable or not • DEPLOYED/NOT_DEPLOYED the resource is deployed or not

As described in Table 5-7, the VRM stores information concerning different types of resources belonging to the 5G Virtualization platform. Each of them implements its own information model. In the design of the VRM internal store, these models are aligned with standards, as described next.

Network services, Virtual and Physical networks functions, and their respective descriptors are aligned to the standard ETSI SOL 006 [26], which follows the models previously defined in ETSI GS NFV-IFA 11 [27] (VNF(D), NS(D)) and ETSI GS NFV-IFA 14 [28] (PNF(D)). MEC Applications follow ETSI GS MEC 010-2 [29]

Network slice model has been already discussed in D2.2 and it is aligned to GSMA General Slice Template (GST) [30].

For what concerns the Radio resources, at the time of this writing, there are no defined standardized information models to abstract them as virtual resources. A set of information models, defined for the purpose of 5GZORRO, are described in the following tables.

Table 5-8 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.
operationMode	String	The operation mode to be used in the spectrum resource can be TDD or FDD.
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 5-9 RAN Resource Information Model

Parameter	Type	Description
ranResource	UUID/String	Unique ID of the RAN resource.

type	String	The type of RAN resource. Possible values may include cell, access point or backhaul link, and more.
location	String	The geographical coordinates the RAN resource is physically installed. They can be expressed in GPS coordinates or in a human-readable form (e.g., City, address).
technology	String	The wireless technology of the RAN resource (Wi-Fi, LTE, NR).
technologyVersion	String	The version of the wireless technology. For Wi-Fi: e.g., AC, 6; for cellular: Rel8, Rel15).
operationBand	List (Numeric)	List the supported operation bands (Cellular base station, 3gpp, MHz) or channel number (WiFi).
minDlFrequency	List (Numeric)	List of the minimum Downlink frequency (MHz) that can be used for each operationBand in list.
maxDlFrequency	List (Numeric)	List of the maximum Downlink frequency (MHz) that can be used for each operationBand in list.
minUlFrequency	List (Numeric)	List of the minimum Uplink frequency (MHz) that can be used for each operationBand in list.
maxUlFrequency	List (Numeric)	List of the maximum Uplink frequency (MHz) that can be used for each operationBand in list.
bandwidth	List (Numeric)	Lists the supported system bandwidths of each operationBand.
txPower	Numeric	The maximum transmission power in dBm.
quota	Numeric	The percentage of the passive resources shared (e.g., backhaul link capacity, baseband processing capacity, etc.) expressed in percentage.

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 5-10, extracted from [37], describes the structure of a VSB.

Table 5-10 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 5-11, extracted from [37], describes the structure of a generic VSD.

Table 5-11 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 [30]. The information model and the relevance within 5GZORRO of these two latter elements has already been reported in [31] .

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[32].

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 4-2)

Table 5-12 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.
Type	ENUM	Establishes if the connection is intra or cross-domain. Possible values: • INTRA_DOMAIN • CROSS_DOMAIN
Endpoints	List	List of endpoints object. See Table 5-13.

Table 5-13 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 5-14 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 5-15 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).
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 eLicensing Manager. As described in section 4.4, the eLicensing Management has two components.

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.

Table 5-16 ELMA Information Model

Parameter	Type	Description
productDID	String	Unique identifier for a resource or service consumer.
timestamp	DataTime	The time when the product was deployed in the ELMA.
resources	List of resources	List of descriptors of each resource. Resources are defined in Table 5-17.

Table 5-17 Resource descriptor Information Model

Parameter	Type	Description
resourceID	String	Unique identifier for a resource
actions	String	TIME, N_INSTANCES, N_GB, N_USERS, etc.
Infrastructure	String	OSM, K8S, etc.
instances	List of instances	Running resources in the domain. Instances are defined in Table 5-18.

Table 5-18 Running resource Information Model

Parameter	Type	Description
instanceID	String	Unique identifier for an instance.
wendpoint	String	Watcher endpoint in the VNF Manager
wbook	String	Watcher playbook.
wstatus	String	Last status from watcher.

The LCEM has a global view of the status of the usage of the xNFs declared in every contract, independently of the location, domain owner or underlying infrastructure technology. It takes care of the context update in case of scaling/migrating an xNF to another domain and synchronize the ELMAs involved.

Table 5-19 LCEM Information Model

Parameter	Type	Description
productDID	String	Unique identifier for a resource or service consumer.

timestamp	TimePeriod	The time when the product was deployed in the ELMA.
businessAgreements	productOfferPrice	Agreements signed between the parties
instances	List of Tuple	List of <instances-domain> deployed. Maps the instance (see Table 5-18) and the domain id where it is deployed

6 Conclusions

This deliverable provides a detailed design of part of the architecture and 5GZORRO core platform. In particular, this report covers security and trust orchestration, intelligent and automated slice & service management, and the MANO and slicing tools enhancements.

The architecture presented in this document is based on deliverable D2.2, where the 5GZORRO high-level reference architecture was introduced. This report has carried out an iteration on the basic architecture to improve the implementation details and the functionalities associated with 5GZORRO services. In order to succeed in our commitment, multiple set of interfaces, information models, and 5GZORRO specific enhancements have been described 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 implementation work that will be carried out in deliverables D4.2 and D4.3, as well as WP5. Thus, the architecture of a trustworthy and intelligent network slice management mechanism for building secure cross-domain slices and services will be iterated and improved in tasks T4.2 (intermediate prototype) and T4.3 (final prototype). Finally, the presented architecture will be also developed in task WP5, through main use cases contemplated, and presented in the validation deliverables D5.1, D5.2 and D5.2.

Table 6-1: D4.1 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).*	n/a
	• *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 Trust management framework, Sec. 2.3 for Intra-domain security and Sec. 4.1 for Virtual 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 Virtual 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
OBJ-2. Design and prototype a security and trust framework,	• *Provide mechanisms for zero touch trust automation in multi-domain scenarios on top of a 5G service management framework (KPI target: to cover*	See Sec. 2.1 for Trust management framework.

OBJECTIVE	Target KPIs	Applicable Design Artefact
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	*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).*	
	• *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 Intra-domain security at the business level.
	• *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 Trust 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
OBJ-4. Define solutions for secure, automated and intelligent resource	• *Automatically discover and "inventorize" various types of resources (i.e., compute, storage, network at core, edge, far-edge), spectrum and services*	n/a

OBJECTIVE	Target KPIs	Applicable Design Artefact
discovery, brokerage and selection, operation with SLA to facilitate workload offloading to 3rd-party resources supporting pervasive computing across multiple 5G domains.	*capabilities from different domains and service providers (KPI target: distribution of resource updates and discovery in less than 10 mins).*	
	• *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 Intra-domain security at the business level.
	• *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 Virtual resource Manager.
OBJ-6. Realize a cloud-friendly network software licensing	• *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

OBJECTIVE	Target KPIs	Applicable Design Artefact
framework for location independent network appliances execution.	• *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 January 2021.

[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 January 2021.

[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 January 2021.

[6] Secure Enclaves for Reactive Cloud Applications (SERACA) project. URL: https://www.serecaproject.eu/. Accessed 27 January 2021.

[7] SecureCloud. URL: https://www.securecloudproject.eu/. Accessed 27 January 2021.

[8] SCONE. URL: https://sconedocs.github.io/aboutScone/. Accessed 27 January 2021.

[9] 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 27 January 2021.

[10] 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)

[11] Common Vulnerabilities and Exposures. URL: http://cve.mitre.org. Accessed 27 January 2021.

[12] 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 January 2021.

[13] 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)

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

[15] Argo Project. https://argoproj.github.io. Accessed 27 January 2021.

[16] Kubernetes. https://kubernetes.io/docs/concepts/extend-kubernetes/operator/. Accessed 27 January 2021.

[17] 5G-MEDIA Serverless orchestration, infrastructure and VIM Driver implementation for OSM. https://github.com/5g-media/faas-vim-plugin. Accessed 27 January 2021.

[18] Etcd. https://etcd.io/. Accessed 27 January 2021.

[19] MySQL. https://www.mysql.com/products/community/. Accessed 27 January 2021.

[20] OptaPlanner. https://www.optaplanner.org/. Accessed 27 January 2021.

[21] Red Hat Operator Framework. https://www.redhat.com/en/blog/introducing-operator-framework-building-apps-kubernetes. Accessed 27 January 2021.

[22] Red Hat Open Cluster Management Kubernetes. https://github.com/open-cluster-management. Accessed 27 January 2021.

[23] Red Hat Advanced Cluster Management. https://www.redhat.com/en/technologies/management/advanced-cluster-management. Accessed 19 January 2021.

[24] 5GCity, Final 5GCity Orchestrator Release (D4.4), October 2019.

[25] Zeek network security monitor types. URL: https://docs.zeek.org/en/current/script-reference/types.html. Accessed 27 January 2021

[26] 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

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

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

[29] 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

[30] 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 27 January 2021

[31] 5GZORRO Consortium, Deliverable D2.2 – "Design of the 5GZORRO Platform for Security & Trust", November 2020

[32] TM Forum Open-API Schema Repository, "GeographicAddress". URL: https://github.com/tmforum-rand/schemas/blob/candidates/Common/GeographicAddress.schema.json. Accessed 13 January 2021

[33] 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)

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

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

[36] 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

[37] 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 January 2021

[38] 5G-Transformer, Deliverable D3.1 – "Definition of vertical service descriptors and SO NBI", Marc 2018

[39] 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

[40] Nextworks Slicer OpenApi specification v2-0. URL: https://github.com/nextworks-it/slicer/blob/master/API/sebastian-openapi-v2-0.yaml. Accessed 27 January 2021.

[41] 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.

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>