5GZORRO

Grant Agreement 871533

H2020 Call identifier: H2020-ICT-2019-2

Topic: ICT-20-2019-2020 - 5G Long Term Evolution

D2.2: Design of the 5GZORRO Platform for Security & Trust

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)

Intermediate version. Pending of EC revision. Do not cite.

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

Lead Beneficiary:	Document version:
NXW	**V1.0**

Work package:
WP2 – Use Case Definition, Requirements & Architecture

Deliverable title:
D2.2: Design of the 5GZORRO Platform for Security & Trust

Start date of the project:	Contractual delivery date:	Actual delivery date:
01/11/2019 **(duration 30 months)**	**01/11/2020**	**30/10/2020**

Editor(s)
G. Carrozzo, P. G. Giardina

List of Contributors

Participant	Short Name	Contributor
Nextworks	NXW	G. Carrozzo, P.G. Giardina, J. Brenes, E. Bucchianeri, G. Landi
Fundaciò i2CAT	I2CAT	C. Herranz, A. Fernandez, M. S. Siddiqui, Javier Fernandez
IBM Israel Science and Technology	IBM	K. Meth, K. Barabash
Telefonica Investigacion y Desarrollo	TID	Diego R. López
Ubiwhere	UW	P. Diogo, L. Conceicao
Fondazione Bruno Kessler	FBK	T. Subramanya
Universidad de Murcia	UMU	J.M. Jorquera Valero, P. M. Sánchez Sánchez, M. Gil Pérez, G. Martínez Pérez
Bartr Holding Limited	BTL	J. Taylor
Altice Labs	ALB	J. Bonnet, P. Chainho
Intracom	ICOM	M. Mertiri, T. Bozios, A. Lekidis, V. Theodorou
Atos Spain	ATOS	F. Bravo Díaz, A. Ramos
Malta Communications Authority	MCA	J.-M. Mifsud, A. Sciberreas

List of Reviewers

Participant	Short Name	Contributor
IBM Israel Science and Technology	IBM	K. Barabash
Altice Labs	ALB	J. Bonnet
Fundaciò i2CAT	I2CAT	M. S. Siddiqui

Change History

Version	Date	Partners	Description/Comments
0.1	31-07-2020	NXW, i2CAT, IBM, UMU	Integration of contributions by NXW, i2CAT, IBM, UMU
0.2	25-08-2020	FBK, ATOS, ICOM	Integration of contributions by ICOM, ATOS, FBK
0.3	07-09-2020	NXW, i2CAT, IBM, UMU, ALB, ICOM, ATOS	Integration of contributions by NXW, i2CAT, IBM, UMU, ALB, ICOM, ATOS
0.4	15-09-2020	UMU, UW, IBM, FBK, i2CAT, ATOS, ICOM, ALB	Integration of contributions by i2CAT, IBM, UMU, UW, FBK, ATOS, ICOM, ALB
0.5	22-09-2020	UW, TID, FBK, UMU, BTL, IBM, i2CAT, ATOS	Integration of contribution by UW, TID, FBK, UMU, BTL, IBM, i2CAT, ATOS
0.6	29-09-2020	UMU, BTL, IBM, i2CAT, ATOS, ALB, ICOM, MCA, FBK	Integration of contribution by MCA, FBK, ALB, UMU, BTL, IBM, i2CAT, ATOS, ICOM
0.7	05-10-2020	BTL, IBM, MCA, FBK, TID, ALB, ICOM	Integration of contribution by BTL, MCA, FBK, TID, ALB, ICOM Partial Review (up to Sec.4 included) by IBM
0.8	13-10-2020	IBM, BTL, i2CAT, UMU, ALB, ICOM, TID	Integration of contribution by IBM, BTL, i2CAT, UMU, ALB, ICOM, TID Full Review from IBM
0.9	15-10-2020	ALB, IBM, UW	Integration of contribution by ALB, IBM
1.0	30-10-2020	NXW, i2CAT	QA and final fixes before submission

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

In view of an architecture consolidation and of the full realization of the goals of pervasiveness and agility originally set for 5G networks, it is needed to work towards the definition of more disruptive approaches for 5G network configurations which can benefit of the integration of the latest technologies for resource and spectrum sharing, network orchestration, end-to-end security and trust.

Three novel design principles are emerging in the industrial and research communities working on 5G. First is Artificial Intelligence (AI), which can transform network management into a cognitive process through which the network can self-adapt and self-react to changing conditions with minimal manual intervention (zero-touch). Second, Distributed Ledger Technologies (DLT)/Blockchains (BC) can be adopted to implement distributed security and trust across the various parties involved in the 5G service chain. Third, Cloud Native technologies allow to achieve the necessary level of flexibility, scalability and resilience of SDN/NFV-based services for 5G. These three technologies, coupled with the advancement of the 5G specifications at 3GPP, can ensure the needed efficient delivery of cutting-edge 5G services.

Moreover, the quest for pervasiveness of 5G network services in an affordable way for Telcos who are called to build new infrastructures with very dense footprints (e.g. in cities, in industrial districts, in high aggregation areas like shopping districts, hospitals, etc.), calls for the overcome of bilateral Business-to-Business (B2B) models traditionally adopted by operators to implement the sharing of passive infrastructure elements and roaming agreements.

5GZORRO envisions a multi-party distributed model for building 5G Networks which can involve Telcos, spectrum owners, infrastructure owners, technology providers and Verticals, who can establish cross-domain service chains with security and trust.

This document presents the first version of the ***5GZORRO high-level architecture*** which is designed to implement the aforementioned vision.

The architecture follows a principle of service-based architecture similar to the 5G Service-based architecture defined in 3GPP and in for the ETSI zero-touch Network and Service Management. Integrating SDN/NFV and Cloud native orchestration technologies with a Permissioned Distributed Ledger infrastructure, the 5GZORRO architecture offers services for:

- cross-domain network slicing,
- resource and service offering via marketplaces,
- discovery, intelligent selection and trading of resources and Services via Smart Contracts
- zero-touch network slice and service lifecycle management
- cross-stakeholder e-license management
- SLA monitoring & breach prediction
- security and trust across multiple domains.

The realization of these services is made possible through the interaction of various functions for slice orchestration, network intelligence and analytics, security trust, management of service virtualized resources, all executed for multi-domain and single domain scope.

The architecture implements also the concept of sharing operational data across the whole system in a logically centralized data reservoir (a.k.a. Data Lake), so that multiple asynchronous management components may act upon this shared data pool towards optimizing a target set of KPIs. The ***5G Operational Data Lake*** component serves as a logically centralized reservoir of all the operational data, channelled by management services of Inter-domain Layer on behalf of domain specific management services running in every domain of the Single Domain Layer. It will provide APIs for adding, processing (in place) and retrieving data for analytical processes. A ***5G Permissioned Distributed Ledger*** component allows to implement Smart Contracts among the parties for 5G network services and slices, and ensures the aforementioned interoperability by providing data governance, multi-party trust, and accounting for data usage by different participating parties.

In this document, the services exposed by each core building block of the 5GZORRO high-level architecture are described, by identifying the envisaged functionality, its level of support (i.e. M-mandatory, O-Optional) and scope (i.e. cross-domain of intra-domain).

The related ***5GZORRO information elements*** are then specified, with focus on the resource offer information models (spectrum, RAN, edge/core cloud resources, VNF – Virtual Network Function / CNF – Cloud-native Functions, Network Slice).

Operational patterns for publishing resource offers, for trustworthy resource discovery, for smart contract setup with spectrum, edge/core resources and network slices are presented, together with initial design decisions related to the ***5GZORRO software platform***

The reference architecture presented in this document fulfils the requirements identified for three 5GZORRO use cases described in Deliverable D2.1. It also leverages on a large base of state-of-the-art technologies and standards for virtualization, NFV, Cloud Native platforms and services, zero touch, SDN, distributed ledgers, data lakes, which have been extensively reviewed to summarise the specific positioning of the 5GZORRO innovative proposition.

The 5GZORRO reference architecture presented in this document is the base input for the low-level design and implementation work planned on the 5GZORRO software platform. An update of this specification is planned for after the completion of the first implementation cycle in order to include refinements and detailed interface specifications for the various services which incorporate decisions and feedback from implementation.

1 Introduction

Despite enormous progress achieved during recent years, current 5G deployments are still far from reaching the level of maturity needed to address all the requirements from the vertical industries. A partial support to the network slicing and network monitoring, with limited possibility to establish a cross-operator end-to-end slice, and a very limited support of the ITU-T 5G vertical applications other than eMBB (e.g. URLLC and mMTC) represent a significant barrier to the achievement of full 5G potential. One main cause resides in the way the network, its actors (mainly telco operators) and the business relationships among them have been considered so far. In fact, current business relationships between Telcos, infrastructure owners and regulators are based on bilateral agreements, intended mainly for enabling roaming and sharing of passive infrastructure elements, which practically represent the only negotiated resource. Further, such kind of agreements are often negotiated offline between the involved parties and take a long time to be defined and finally deployed in the network environment. This results into a general difficulty and lack of efficiency in establishing end-to-end services based on resources other than those in passive infrastructure, across different providers, with a subsequent limited capacity of the current 5G operators to become truly pervasive with their service offering.

The current context of deployment and operations of 5G networks in Europe and worldwide suffers the lack of frameworks for the 5G stakeholders to provide services and tools necessary to:

- Evolve the concept of **shared network resources,** from passive infrastructure to a set of heterogeneous resources that includes computing, transport, storage, radio and spectrum. This consequently leads to a differentiation of 5G network stakeholders which will include not only traditional operators but also very specialized (virtual and/or physical) resource providers: fiber, radio, spectrum or even slice/service owners. Resources from 3rd parties (e.g. edge computing at street cabinets) should be also taken into account, in order to improve the overall edge computing capacity and achieve a truly ubiquitous network;
- Enhance the way the **different parties conduct business** with each other, from long and offline to fast, trusted, secure and, above all, automatic (zero-touch) online negotiations and agreements;
- Establish a new common view on the **different network stakeholders**, from a set of parties that could, occasionally, be in business together, to a set of providers able to offer heterogeneous resources that can be traded and used to establish multi-party services/slices.

The 5GZORRO architecture described in this document aims to specify such a framework, characterized by services and tools enabling the network stakeholders to select and trade the resources, slices and services from different providers and to establish business relationships exploiting automatic and secure procedures.

1.1 Document outline

This document is structured as follows:

- Section 2 describes the benefits and solutions 5GZORRO aims to introduce and the technologies enabling these solutions.
- Section 3 details the services offered by the 5GZORRO architecture supporting the benefits mentioned above.
- Section 4 describes the architectures and technologies on top of which the 5GZORRO concept will be built.

- Section 5 states the principles used to build the architecture together with the identified functional and the main services offered by each one of them to the rest of the architecture.
- Section 6 illustrates how the functional elements interact in the most relevant operation patterns of the architecture.
- Section 7 establishes the software design principles and the initial software design.
- Section 8 concludes the document providing a table where the project KPIs are mapped to the sections of this document in which the handling of the specific KPI is covered.

The 5GZORRO architecture builds on a significant set of enabling technologies (e.g. for DLT, data lake, virtualization, cloud native) and standards which have been analysed and reviewed to elaborate a proper positioning of the 5GZORRO innovations. For sake of compactness of the document, this extensive review of state of the art has been included in Appendix I of this document, and structured in four main areas: Intelligent zero-touch management (see section 11.1), Cross-domain resource & service trading (see section 11.2), Security & Trust (see section 11.3) and Technology enablers (see section 11.4).

2 5GZORRO Concept

5GZORRO incorporates solutions based on three novel concepts:

i) Data-driven and AI-based solutions which can enable automatic and autonomous network operations following AIOps paradigm;
ii) Distributed Ledger Technologies (DLT) which enable trust and security in multi-party end-to-end service/slice implementation
iii) Cloud-Native technologies which once integrated into SDN/NFV environments can increase the level of flexibility required by advanced 5G based services (e.g. scalability, resilience).

The combination of these three concepts is the basis for realization of the three main 5GZORRO innovations, briefly summarized in the following.

Zero-touch/Automated Resource discovery using DLT/Blockchains. The automatic (zero-touch) resource discovery is based on the extensive use of AIOps and DLT solutions. The main goal is, from the one hand, to allow different stakeholders to publish their own resource/service offerings and, from the other hand, to enable the business logic to automatically discover the most suitable set of resources while minimizing the human intervention. For the resource/service trading, 5GZORRO offers a set of modules that build a proper Marketplace Application (described in Section 7.5), where the business agent can discover and classify the available resources and services. Each resource/service offering published into the Marketplace, is stored into the blockchain and becomes immutable, facilitating the process of discovery and classification and making it secure from a business point of view. The discovery and classification process can be hence completely automatized (zero-touch), directly affecting the way the various parties establish business relationships: offerings are clear, immutable and need no human interaction and/or offline negotiations. Further, the concept of Marketplace enlarges the set of network resources and extends it to abstractions like services and slices, opening the door to a new generation of network stakeholders beyond classical Telcos.

Intelligent 3rd party resource selection, request and access/usage. Once resources have been made available on the DLT-based resource catalogue and automatically discovered and classified, an automatic AI-based process can select the most suitable ones, request them from the owners and, after the business transaction has been fixed into the DLT, finally use them. The decision process is driven by analysing the historical information stored into the operational Data Lake, like costs and KPIs. Static rules can be set manually by the potential resource consumer that can act as a pre-filter, reducing the set of resources the AI-based agents can use for selection. The transactions are stored in the form of Smart Contracts, legally binding, automatically generated by the Platform when resources/services are requested for deployment,

that also happen automatically. This last aspect involves also the lifecycle management of the resources/services, not only the deployment phase, but also of the configuration and the optimization of the service based on the resource selected whose conditions are fixed into the smart contract. In particular, the Intelligent 3rd Party resource selection heavily applies the zero-touch management paradigm that guarantees that different resources/services offered by different providers (administrative domains) can be seamlessly composed (service creation/service stitching) across the different domains. AI-based mechanisms apply the correct configuration of the services/resources while guaranteeing that the Service Level Agreement is properly applied in all of the parts of the service chain belonging to all the different domains. Special SLA monitoring mechanisms are implemented and will react in case of SLA breaching. The application of the zero-touch paradigm dramatically reduces the time of resource/service negotiation between the involved parties: everything happens in an automatic way, from the selection of the resources, to the deployment of the services, passing through all the business and legal aspects.

Trust establishment among multiple parties. In view of enabling the automatic establishment of business relationships, 5GZORRO offers a mechanism that guarantees the trust and the security among the parties involved, with end-to-end security for the deployed services. Each stakeholder that wants to deploy a slice/service needs to be sure that all the resources/services provided by the 5GZORRO framework are secure and provided by trusted sources. The level of security and trust of each party is established in the smart contracts between the parties.

In Figure 2-1 it is depicted how different business parties can take advantage of the 5GZORRO framework innovations to offer different resources and to establish multi-party end-to-end services.

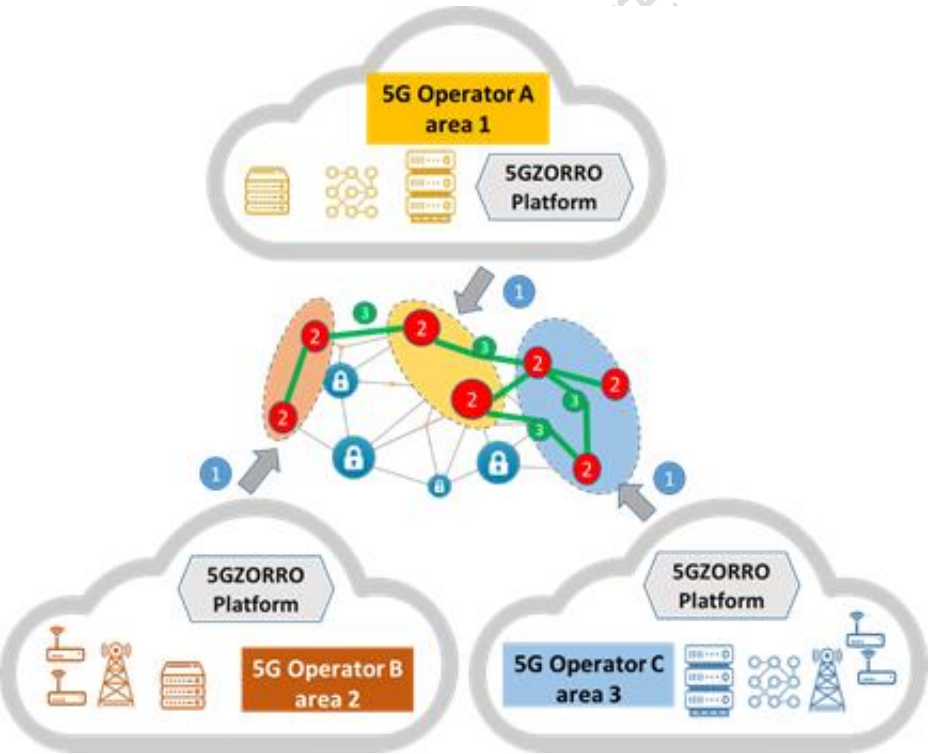

Figure 2-1: zero-touch/Automated Resource discovery (1), Intelligent 3rd party resource selection, request and access/usage (2) and Trust establishment among multiple parties (3) in 5GZORRO.

As first step (see spot 1 in Figure 2-1), Operators use 5GZORRO DLT-Based marketplace to publish and to check for new resources (*zero-touch/Automated Resource discovery using DLT/BC*). In order to build a cross-operator service, the framework intelligence automatically selects proper resources (see spot 2 - *Intelligent 3rd party resource selection*) whose usage and chaining is automatically formalized through the mechanism of the Smart Contracts (see spot 3 - *Trust establishment among multiple parties*).

A more detailed discussion can be found in D2.1 [1] where all of the 3 innovation concepts are planned and validated in representative use cases.

3 5GZORRO Services

The core 5GZORRO services are introduced in this section detailing principles and core concepts for

- cross-domain network slicing,
- resource and service offering via marketplaces,
- discovery, intelligent selection and trading of resources and Services via Smart Contracts,
- zero-touch network slice and service lifecycle management,
- cross-stakeholder e-license management,
- SLA monitoring & breach prediction,
- security and trust across multiple domains.

3.1 Cross-domain network slicing

3.1.1 Concept of multi domain slice and relevant scenarios

In the 5G vision, a high degree of reutilization of the infrastructure resources in order to overcome the elevated costs of investment and operation of the computing, transport, and radio resources. In a similar manner, it is envisioned that end-to-end services will traverse multiple geographical areas and will have varying demands in terms of edge resources, which are by definition scarce and hence it is unlikely that one single network operator will have enough resources to satisfy all the demand. Therefore, 5G network services and management platforms shall support mechanisms to enable flexible and on-demand resource sharing across multiple administrative domains and across different segments of the network.

The 5GZORRO architecture specifically aims to offer a set of interfaces to enable the transparent deployment of services relying on resources belonging to different network operators, infrastructure providers, etc. For this we leverage on the network slicing concept introduced in 5G, where a Network Slice (NS) represents the complete logical network, offering specific services over a computing, network and storage infrastructure. A Network Slice Instance (NSI) in this context is the realization of a network slice targeting the specific service constraints. Figure 3-1 illustrates the 3GPP [2] concept on how NSIs can be used to support the end-to-end communication services and how NSIs are further divided into network slice subnet instances (NSSIs), which are shareable logical (sub)networks providing a specific functionality and the corresponding resources.

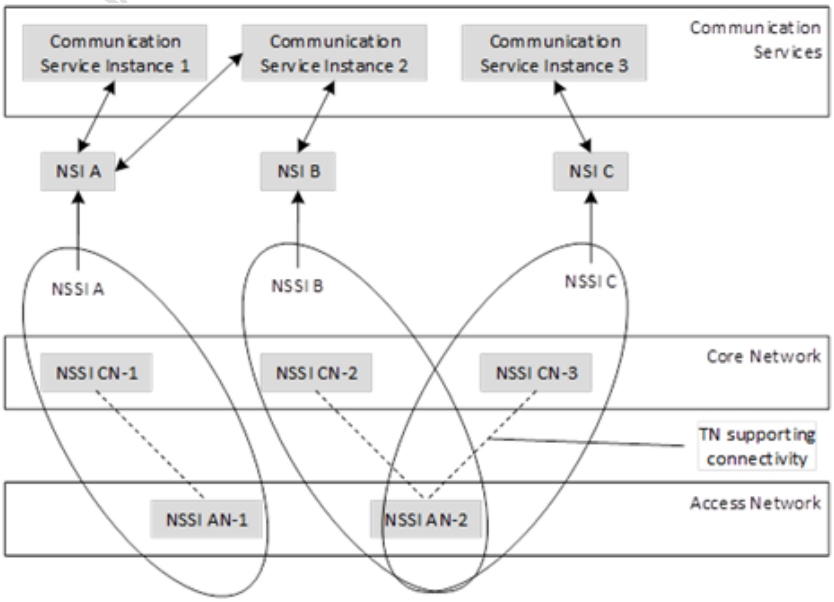

Figure 3-1: 3GPP concept of Network Slice Instance (source 3GPP) [2].

Figure 3-2 illustrates how an NS concept is mapped to the multi domain environment of 5GZORRO using a resource-oriented view of the most representative service deployment scenarios of the project.

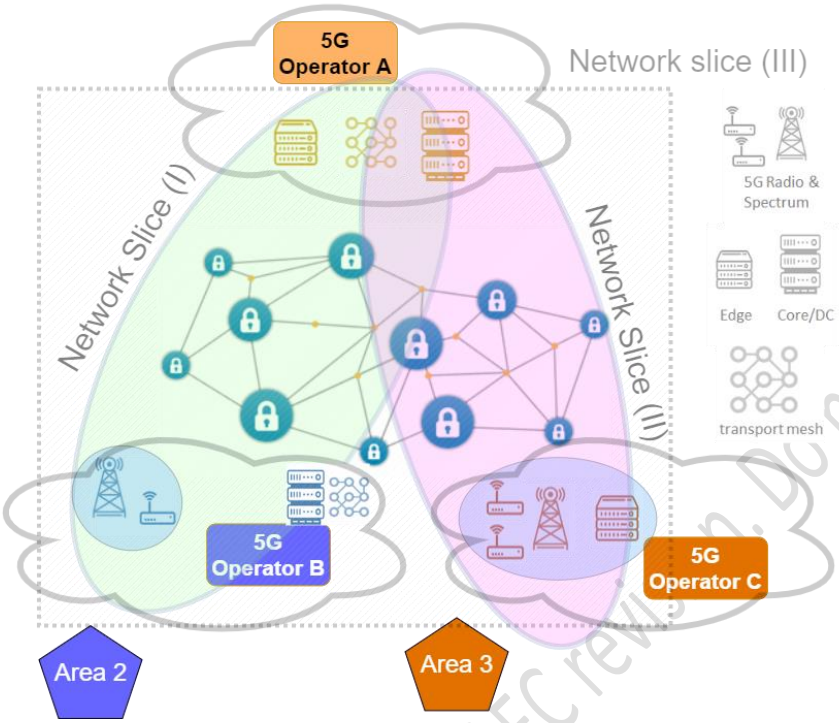

Figure 3-2: 5GZORRO Network slicing scenarios.

The three scenarios depicted in the picture are:

- '*multi domain RAN*': as represented in Network Slice (I) the service is deployed using slice relying on core, edge and RAN resources from operator 'A' and RAN and spectrum resources from operator 'B', probably due to the coverage constraints of the service. Since operator 'A' does not have the means to offer the service in 'Area 2', the RAN and spectrum resources shall be acquired using the 5GZORRO Marketplace. It is important to highlight that the RAN and spectrum resources could even be provided by different parties;
- '*multi domain RAN and edge*': as in the previous case the core functionalities and resources are provided by operator 'A', but in this case the edge, RAN and spectrum resources are provided by operator 'C'. This is the typical scenario of industrial facilities, where it is foreseeable private deployments of 5G RAN and edge infrastructure will appear in the near future to accommodate latency sensitive service on the edge;
- '*dynamic edge and RAN allocation*': in this scenario the network slice is dynamically scaled to accommodate new RAN, edge and spectrum resources due to a change in the service constraints. In this scenario, the slice can start as in Network Slice (I) and be scaled dynamically to embrace the resources used in Network Slice (II) (as represented in Network Slice (III)). This scenario represents the case where the terminals, using latency sensitive services, are expected to move across different areas.

From perspective of the logical functionality, the network slices will leverage also other types of 3rd party resources available on the market (i.e. VNFs)

It is clear that in all the scenarios, the requirements and constraints for the different segments (e.g. Cloud/Edge, RAN, Transport, etc) could be different. Similarly, the different sections will offer different kinds of information that the ML/AI algorithms could use to derive the orchestration decisions. In the following subsections we detail the specifics of each segment.

3.1.2 Generic Network Slice template & abstract parameters

A Generic Network Slice Template (GST) defined by GSMA in [3] is a set of attributes that can characterise a type of network slice/service. A GST where attributes are filled with desired values of a Network Slice is called Network Slice Template (NEST). The NEST is used for instantiating the Network Slice
Instance (NSI) and one or more NSIs can be created from the same NEST. The Table 3-1 describes a subset of relevant attributes of GST.

Table 3-1: Relevant attributes of a Generic Network Slice Template (GST).

NAME	DESCRIPTION
Area of service	Specifies the area where the terminals can access a particular network slice
Delay tolerance	Describes service delivery flexibility, if supported
Downlink throughput per network slice (guaranteed)	Describes the guaranteed data rate supported by the network slice in downlink
Downlink throughput per network slice (maximum)	Defines the maximum data rate supported by the network slice for all UEs together in downlink
Downlink throughput per UE (maximum)	Describes the maximum data rate supported by the network slice per UE in downlink, it could be used to offer different contract qualities
Isolation level	Describes different types of isolation
Maximum supported packet size	Describes the maximum packet size supported by the network slice and may be important for URLLC (Ultra-Reliable Low Latency Communication) and MIoT (Massive IoT), or to indicate a supported maximum transmission unit (MTU)
Radio spectrum	Defines the radio spectrum supported by the network slice. This is important information, as some terminals might be restricted in terms of frequencies to be used
Slice quality of service parameters	Defines all the QoS relevant parameters supported by the network slice. For some of these parameters 3GPP has already defined standard values in [4]
Supported device velocity	Defines the maximum speed supported by the network slice
UE density	Describes the maximum number of connected and/or accessible devices per unit area (per km2) supported by the network slice
Uplink throughput per network slice (guaranteed)	Describes the guaranteed data rate supported by the network slice in uplink (and not per user)
Uplink throughput per network slice (maximum)	Describes the maximum data rate supported by the network slice in uplink (and not per user)
Uplink throughput per UE (maximum)	The maximum data rate supported by the network slice per UE in uplink, it could be used to offer different contract qualities

According to 3GPP TS 28.541 [5], a Network Slice has a Slice/Service Type (SST) field to describe the expected network behaviour. There are three standardised values for SST described in Table 3-2. Each SST is defined by a set of additional parameters with standard values, listed in [5].

Table 3-2: Standardised SST (Slice/Service Type) values.

NAME	SST value	Description
eMBB	1	Slice suitable for the handling of 5G enhanced Mobile Broadband.
URLLC	2	Slice suitable for the handling of ultra- reliable low latency communications.
MIoT	3	Slice suitable for the handling of massive IoT.

The characteristics in network slice described by 3GPP using SST has reference in GSMA GST. Standardised SST values refer to Network Slice characteristics defined by GST attributes populated with standardised values.

3.1.3 Principles of mapping of network slices in RAN

At the time of instantiating a network slice, a set of computational and infrastructure resources are allocated in order to provide the agreed end-to-end service requirements. When the network slice involves a wireless or cellular network deployment, spectrum and radio infrastructure resources must also be allocated, thus creating a 'RAN slice'.

Typically, the end-to-end service does not define the radio access requirements and, consequently, there is a need to translate the service requirements to RAN resources. This translation logic is implemented at the RAN Controller in the 5GZORRO architecture. The RAN Controller takes some of the end-to-end service requirements of a network slice as input to create a 'RAN slice.' The input parameters include delay tolerance, maximum (guaranteed) throughput, UE density or number of simultaneous connections, type of service, available spectrum, and the area of service. With this information, the RAN Controller will deploy the necessary RAN resources in the geographical area of application of the service. The RAN resources devoted to the resulting 'RAN slice' is composed of a set of Access Points (AP) or cellular base stations and, for each of them, its radio access technology (e.g. Wi-Fi, LTE, 5G NR), a set of operation bands or channels, central operation frequencies, and operation bandwidths.

On top of this, the 'RAN slice' also determines the amount of total spectrum resources of each access point or cell allocated to a service. In particular, the global scheduler implementation in the 5GZORRO RAN Controller enforces that each slice obtains the necessary amount of spectrum resources to meet the service requirements. The RAN Controller enables the possibility to isolate slices, meaning that idle spectrum resources are not shared among slices. Once the global scheduler allocates spectrum resource to each slice, users within a slice are served following any of the wide-spread scheduling algorithms in the Wi-Fi APs or cellular base stations. The RAN slicing configuration is flexible so it can accommodate the addition or removal of slices or modify the amount of spectrum resources allocated to an active RAN slice.

The RAN Controller in the 5GZORRO may push to or collect data from the data lake in order to generate statistics, which can be used by a ML/AI model to determine the optimal radio configuration for a given type of service. For instance, the intelligence in the RAN Controller would favour the creation of a RAN slice with 5G resources for a service with a stringent low latency requirement with a given bandwidth based on the service requirements and statistics in the data lake.

3.1.4 Principles of mapping of Network slices in edge/core

The inclusion of edge/core infrastructure resources in network slices is a fundamental aspect in the creation of logical networks on top of common physical 5G infrastructures. Since a network slice instance is by nature purpose-built, business requirements related to edge/core resources need to be mapped into slice ingredients. In particular, compute, storage and/or hardware-based (e.g. GPU, TEE) capabilities as well as affinity rules, based for instance on geographical location or trust, are potential inputs to be taken into account for the setup of network slices. Moreover, core properties of network slicing such as guaranteed performance, isolation and reliability must be also ensured.

For network slices containing edge resources, a first level of slice mapping decisions could regard the selection of the appropriate Edge Point of Presence (PoP) on which to instantiate each slice subnet. Decision criteria could regard hard constrains, such as geographical coverage, end-users' location, infrastructure resources availability; as well as non-functional requirements about latency, throughput; and even reputation-based criteria.

In order to meet requested criteria and accommodate virtualized edge/core resources as part of network slices, translation rules need to be implemented as part of the slice provisioning and adaptation mechanisms. To do so, a common approach is to rely on a pure imperative scheme by specifically stating what edge/core resource chunks a slice needs to include. The use of monitoring data analytics coupled with ML/AI techniques

can be exploited to perform a more intelligent slice provisioning by stating what a slice needs to support and/or how a slice needs to perform. Likewise, by leveraging smart prediction mechanisms, potential problems can be anticipated and proactive actions can be taken accordingly to avoid reliability or performance degradation during the slice operation. In the scope of 5GZORRO, the combination of both schemes is foreseen, which allows not only to achieve a higher degree of customization and elasticity during the entire slice lifecycle but also to optimize the use of shared infrastructure resources.

3.2 Offer and resource catalogues

One of the key services offered by the 5GZORRO architecture is the marketplace service which enables the trading of resources. This will allow resource providers (i.e. infrastructure providers, network operators, NS providers, VNF providers, etc) to publish the resources they aim to trade along with the particular business and pricing conditions associated with the offer. Resource consumers (i.e. vertical companies), on the other hand, shall be able to query the market to acquire the resources needed by the service they aim to instantiate or modify.
In 5GZORRO, the approach used in [29] is used, where a resource market is mainly supported by three different catalogues: (i) the resource catalogue; (ii) the service catalogue (iii) the offer catalogue. The resource catalogue holds the inventory of the 5GZORRO available resources (VNF/CNF packages, NSDs, RAN elements, Spectrum resources).

In general terms, a resource entry shall contain: the type of resource, a reference to the specific resource definition, relationships with 3rd parties (i.e. owner, provider, etc) and a set of specification characteristics and configurable parameters. Due to the decentralized nature of 5GZORRO, the reference to the specific resource definitions, and all the catalogue entries shall contain a decentralized identifier (DID), as detailed in Section 11.4.2.

Resources are usually associated with service entries that detail how a resource can be consumed, instantiated or deployed. These service entries contain therefore: the set of resources associated, relationships with other service specifications, a set of service characteristics and agreements established for the service. In order to be available for use by other parties, services and resources need to be mapped into product offers. In broad terms, product offers contain a reference to the related services and resources, the pricing options and the agreements established for the offer. In 5GZORRO, the offers shall be also supported by smart contracts which ensure the cohesion and security across the multiple domains. 5GZORRO also relies on the DLT and smart contract capabilities to perform all the currency procedures. The 5GZORRO architecture is therefore agnostic to the specifics of the money transfers and billing mechanisms.

Service providers will be able to specify new services based on the services and resources available as product offers. When a new service is to be instantiated, the 5GZORRO Marketplace user will need to acquire the related resources, referring to the product offers available. The following sub-sections detail the specific offer and resource specification considerations for the most important assets within the project, and the high-level workflow to publish a new a resource offer.

For each type of offer specified in the following subsections, the resource provider – being it spectrum resource provider or edge/cloud infrastructure provider or RAN provider, etc) should create an offer containing:

- *Agreements*: the agreement between the parties, e.g. to exploit the licensed spectrum (auction), a pricing agreement between the resource provider and the resource consumer, etc. For example, the license/unlicensed spectrum bands, the set of base stations and their location, an agreement on using certain backhaul connections, an agreement on leasing baseband capacity for the antennas, an agreement on the percentage of resource to guarantee, etc.
- *Pricing models*: These models rely on input parameters (such as the SLA, the number of spectrum resources, etc), and determine the price using the 5GZORRO currency.
- *Business terms*: These are the terms for which the offer is valid. For example, the region of applicability, the specific frequency resources, the prohibition of reselling the spectrum, an

agreement not to exceed a determined transmission power, region of applicability, leasing time, expected total uptime, etc.

3.2.1 Spectrum offers

A Spectrum Resource Provider (SRP) in the 5GZORRO Marketplace can freely share their spare spectrum resources. Then, spectrum resource consumers can acquire the resources available in the Catalogue.

Before placing a spectrum offer in 5GZORRO, the SRP must get the recognition from the National Regulator as the legitimate owner of the claimed spectrum. Then, the Regulator will issue a spectoken for the SRP, which will be included in the spectrum offer. The owner of the spectoken is recognised in the 5GZORRO architecture as the sole operator of the frequency range defined in the spectoken.

In order to publish spectrum offers in the 5GZORRO Marketplace, the SRP will provide the spectrum details and the spectoken generated by the National Regulator. Both the offer and the spectoken refer to the same range of frequencies and geographical area.

3.2.2 RAN elements (active & passive) offers

Mobile Network Operators (MNO) and RAN infrastructure providers may share their RAN assets with resource consumers within the 5GZORRO architecture. To this aim, the RAN infrastructure providers put RAN infrastructure offers in the 5GZORRO Marketplace. The RAN resource consumers can acquire RAN elements by selecting the most appropriate offers in the 5GZORRO Catalogue.

3.2.3 Edge/Core Cloud resources (IaaS, PaaS) offers

Edge/core cloud infrastructure providers may offer their cloud or edge computing assets, such as CPU/GPU servers, storage, and networking, to resource consumers connected to the 5GZORRO Marketplace. To this aim, the edge/cloud infrastructure providers publish the edge/core cloud resource offers in the 5GZORRO Marketplace. The edge/core cloud resource consumer can browse the 5GZORRO catalogue and choose the most suitable offer for deploying their services.

3.2.4 VNF/CNF offers

Software vendors will be able to trade their VNFs/CNFs using the 5GZORRO Marketplace. Other software vendors will be able to use the VNF/CNFs in the Marketplace as part of their service, while end-users will be able to acquire the required VNF/CNF licenses based on the offers available.

In order to publish a VNF/CNF in the 5GZORRO Marketplace, Software Vendors will provide an identifier pointing to the VNF package or the CNF image. A resource entry will be created to register this asset containing: identifier, package/image format software version, package signature, configuration parameters, etc. A service entry will be created to detail how this VNF/CNF can be instantiated, identifying supported platforms, access points, etc.

3.2.4.1 Management of descriptors across multiple-domains

VNF vendors may offer their software products in the 5GZORRO marketplace, allowing other providers to compose services or slices with software products from diverse vendors, trading them in the marketplace. Resources and services will be published in the marketplace with a decentralized identifier (DID) associated, in order to unequivocally and globally identify the asset; this approach is discussed further in Section 3.3.1 .

The resource offer is composed by the business and legal offer definition, reflected in the smart contract and published in the DLT, but also by data content like descriptors, VIM endpoints, or VM/container images.

Sizeable binary data such as resource offer technical specifications will not be stored on the DLT as it may have scaling implications depending on the DLT implementation. Instead, this data will be stored 'off-chain' in traditional storage mechanisms e.g. database, a common optimisation technique used in conjunction with distributed ledgers. A minimal set of data attributes, including the DID of the offer and a cryptographic hash of the offer contents will be committed to the DLT. This will ensure that the amount of data stored on the ledger remains relatively small, and services are able to locate full offer meta-data from off-chain sources. Taking this approach will ensure that the DLT remains performant and the hash of the offer contents will

support non-repudiation of the offer contents itself; this will need to be updated each time an offer is updated in order to remain in sync.

Once the resource agreement is completed, the purchaser 5GZORRO platform will request the metadata associated to the offer through the universal resolution system. The resolution system will translate the DID to resolve the offer's metadata and its location in the whole ecosystem, making this data available for the resource/service roll out or modification.

3.2.5 Network Slice and Network Service Offers

Network Service (NS) providers will offer their Services through the 5GZORRO platform. Services themselves are defined by a composition of several embedded resources, potentially originated from multiple providers, that are required for their instantiation and execution, such as spectrum, software (one or multiple VNFs/CNFs), and hardware specifications.

To be able to publish a NS in the 5GZORRO Marketplace, the provider operator builds the service definition comprising the necessary resources resulting in the creation of a new smart contract that includes the service offered, the terms of service and pricing. After publishing of the document, the system verifies all aspects that pertain the validity of the service and, if all is correct, a smart contract is published in the 5GZORRO Marketplace rendering the Service available at the marketplace for consumption.

Network Slice providers will also be able to offer "slices" using the 5GZORRO platform. Network slices are composed by a set of resources, such as spectrum, computing and network resources. Although not enforced, Network Slice offers should be created alongside Network Service offers to accommodate their deployment. In other words, a Network Slice Instantiation (NSI) is typically devoted to accommodate a particular Network Service instantiation. Furthermore, given that the Network Service resources, as well as its requirements, are specified, the creation and publishing of a Network Slice Offer in 5GZORRO Marketplace is a relatively straightforward procedure.

3.3 Discovery, intelligent selection and trading (5GZORRO Marketplace)

3.3.1 Resource and Service discovery

The 5GZORRO system will serve as a decentralized marketplace whereby participants can freely trade resources and services without the need for a trusted intermediary. Each participant will host a distributed application (DApp) that interfaces with their domain's DLT node, forming a peer-to-peer (P2P) network consortium.

Providers will register resources and services via the marketplace using a standardised data format or schema in order to form a catalogue of items available to consumers. These resources and services can be seen as tokenised digital assets stored on the DLT, forming an immutable record of ownership and availability on the ledger that is tracked and updated over time in accordance with an associated Smart Contract that governs any state change; e.g. metadata update, or when leased to a particular consumer. By way of optimisation, metadata associated with an asset will be stored off-chain with just essential properties and a cryptographic hash of the resource definition being recorded on the ledger.

Resources & services will be universally identifiable and discoverable thanks to the employment of Distributed Identifiers (DIDs) and Verifiable Claims. A DID can identify any subject such as a person, organisation, or thing. DLT is a key enabler for DIDs, which removes the need for a centralized registry or authority and empowers the controller of the DID to prove control over it without permission from a third party. The process of registering a resource or service will involve generating a DID and associated DID document that describes associated cryptographic material, verification methods and service endpoints, allowing the owner (controller) to prove control over the resource and present any associated claims, and third parties to discover endpoints. This DID will be referenced in the tokenised state stored on the DLT and catalogue associated with the resource or service meaning that any participant can utilise the DID to query associated metadata.

When advertising a resource or a service, a provider will also associate legal prose with its definition. Prose will be generated from templates that are subject to a governance process, defining the legal framework and SLAs for the resource; Smart Contracts, derived from approved legal prose templates will facilitate the autonomous validation of actions made against the digital asset and management of the contract lifecycle in-line with the terms defined within.

The advertisement of resources or services by providers will be realised via automated mechanisms that will involve interfacing with their domain's management and/or orchestration entities. NFV MANO frameworks (e.g. OSM), edge orchestration platforms (such as Kubernetes lightweight distributions) or plain resource management entities will interwork with 5GZORRO platform in order to announce, register and update offers about available resources and services to the marketplace.

3.3.2 Intelligent 3rd party resource selection

The 5GZORRO architecture aims to facilitate multi-party collaboration in dynamic 5G environments where Operators and Service Providers often need to employ 3rd party resources to satisfy a contract. In 5GZORRO, there are several stages in the process of obtaining access to 3rd party resources.

As described above in Section 3.3.1, resource providers make their resources available for share by advertising them in 5GZORRO marketplace. The Cross-domain Monitoring and Analytics component keeps track of the resources available for sharing continuously over time, collecting the information that Smart Resource Discovery component can rely upon. Resource Consumers then use Smart Resource Discovery component to obtain the list of resources available and suitable to satisfy their need and to decide what 3rd party resources are the most appropriate to use. Upon making the choice, resource consumer initiates Smart Contract creation as described below in Section 3.3.3. When the agreement is sealed, 5GZORRO marketplace is updated about the new stratus of resources and offers catalogues.

For making decisions about what resources are the most appropriate to use in each particular case, Smart Resource Discovery component can rely on different data sources and algorithms: static considerations, e.g. performance and QoS characteristics, cost, historical or business preference for connecting to a certain provider, etc.; and dynamic considerations, e.g. current and predicted load and performance of relevant systems, components and services, topological network proximity of available resources to satisfy latency constraints, etc. While static considerations can be provided by the resource consumer as part of resource lookup request, dynamic considerations can only be computed using data collected at runtime. To achieve the latter, Smart Resource Discovery component will use streamed and historical data of 5GZORRO Operational Data-lake and Machine Learning algorithms to create, train, and validate models for each specific resource usage case. For example, network topology awareness can be infused into Smart Resource Discovery component by correlating the reported service QoS with monitored network characteristics such as technology, provider, proximity, hop-to-hop latency, etc., over historical data across many similar past resource usage contracts.

Of course, it must be noted that making data-based decisions in production business environments requires that all the data taken into consideration is trustworthy at all stages – at collection, transmission, transformations, computation, etc., putting strict security requirements on 5GZORRO AIOps platform. This ties in with architectural decisions already made by 5GZORRO team: incorporate Smart Contracts, DLT, and Trusted Execution Environments (TEEs) to ensure multi-party trust to ensure the data is verifiably authentic, traceable to its source, and not tampered with while in transfer or in processing.

3.3.3 Resource and Service trading via Smart Contracts

Creating a commercial trading agreement between provider and consumer autonomously will be facilitated through smart contracts. Smart contracts ensure that an agreement and any associated actions on that agreement are processed in accordance with the agreed terms by validating any transition of ledger state. What this means is that on entering into an agreement, whereby each party agrees terms and signs the transaction, from that point on there is a commercial agreement between the two legally identifiable entities backed by a legally enforceable contract (Ricardian Contract [6]).

Smart Contract templates will be developed to capture both the broader general terms of an agreement, and operational terms relating to a Service Level Objective (SLO), with specialized templates to serve the needs of each resource type to be traded as necessary. These templates will consist of parametrised legal prose to be utilised by stakeholders, crucially encapsulating real-world legally ratified contracts. Smart contract templates will give rise to legally enforceable smart contracts, but also the compelling improvement over existing working practices by standardising contract terms across all stakeholders.

Resource and service business meta-data will comprise concrete instantiations of these templates, producing a hierarchy of terms that outline the legal terms of the agreement, SLAs and their associated SLOs.

These agreements will be deployed and managed by a component that manages the lifecycle events of the contract. Smart Contracts will mirror that of the real-world contract and encapsulate logic to automate the calculation of SLA compliance. On deployment of the contract to the ledger, the autonomous set-up of monitoring and configuration of aggregation algorithms will be initiated by the Smart Contract Lifecycle Manager. During the course of the contract's lifetime, metrics can be posted to the smart contract by the monitoring aggregation service and at frequencies as agreed in the contract. Should a breach occur, the Smart Contract will enact any subsequent events, which might simply be to record the breach until such time that a threshold is reached or trigger the termination of the contract.

Smart contracts are ultimately providing autonomous near real-time execution of contract lifecycle stages, from creation, monitoring & SLA enforcement through to settlement, disbursement and finally termination.

3.4 Zero-touch lifecycle management for network slices and network services

This sub-section describes how 5GZORRO will deal with the zero-touch concept for life-cycle management of the Network Slices and/or Network Service, especially when cross-operator aspects of it are addressed.

3.4.1 Cross-Domain Network Slice Lifecycle Management

Achieving zero-touch Network Slice life-cycle management is in itself particularly challenging, even when just within the domain of a single operator. But in 5GZORRO we believe we can handle this challenge if we adopt some tools and processes, as described next.

The context for the following text is:

> *we want a Network Slice to be instantiated and managed (i.e., expanded, re-configured, etc.), cross domain (e.g., expand only its radio part, or its edge computing, only 5G connectivity, etc.) and even cross-operator.*

As illustrated in Figure 3-2, multiple types of resources of a slice (core, edge, RAN & spectrum) might be subject to these lifecycle operations, with the added complexity that these resources might belong and/or be managed by more than one operator. In this scenario, specific interactions between the owners/managers of those resources must take place. Given the above context, an effective Network Slice definition will be comprised of the above referenced fields (see sub-section 3.1.2), plus more fields (to be defined later on the design of the concrete solution) allowing the verification of each one of the resources' owner.

5GZORRO will use a DID/DLT mechanism to uniquely identify these different resources, even when they are described in a distributed manner (see sub-section 3.2, above). These descriptors will be extended with the interconnection data mentioned above, thus allowing for a far more extensive scope of a Slice Descriptor, e.g. to include RAN, spectrum and other relevant resource components of the concrete slicing implementation.
Please note that those service provider-specific credentials should cover the whole Network Slice lifecycle. If a service provider does not have the capability to expose certain parts of the Network Slice lifecycle, this must be known right at the Network Slice definition phase, since it limits further actions on subsequent phases.

3.4.1.1 Network Slice instantiation

With a clear and complete Network Slice Descriptor, its instantiation should be fairly simple: it implies negotiating with every Network Service provider that provides each of the Network Services that are part of the slice, the instantiation of that service and its connection to the other services.

3.4.1.2 Network Slice expansion/reduction

Expanding/reducing a slice can mean different things, some of them simultaneously within a single use case, which we address below:

- Expanding/reducing the capability of each one of the Network Services comprising the slice: in this scenario, more/less instances of the same Network Service might be added/reduced, depending on more or less complex licensing schemas (see below, the sub-section about licensing), and inter-connected/disconnected. This can be seen as a scaling-out scenario of the Network Services. An alternative to this practice would be to maintain the number of (interconnected) Network Services but scale-out (expand) or scale-in (reduce) each one of those instances. This latest option might be preferable in scenarios where the establishment of the new connections between new Network Services instances and/or deletion of existing connections are considered more expensive than simply expanding/reducing the capacity of the existing network services;
- Migrating the currently deployed Network Service instances and their interconnections to more performant/with more available resources node(s) (or less performant/cheaper nodes): the externally perceived performance of the Network Slice would be different, even though the only characterization that would change would be the nodes where it is currently running;
- Expanding/reducing the capacity of the existing Network Service instances: this is the simplest scenario of all, being applicable only in some very specific scenarios (usually related to a pattern of heavy data transportation/transformation), where the bottleneck is the interconnections between the multiple Network Service instances.

For all these scenarios the (externally) perceived capacity of the Network Slice (including its cost) can expand/reduce with different approaches taken at a lower level.

3.4.1.3 Network Slice instance migration

Depending on the concrete implementation(s), Network Slice instance migration might translate directly into the migration of the corresponding Network Service instances and interconnections between those instances. The obvious use case of a Network Slice migration is the one related with performance: for example, if the use case implies keeping a low latency in the end-to-end service, migrating some of the service instances comprising the slice instance might achieve the required objective.

This feature is one of the most dependent on an accurate run-time measurement, collection and analysis of data, in order for the whole set of involved service providers to be able to provide this capability on time. The ideal scenario here is to have at least some prediction capabilities for the need of this migration, so that such a complex operation can be triggered on time to fulfil the desired expectation.

Network Slice migration strongly depends on more basic features, such as instantiation (see above) and instance tear down (see next sub-section).

3.4.1.4 Network Slice instance tear down

Tearing down a Network Slice instance should be available only when any of its comprising service instances do not have any connection to any user. It should start by requesting any interconnection between these instances to be teared down, followed by tearing down each Network Service.

3.4.1.5 Network Slice deletion

Network Slice deletion capability should be available only if the specific Network Slice does not have an instance currently running. Again, all the necessary credentials must be in place in order for this feature to be executed, given that it will, in the general case, affect business rules on the involved partners.

3.4.2 Cross-Domain Network Service Lifecycle Management

A network slice can be considered as the composition of a set of slice components, connected according to a slice topology, and managed end-to-end, integrating the different participant domains by means of the necessary trust fabric (credentials, policies…) to coordinate its lifecycle across them.

The ETSI NFV [10] analysis of the support of network slices in software-based virtualized infrastructures maps the concept of the component of network slices, known in 3GPP as *network slice subnets* [11], onto the NFV concept of a network service. A network service constitutes the orchestration and management unit for a given NFV domain, and includes the description of the network functions of any nature, either physical (PNF) or virtual (VNF), the necessary connections among them (through the so-called Forwarding Graphs) and its attachment points to allow network service users to access it.

The management and orchestration domain delimiting service provisioning is not pre-defined, and relies on the different structuring a given network provider, or set of providers, can decide, according to technology, architecture and/or business considerations. In a 5G network environment, a natural division could belong to the different network segments: RAN, transport, edge, core and cloud, though other division are possible. It is important to remark that service management and orchestration (MANO) takes place under the control of a single MANO stack, coordinated by a single *orchestrator*, and supporting a single interface for service lifecycle and assurance management, as defined by ETSI NFV SOL005 [12]. The integration of the cooperating management domains is foreseen by the loosely coupled Service-Based Architecture defined by the ETSI ISG ZSM architecture [23], through the components of the slice descriptor depicted in the previous section.

The SOL005 interface supports the following lifecycle management operations:

- Related to network service descriptors
  - *Create NSD info*
  - *Upload NSD archive*
  - *Fetch NSD archive*
  - *Update NSD info*
  - *Delete NSD*
  - *Query NSD info*
  - *Read NSD*
  - *Fetch NSD archive manifest*
  - *Create PNFD info*
  - *Upload PNFD archive*
  - *Fetch PNFD archive*
  - *Update PNFD info*
  - *Delete PNFD*
  - *Query PNFD info*
  - *Read PNFD*
  - *Fetch PNFD archive manifest*
- Related to network service management
  - *Create network service identifier*
  - *Instantiate network service*
  - *Scale network service*
  - *Update network service*
  - *Query network service*
  - *Terminate network service*
  - *Delete network service identifier*
  - *Heal network service*
- Related to network service performance management
  - *Create monitoring job*
  - *Query monitoring job*
  - *Delete monitoring job*
- Related to operational and monitoring data
  - *Get operation status*
  - *Subscribe*
  - *Query subscription information*
  - *Notify*
  - *Terminate subscription*
  - *Create monitoring threshold*
  - *Query monitoring threshold*
  - *Delete monitoring threshold*

The interface also enables to invoke error handling procedures (Retry, Rollback, Continue, Cancel, Fail) on the operation occurrences, and API version information retrieval.

3.4.2.1 Cross-Domain Data-Driven Network Service

At the core of the 5GZORRO proposal are the distributed data mechanisms (data lakes, DLTs and off-chain data stores) used to support data-driven management of infrastructures and user-facing services. These data mechanisms become essential for many of the management operations described above, in particular:

- Service descriptors will be found by querying the different offers available at the off-chain data stores, as part of the operational data lake(s) or elsewhere, and trust scoring assigned by records held by the DLT(s). The same applies for the individual components of the service descriptors (VNFs and PNFs, and embedded services).
- Service instantiation and scaling will rely on resource offerings selected through a similar combination of queries and trust evaluations.
- Update and healing procedures of services and their components will follow an approach like the described above for their location and trust evaluation.
- Monitoring procedures of any nature will use as main destination the appropriate data lake(s).
- Subscription mechanism will include data lake importer interfaces as their main targets.
- All operation requests, results and error handling mechanisms will be recorded through the appropriate data lake(s) and DLT(s) to enable compensation schemas through tokenization and support further auditing and trust evaluation.

3.5 Cross-stakeholder e-license management

Software vendors need to materialize 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, like the cloud licensing models developed by Amazon [13] or Google [14].

However, Virtual Functions (including VNFs and CNFs) are software functions that can be instantiated and replicated very quickly thanks to the NFV technology in a multi-domain ecosystem. Their agility increases the challenge of the license control and management.

5GZORRO offers a cross stakeholder e-License management service to provide operators and software vendors the mechanisms to trustworthy control the usage of the vendors' software products, involving the operators and communication providers. Vendors will onboard their products specifying in the smart contract of the offer the license conditions, negotiation goal and constraints. The design of the service is focused on control the licenses at VF level, allowing the management of the licensing regardless of the location or the domain in which the VF is running. Besides, design contemplates the possibility of controlling the licenses of a Network Service, or a Network Slice composed by VFs from different providers. Different business models are evaluated for their implementation:

- *Flat*: Contract of the VF Vendor for a part or the complete set of features of the VFs without considering time or usage of the service.
- *Pay-as-you-Grow*: In this model, the price varies depending on the increase or decrease of the customer business. Thus, the final cost will be calculated based on one or several conditions of usage of the VFs, like number of instances of VFs or the number active users in a certain moment.
- *Subscription*: Operator contracts the right to use the VF for a certain period of time.

The e-License management service is designed on the metric-based control of the proprietary VFs in the different domains. In this way, every action produced in each domain for a controlled VF will be tracked and evaluated for the licensing fulfilment. The key concept that guarantees the trust in the e-licensing framework is provided by the nature of the DLT, i.e. all stakeholders are involved in the agreement and in the consensus of the usage of every VF as it is detailed in Section 5.3.12.

3.6 SLA monitoring & breach prediction

A *Service Level Agreement (SLA*) is an element of a formal, negotiated commercial contract between two Organizations, i.e. one with a Service Provider (SP) role and one a Service Consumer (Customer) Role [15]. It documents the common understanding of all aspects of the Product (what a service provider offers) and the

role and responsibilities of both Organizations from product ordering to termination. SLAs can include many aspects of a Product, such as performance objectives, customer care procedures, billing arrangements, service provisioning requirements, etc. The specification of the Service Level Commitments on the SP side is the primary purpose of an SLA. They include 1) the *Service Level indicators (SLIs),* which are the parameters (or metrics) chosen to be measured in a monitoring system, 2) the Thresholds, which are quantitative values to be reached by metrics and 3) a description of measuring, reporting and violation handling processes. Examples of SLI parameters are availability, reliability, throughput, bandwidth, response time, etc. Furthermore, the SLI parameters and the related thresholds express the *Service Level Objectives (SLOs*). Specifically, SLOs are the objectives that must be achieved, i.e. the target value or range of values for a service level. A typical example of SLO structure is: *lower bound ≤ SLI ≤ upper bound.*

SLA Monitoring is the process of comparing the measured SLA parameters (the SLIs) against the thresholds defined in the SLOs of an SLA. The SLIs can be periodically obtained from a monitoring subsystem and are examined against the guarantees given in the SLA. In case of violation a management system could be notified in order to take the appropriate actions.

3.6.1 SLA Monitoring service

The SLA Monitoring service collects, and analyses monitoring data in order to detect violations in SLAs. After each new contractual agreement, it receives the SLAs of each new contract from the service providers. Furthermore, it keeps the SLAs of all the active contracts and periodically requests resource metrics from the monitoring data provider.

SLA monitoring examines SLIs such as availability or response time by retrieving monitored data representing the overall service levels. SLA Monitoring analyses the monitored data and compares the metrics with the thresholds in SLAs in order to detect SLA violations. The service provider can be notified about SLA violations, which are subsequently propagated to the smart contract for the purposes of re-calculating SLA status. There may be momentary violations, violations of some duration, and violations that appear to be permanent. Contracts may specify the duration that an SLA violation must have in order to be considered as an SLA breach. Each SLA violation can by characterized by its type, the start time, the end time (if any) and the level of violation.

Available implementations of the SLA monitoring service are provided by the 5G TANGO project [16], where Prometheus [17] is used to analyse SLAs and detect violations as well as RabbitMQ [18] to produce violation alerts.

This service could operate in a Trusted Execution Environment (Section 11.4.5) to enforce trusted analysis on monitoring data and to assure, combined with DLT mechanisms, the trusted level required among Service Consumer and Service Provider for the analysis of monitoring data.

3.6.2 SLA Breach Prediction service

The SLA Breach Prediction service goes one step further from SLA monitoring as instead of detecting SLA violations in real-time, it can predict them before their actual occurrence. For this reason, it collects and analyses monitoring data from the monitoring data aggregator service of 5GZORRO using AI techniques in order to predict possible breaches in SLAs and detect anomalies. Anomaly Detection refers to the problem of finding instances or patterns in data that deviate from normal behaviour. It is important because anomalies often indicate useful, critical, and actionable information (for example problems in the provisioning of a resource or an intrusion which exhausts a resource).

SLA Breach Prediction is a novel service and hence not well established for 5G architectures. Hence, in the following part we will try to provide an introduction for the service in existing service-based architectures [20]. In such architectures, machine learning models are used for the prediction and are subsequently fitted with sufficient historical data. These data can be:

1) *Facts,* representing data which is already known at prediction time.

2) *Unknowns,* which are the opposites of facts, in that they represent data which is entirely unknown at prediction time.
3) *Estimates,* which lie in the middle between facts and unknowns, in that they represent data, which is not yet available, but can be estimated. An example of such data are the Quality of Service (QoS) data, since techniques as QoS monitoring can be used to get insights on the response time of a service before it is actually invoked.

The gathered data can be used for training and testing of the prediction model accuracy as well as can be evaluated with real-time data.

SLA Breach Prediction approaches are used for maintaining SLAs between service providers and consumers, such as the work presented in [21]. Specifically, in this work the authors propose a profile-based model for the prediction of SLA violations from the provider's perspective. Each consumer has a different profile and the prediction helps service providers in making decisions about whether to form SLAs as well as in avoiding SLA violations. The prediction model generates an alarm to the service provider that a violation is likely as well as generates a recommendation for remedial action. This gives the provider the opportunity to arrange appropriate resources to avoid the violation. Further methods to predict violations are linked to Bayes models for predicting the mean load over a long-term time interval [19]. Additionally, this method can be used for predicting the mean load in consecutive future time intervals by identifying novel predictive features of host load that capture the expectation, predictability, trends and patterns of host load. The combination with Bayes logic allows a service provider to estimate the probability of whether SLA violation will occur.

SLA breach prediction can be also used to provide resilience in a service-based architecture. Specifically, resilience is linked not only to operational violations, but also to cyber-security-oriented violations. Such violations occur as a result of malicious activities that are initiated by adversaries and are located at different service levels as the network, application or even the operating system level. To perform such activities, adversaries need to first gain access to a service by identifying and exploiting its vulnerabilities (Section 3.7.2). This allows them to either cause direct damage to the service or even leverage connections to other services or domains to magnify the attack's impact. The malicious activities that are triggered by adversaries can be detected using rule-based or knowledge-based techniques. Rule-based techniques are very effective in detecting known cyber-attacks based on a list of existing signatures. However, they require a frequent update of their signatures as well as cannot cope with sophisticated or zero-day attacks. Hence, knowledge-based techniques are also employed in order to learn the normal behaviour of each service and create a profile of nominal operation. Such profile can be created using statistical or AI techniques. Both rule and knowledge-based techniques lead to the detection of anomalies that can either be: 1) accurate SLA violation predictions or 2) false positives and false negatives. For the first point the predicted incident actually occurs and such SLA violation is categorized as a true positive. For the second point a false positive indicates that a predicted incident does not occur and a false negative that a real incident that has occurred was not predicted.

Finally, the SLA Breach Prediction service could also operate in a Trusted Execution Environment (Section 11.4.5) to enforce trusted prediction of SLA violations on monitoring data.

3.7 Security and trust across multiple domains

The massive increase in device quantity, together with the increase in activities and connections available on devices that interact in a 5G network lead to an increase in potential risks and threats that both end-users and service providers may suffer.
In this sense, it is necessary to define new distributed models to develop efficient connectivity in 5G networks, through which a group of entities (whether small or large) can establish cross-domain/operator service chains, with trust and security.

3.7.1 Identities and trust across multiple domains

Distributed trust models can allow network connections to be established between domains reliably, avoiding possible connections that could endanger user data integrity or compromise the security of service providers and end-users.

Key to the realisation of trust across domains is the use of decentralized identity management (DIdM), which is based on Decentralized Identifiers (DIDs).

DIDs are a novel type of identifiers proposed by W3C [55] that allows associating any subjects such as stakeholders, resources, services, organizations, entities, and so on, with a digital identity. DIDs are global identifiers which enable verifiable and decentralized digital identity, allowing to uniquely identify any subject, e.g. a person, organization, abstract entities, etc. To achieve this purpose, DIDs are associated with cryptographic material, such as public keys, and service endpoints, making each DID globally unique, resolvable with high availability, and cryptographically verifiable.

The usage of DIDs provides to an application of self-administered identity management, enabling further self-managed capabilities such as authentication, authorization, role management, and identity information exchange between two identity domains.

Another concept related to decentralized identity management is Verifiable Credentials. A Verifiable Credential (VC) [56] is a tamper-evident and privacy-preserving credential (set of claims) that can be demonstrated through a cryptographic process. Verifiable Credentials can represent the same information that physical credentials represent in real life such as driving licenses, passports, health insurance card, and so on. Therefore, Verifiable Credentials represent statements made by an issuer in a tamper-evident and privacy-preserving manner.

3.7.2 Detection and countermeasures for security vulnerabilities

The massive number of new devices connected in 5G networks entails the emergence of new security challenges, which in turn leads to a bigger attack surface. 5G systems deployment is expected to bring numerous security risks, threats, and challenges in a multitude of scenarios such as network virtualization, the rise of new communication protocols and their compatibility with existing, or even critical communication infrastructure.

Security is an essential characteristic that is usually interconnected to other characteristics, and therefore should not be addressed properly on its own without considering other factors. A possible example is the correlation between security and trust. By definition, both characteristics complement each other, that is, a trust establishment between two providers will be guaranteed if services and protocols deployed in both entities are secure, and a communication will be safe if the involved entities are trustworthy (non-malicious) and do not attempt to alter information flow.

To tackle all efforts towards a risk-free 5G multi-stakeholder scenario, a risk analysis service is required. 5GZORRO architecture includes actions related to risk detection, assessment, and treatment via a risk management methodology, which is part of one of the services that make up the end-to-end trust and security model in distributed multi-stakeholder scenarios. In particular, the actions related to risk management are inspired by the ISO/ICE 27005 standard, but it should be noted that none of the entities that make up 5GZORRO's ecosystem possess and/or such a certificate. In this sense, 5GZORRO intends to use only some steps of this standard, since they are considered a universal approach to risk management. Among the actions to be followed are the definition of assets to be included in the process and collection of all necessary information to gather the relevant risks (also known as context establishment), and the risk identification, risk analysis, and risk evaluation (risk assessment). Besides the detection of software vulnerabilities and compromises, risk analysis service provides a set of countermeasures for any of these detected vulnerabilities and its implications on the network services (risk treatment).

4 Reference architectures & technologies

Specifications from many Standard Development Organizations fall into the scope of the 5GZORRO architecture. Similarly, several technologies and tools are used as core enablers of the 5GZORRO architecture. Indeed, the 5GZORRO architecture has been designed considering various specifications and technologies in order to leverage on the most of them and realize a multi-operator zero-touch service management.

For sake of readability and because of the large set of relevant references, the following tables provide a summary overview of the 5GZORRO core focus aspects with respect to

- reference standards and architectures in state of the art, split across three major area: *Intelligent zero-touch Management*, *Cross-domain resource & service trading*, and *Security and Trust* (see Table 4-1)
- reference technology enablers (see Table 4-2).

Details for the cited references are provided in Appendix I (see Section 11), assuming that most of the general overview information on applicable SDOs and technologies might be already known to the reader.

Table 4-1: 5GZORRO focus aspects from reference architectures in different areas

Area	Standard Development Organisation	5GZORRO Focus Aspects	Overview details
Intelligent zero-touch Management	• **ETSI ZSM**	• Zero-touch and domain-oriented management • Service-based architecture	See Sec. 11.1.1
	• **ETSI NFV MANO**	• Network Slice (incl. GSMA and 3GPP reference templates) • Network Slice Network Service, VNF lifecycle management	See Sec. 11.1.2
	• **ETSI ENI**	• AI/ML based policy management, orchestration decisions and suggestions	See Sec. 11.1.3
Cross-domain resource & service trading	• **TMForum Telecom infrastructure marketplace**	• Infrastructure sharing & trading principles and trials • Spectrum sharing and licensed spectrum trading • Marketplace-Orchestration platforms interoperability	See Sec. 11.2.1 and 11.2.2
	• **ITU-T**	• Permissioned DLT reference architecture	See Sec. 11.2.3
	• **ETSI PDL**	• Permissioned DLT reference architecture	See Sec. 11.2.4
	• **MEF LSO SONATA**	• Offering information models and management • Multi operator connectivity services	See Sec. 11.2.5
	• **CBAN**	• Permissioned DLT reference architecture • Offering information models and management • DLT-based Applications and Marketplace	See Sec. 11.2.6

Security and Trust	• **ITU-T Y.3054** • **ETSI TR 10368** • **ISO/IEC TR 23186**	• Trust modelling and computation techniques • Marketplace-Orchestration platforms interoperability	See Sec. 11.3

Table 4-2: 5GZORRO focus aspects from technology enablers

Technology	Enabling solutions	5GZORRO Focusing Aspects	Overview details
Distributed Ledgers & Smart Contracts	• **R3 CORDA**	5GZORRO will employ DLT and Smart Contracts technologies for realization of a Marketplace with contract negotiation and provisioning of resources & services and SLA enforcement.	See Sec. 11.4.1
Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs)	• **Hyperledger Indy** • **Hyperledger Aries**	5GZORRO will use DIDs and VCs to enable a multi-stakeholder decentralized identity management solution. Also, DIDs will be employed to identify the resources offered in the 5GZORRO marketplace, and VIM/NFVI.	See Sec. 11.4.2
Data Lakes	• **OpenDataHub** • **CNCF OpenTelemetry**	5GZORRO will use and enhance data lake technologies to aggregate and store relevant monitoring data from different stakeholders, and to perform analytics to predict SLA breaches.	See Sec. 11.4.3
Artificial Intelligence solutions for Network Management	• **Feed Forward Neural Networks (FFNN)** • **Long Short-Term Memory networks (LSTM)** • **Convolutional Neural Networks (CNN)**	5GZORRO will employ different machine learning and deep learning algorithms for VNF auto-scaling, Network Slice resource auto-scaling, SLA breach prediction and smart resource Discovery	See Sec. 11.4.4
Trusted Execution Environments	• **HW TEE** • **SW TEE** • **Mixed HW/SW TEE**	5GZORRO will be using TEE to protect the data that are stored locally in each DLT node and to isolate the VIM or NFVI components allowing to protect the sensitive data of applications and services that are running on them.	See Sec. 11.4.5
Cloud-native technologies for 5G	• **Kubernetes** • **ISTIO** • **NSM**	5GZORRO will be using a container engine in order to realize an environment where CNF and VNF can coexists and an orchestrator (Kubernetes) to manage these containers. ISTIO and NSM will be used to provide communication between services, possibly instantiated through VNF or CNF.	See Sec. 11.4.6

5 5GZORRO High-level Reference Architecture

5.1 Design principles

A set of common best practices in design of automated systems implementing zero-touch service orchestrations have been analysed and adopted to design the 5GZORRO architecture. In general terms, the 5GZORRO architecture is inspired by the following design principles:

- **Service based architecture**, as in 3GPP and ETSI ZSM
- Allow separation of **responsibilities & scopes per domain/inter-domain**
- **Modular and scalable architecture** which offers self-contained services, which can be independently deployed and scaled.
- **Extensible architecture** which allows to add new services, capabilities and service end-points in a pluggable manner, without requiring changes to existing designs, implementations and interactions
- **Model-driven architecture with open interfaces,** which uses information models to capture the attributes and supported operations of the managed objects. The information models and interfaces are independent from implementation and are modelled in YAML and Open API specification to facilitate portability and reusability.
- Adopt **communication mechanisms** capable to implement both publish-subscribe patterns and direct invocations among functions for: a) Network Slice lifecycle management; b) Service and resource discovery/management; c) Network analytics; d) Security & trust.
- **Distributed architecture with instances in all domains** of the involved 5GZORRO parties
- Includes a **distributed trusted data layer** for SLA enforcement, resource discovery and smart contracts management implemented through **Permissioned Distributed Ledger Technologies**
- Includes an **Operational Data Lake** capable to collect telemetry data from various domains and services and to implement AI-driven insights on service, resource and infrastructure operations.

5.2 Architecture overview and core building blocks

The 5GZORRO High Level reference architecture is depicted in Figure 5-1. It is comprised by four major logical layers grouping different functionalities types according to previously defined service-centric architectural model principles. It should be noted that such logical grouping has no implementation meaning as shown in chapter 7.

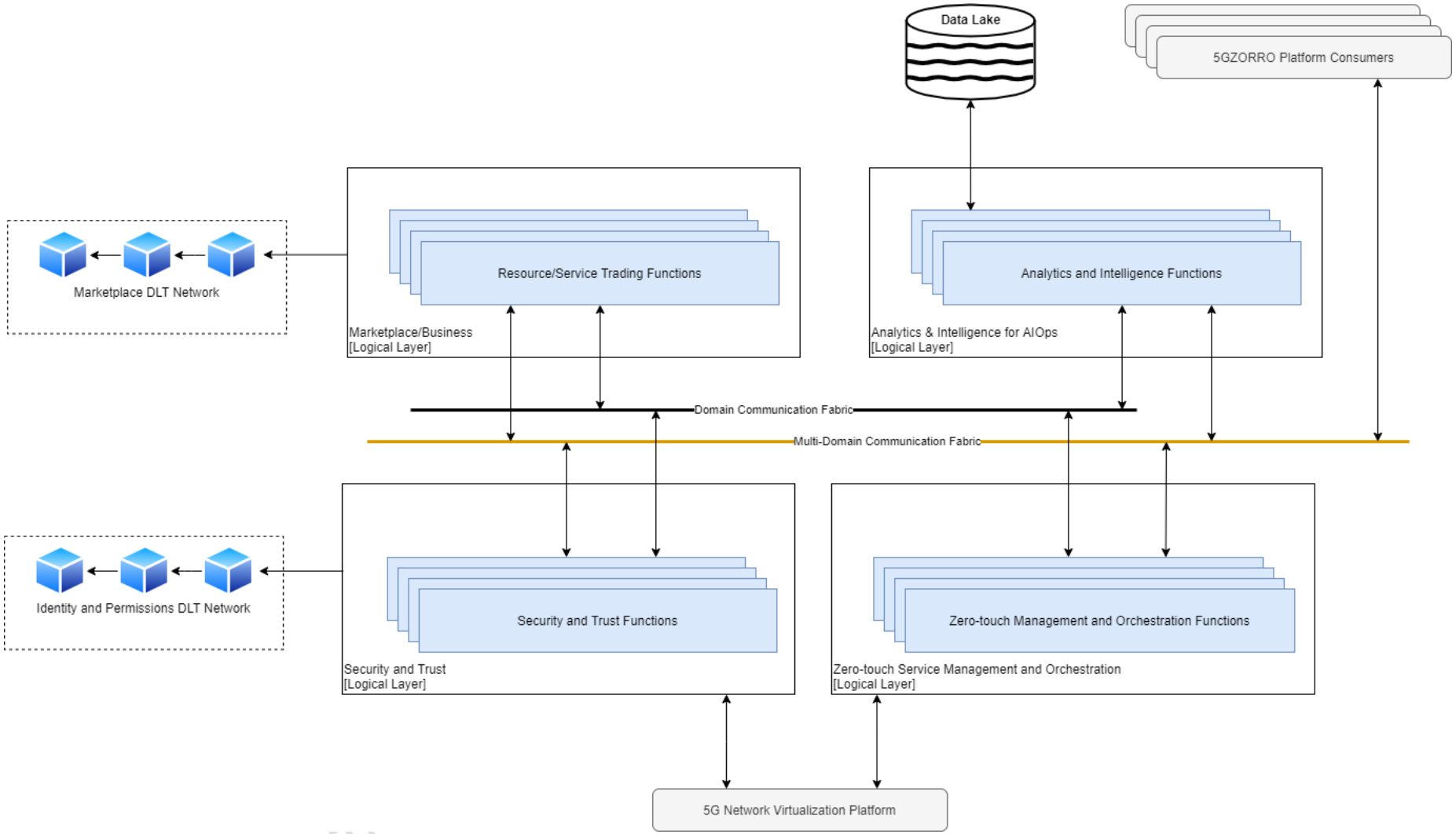

Figure 5-1: 5GZORRO High Level reference architecture

The **Zero Touch Management and Orchestration** layer (see Figure 5-2) provides the required functionalities to control 5G Managed Infrastructure Resources including Radio Spectrum resources, Transport Networking resources and Computing resources (at data centers and at edge computing nodes) as well as existing legacy resource controllers from previous 5G deployments. It applies ETSI zero-touch network and service management architectural patterns to enable zero-touch automated management of 5G networks including the end-to-end management of the life-cycle of network slices and associated services. 5G Resources are managed through Intent-based interfaces (or end-points) that provides a very high-level abstraction of 5G resources capabilities hiding complexity, technology and vendor-specific details.

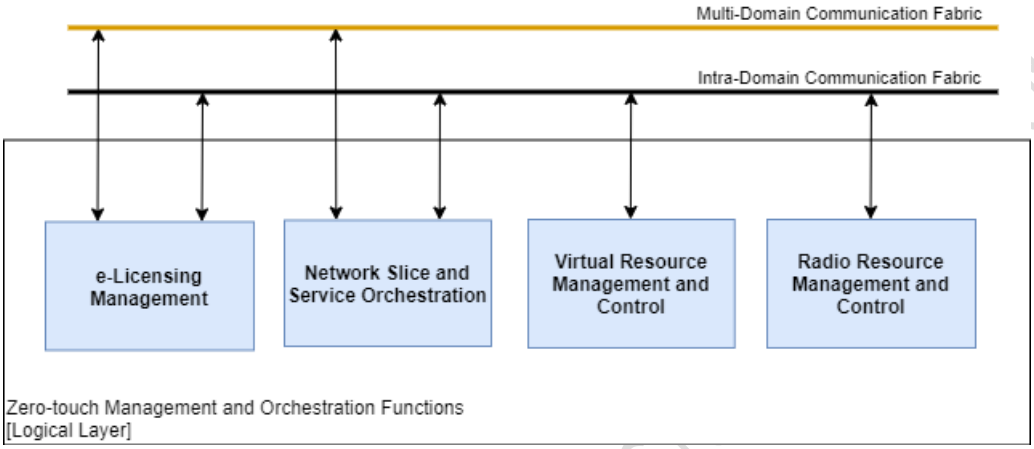

Figure 5-2: Functional Elements populating the Zero Touch and Orchestration layer

The **Security and Trust** layer (see Figure 5-3) provides a generic framework to administrate the trust and security evaluation of internal entities and resources, and the ones of other stakeholders. A key feature of the security and trust framework is to evaluate the confidence on the infrastructure and activities of other stakeholders in order to decide with which stakeholder commercial relationships will be established based on its security and trust properties. Another major feature of the Security and Trust layer is the management of global (cross-domain) identifiers (e.g. stakeholder identifiers and 5GZORRO resource identifiers) according to Self-sovereign Identity principles by leveraging DLT technologies. It supports the creation, verification and revocation of certificates as well as authentication and authorisation of identities across 5GZORRO domains.

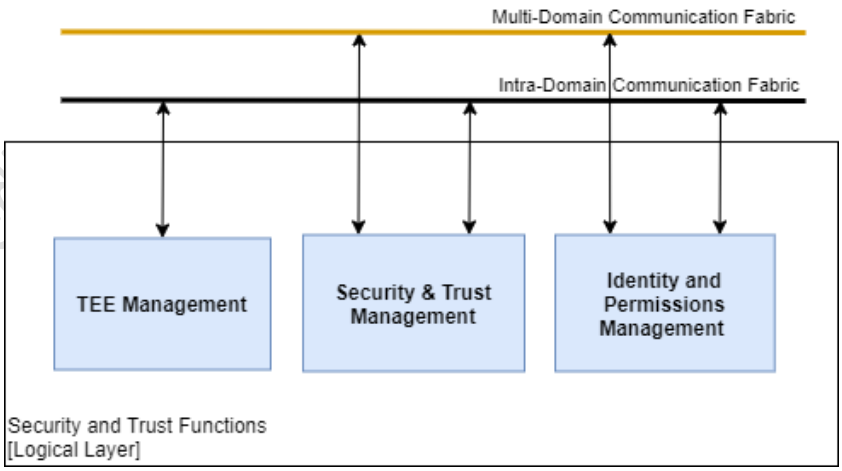

Figure 5-3: Functional Elements populating the Security and Trust layer

The **Marketplace and Business layer** (see Figure 5-4) enables the trading of 5G resources (including Radio Spectrum resources) across different domains by using DLT Smart Contracts. The Marketplace is ruled by a decentralized Governance Model where Governance Administrators (i.e. 5GZORRO stakeholders with permissions to vote) can take decentralized decisions such as accept/reject network participation, issue/revoke membership rights, dispute resolutions. Major Marketplace features are decentralized

catalogues for 5G Resource offers and 5G Service offers, decentralized repository for legal prose statements to be used in smart contracts and the life-cycle management of smart contracts for offers and agreements between providers and consumers.

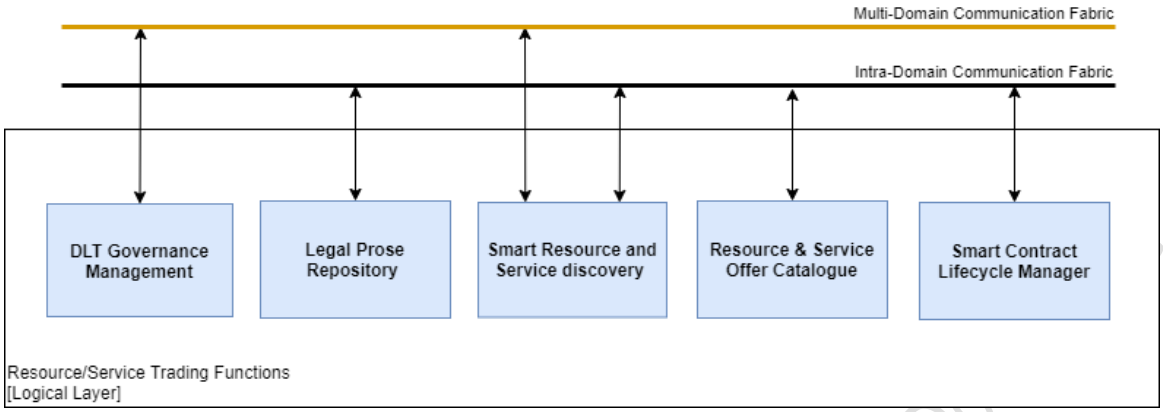

Figure 5-4: Functional Elements populating the Marketplace and Business layer

The **Analytics & Intelligence for AIOps** (see Figure 5-5) layer leverages data lake and AI technologies to provide data persistence, data share and data analytics, across 5GZORRO framework within and across domains. It enables the automation of complex resource management procedures like proactive scaling mechanism to increase or decrease the infra-structure capacity by using external resources published in the marketplace by Resource Providers. The prediction of SLA breaches and the discovery of the most appropriated resources available in the marketplace to handle it, is another example of a complex service that can be provided by this layer.

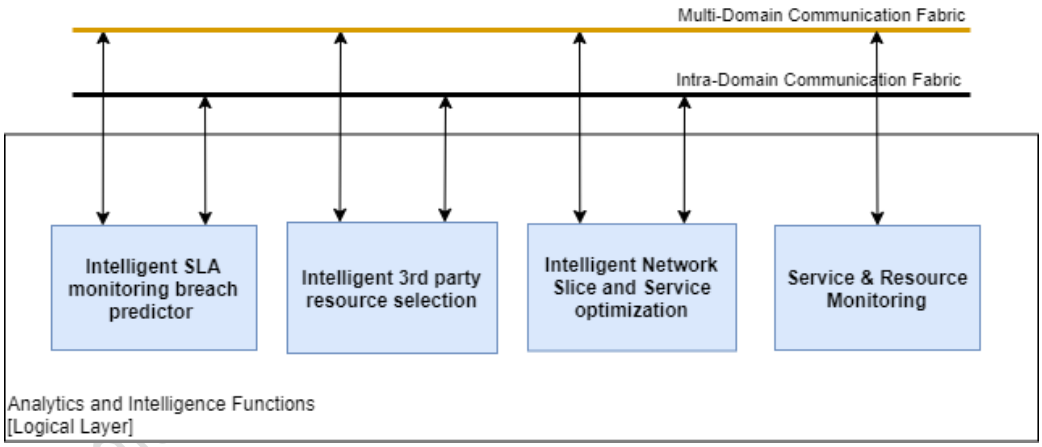

Figure 5-5: Functional Elements populating Analytics & Intelligence for AIOps layer

The **Communication Fabric** provides all required functionalities to support the communication and interoperation across 5GZORRO framework in a loosely coupled way, within or across domains. The registration, discovery, invocation of 5GZORRO services, are major functionalities provided by the Communication Fabric. Depending on the communication needs and each Domain policies, different types of communication patterns should be supported including: Synchronous communications; Asynchronous (pub/sub) communications; Point-to-point communications; Brokered communications.

Different instances of Communication Fabrics may be deployed in 5GZORRO:

- *Intra-Domain (or Domain) Communication Fabric*, where registered services are available within an administrative 5GZORRO domain

Cross-Domain Communication Fabric, where registered services are available across all administrative 5GZORRO domains. It is envisaged to leverage the DIDs DLT to provide a decentralised inter-domain service registry.

5.3 Specification of the 5GZORRO functional blocks

This section describes the 5GZORRO Functional Entities deriving from the functional breakdown of the entire reference architecture. Not all of these entities are exposed to 5GZORRO users: in fact, some of them remain internal and support the execution of zero-touch or security actions.

5.3.1 DLT Governance Management

5.3.1.1 Overview

The marketplace will be subject to a consortium governance model. This will ensure that decisions such as admittance, revocation of membership and dispute resolution is managed in accordance with a mutually agreeable governance model; acceptance of these terms would be mandated as part of registration along with KYC & AML checks.

5.3.1.2 Provided Services

To support this, a Marketplace Governance service will provide the necessary API to enact all governance actions. It will be subject to Role Based Access Control (RBAC) in order to ensure that only members of the Governance Admin role are able to - for example – vote on governance decisions. Stakeholders should be notified of any governance decisions, as well as any intermediary steps e.g. received, accepted, approved, and rejected.

Table 5-1: Definition of Governance Service (cross-domain level)

Service name: ***Governance Service***		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Apply for membership	**M**	Facility for a new stakeholder to apply for Marketplace membership including the claims that are being requested and specification of how to be notified of marketplace events e.g. governance decision
Check membership status	**M**	Allow a platform user to check the membership status of a particular Stakeholder, for example a stakeholder can check the status of their application.
List members	**M**	Obtain a list of active Marketplace members to allow a governance administrator to review members, their status, approved roles etc.
Revoke membership	**M**	Revoke a stakeholder's Marketplace membership including to enable a stakeholder to 'leave' the Marketplace. In the case where a governance decision is against the will of the member being revoked, the decision will be taken according to the governance model by using the operations below
Propose governance decision	**M**	Propose something that is to be decided according to the governance model
Get a proposed governance decision	**M**	Retrieve the details pertaining to a particular proposed governance action
Vote on a governance proposal	**M**	Accept/Reject something to be decided proposed using the 'Propose governance decision' function

Vote on a governance decision	**M**	Vote on a governance decision accepted using the 'Vote on governance proposal' function
Notes		
none		

5.3.2 Resource & Service Offer Catalogue

5.3.2.1 Overview

The Resource & Service Offer Catalogue is the place where shared network, infrastructure and spectrum resources for exchange are advertised.

5.3.2.2 Provided Services

The Resource & Service Offer Catalogue provides to 5GZORRO users the possibility to add resources or services offers to the 5GZORRO Catalogue, list the active resource and service offers, query, modify or remove a resource or service offer, and make an offer for a specific resource or service. The catalogue management service is also responsible for maintaining the Resource & Service Offer Catalogue up to date. Resource and Service catalogues will offer interfaces for querying Marketplace offers. Requesters will be able to make 'simple' criteria requests that simply query the catalogue's off-chain storage for available resources or services meeting a basic set of requirements. The requester will then be able to apply any domain-specific intelligence they wish to further filter results.

Table 5-2 defines the Resource & Service Offer Catalogue services.

Table 5-2: Definition of Resource & Service Offer Catalogue service

Service name: **Resource & Service Offer Catalogue**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Register Resources	**M**	Update list of resources that Operator declares to be available for external users.
Manage offer lifecycle	**M**	The Resource & Service Offer Catalogue is able to perform the typical management operations including creation, modification, query and removal of resource and service offers
Provide offer information	**M**	The Resource & Service Offer Catalogue can be requested to provide information on a general or a particular set of offers in the Catalogue
Purchase an offer	**M**	The Resource & Service Offer Catalogue also has the mechanisms to enable transaction on the resources and services offered in the 5GZORRO architecture. Once a transaction is done, the offer in the Catalogue must be marked as not available
Notes		
none		

5.3.3 Legal Prose Repository

5.3.3.1 Overview

The Legal Prose Repository will be a shared repository of parameterised legal statement templates that can subsequently be associated with a given resource or service by providers. These statements will be subject to a governance process such that admin network members will need to approve any new or updated statements since their definition will relate directly to verification logic in equivalent Smart Contracts. Legal prose instances will consist of both human readable and machine-readable portions, the former being referenced by smart contracts should a need to refer to it arise e.g. to mediate a dispute.

Legal prose will reflect SLAs and their associated SLOs and SLIs, feeding into active agreements and associated monitoring during the agreement's lifecycle.

5.3.3.2 Provided Services

Services will include the ability to create, modify and archive legal prose templates, as well as additional governance operations to manage their versioning and publication.

Table 5-3: Definition of Legal Prose service (cross-domain level)

Service name: **Legal Prose Repository**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Get Legal Statement Templates	**M**	Retrieve a set of legal statement templates that can subsequently be used to define terms for a given Resource or Service Offer
Create Legal Statement Template	**M**	Provide the ability to create a new template that corresponds with a particular Smart Contract
Update Legal Statement Template	**M**	Provide the ability to update an existing template
Remove Legal Statement Template	**M**	Provide the ability to remove a template from active use
Approve Legal Statement Template	**M**	Facilitate a review/governance process that ensures a template has been legally ratified but also that it meets the requirements of the associated Smart Contract
Notes		
none		

5.3.4 Smart Resource and Service discovery

5.3.4.1 Overview

For more complex 'smart' queries, the catalogue will utilise a Smart Discovery module provided by Cross Domain Monitoring and Analytics. Results will be filtered/prioritised using smart selection techniques based on machine learning. Catalogue data and metrics will be provided by the public data lake for smart selection to be based on.

5.3.4.2 Provided Services

Discovery will be facilitated by a Resource Catalogue and Service Catalogue services described above. Each exposing endpoint for querying resources and services respectively; both simple and smart selection methods.

Table 5-4: Definition of Resource and Service Catalogue service (domain level)

Service name: **Resource Catalogue**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Get Resource Offers	**M**	Provide a mechanism for querying Resource Offers for the requester, with results being filtered based on the provided criteria. Criteria may dictate 'smart selection', in which case a request would be made to the Smart Discovery module for results rather than querying local off-chain storage.
Get Service Offers	**M**	Provide a mechanism for querying Service Offers for the requester, with results being filtered based on the provided criteria. Criteria may dictate 'smart selection', in which case

		a request would be made to the Smart Discovery module for results rather than querying local off-chain storage.
Notes		
none		

5.3.5 Intelligent 3rd party resource selection

5.3.5.1 Overview

If an Operator requires resources beyond what he has available, he may be able to obtain use of resources from some other Operator. The Smart Resource Discovery entity in the Cross-domain services provides an inventory of resources that other Operators have declared as available for sharing. The Domain Intelligence entity in the Core Platform receives the list of available resources from the Smart Resource Discovery entity. From among these available resources, the most appropriate resources must be chosen. To decide which resources to select, various considerations can come into play. There may be a historical or business preference to work with a particular 3rd party. There may also be considerations of network topology, such as proximity of available resources to existing resources in use to reduce network latency. It may be possible to collect historical statistics on use of 3rd party resources, build a model on effectiveness of using those 3rd party resources, and decide what 3rd party resources to choose based on the model.

5.3.5.2 Provided Services

The Smart Resource Discovery entity in the Cross-domain services provides an inventory of resources that other Operators have published in the Marketplace catalogue. There is an interface to declare which resources are made available to others and an interface to obtain a list of available services.

Resources are selected among those which satisfy some input SLA/SLO. If there are insufficient local resources, the system may turn to the cross-domain Smart Resource Discovery to obtain information on external resources that may be available.

Table 5-5: Definition of Intelligent 3rd party resource selection service (domain level)

Service name: **Smart Resource Discovery**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Get Resources	**M**	Obtain list of resources needed to satisfy some customer/SLA, etc. May call cross-domain Smart Service Discovery to obtain externally available resources.
Notes		
None.		

Table 5-6: Definition of Intelligent 3rd party resource selection service (cross-domain level)

Service name: **Smart Resource Discovery**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Get Resources	**M**	Provide list of available resources.
Notes		
None.		

5.3.6 Smart Contracts Lifecycle Management

5.3.6.1 Overview

Management of contracts throughout their lifecycle, from agreement negotiation and instantiation through to termination will be managed by an SLA & Licensing Manager service and Smart Contract Lifecycle Manager. The SLA & Licensing manager will mediate interactions with stakeholders such as coming to an off-chain

agreement of terms or viewing active agreement i.e. it's primary role will be based around viewing and updating contract state. The Smart Contract Lifecycle manager on the other hand, will work closely with the SLA & Licensing manger but provide services specific to the lifecycle of an agreement and associated events triggered by contracts.

5.3.6.2 Provided Services

Table 5-7: Definition of SLA & Licensing Manager service (cross-domain level)

Service name: **SLA & Licensing Manager**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Resource Agreement Proposal	**M**	Facility to create an agreement proposal for a given set of Resource Offers. Each resource's availability status would be checked and confirmed to the requester.
Resource Agreement Creation	**M**	Facility to take a set of proposed resource agreements, and 'deploy' the necessary smart contracts to realise the agreement on the DLT. On successfully establishing the agreement this event will be published to prompt a cross-domain monitoring & analytics component to configure a data aggregator to provide the measurements at intervals agreed in the contract (see 5.3.13 & 5.3.14)
Service Agreement Proposal	**M**	Facility to create an agreement proposal for a given Service Offer. The availability status of each component Resource Offer that comprises the service will be checked and confirmed to the requester
Service Agreement Creation	**M**	Facility to take a Service Offer agreement, and 'deploy' the necessary smart contracts to realise the agreement on the DLT. On successfully establishing the agreement this event will be published to prompt a cross-domain monitoring & analytics component to configure a data aggregator to provide the measurements at intervals agreed in the contract (see 5.3.13 & 5.3.14) for each resource/service that comprises the service
Terminate an agreement	**M**	Facility for a stakeholder to terminate an active agreement. As per creation, termination of an agreement should teardown the necessary configured monitoring by publishing the event for a cross-domain monitoring & analytics component to process (See 5.3.13 & 5.3.14)
Get agreement	**M**	Facility for stakeholders to retrieve the details of one of their agreements
Get agreements	**M**	Facility for stakeholders to retrieve a filtered list of one or more of their agreements
Update resource agreement	**M**	Facility for a stakeholder to update an active resource agreement e.g. make a pricing adjustment. When an agreement is updated, the update will be published to prompt a cross-domain monitoring & analytics component to re-configure monitoring aggregation to align with the updated contract terms (see 5.3.13)
Update service agreement	**M**	Facility for a stakeholder to update an active service agreement e.g. make a pricing adjustment. When an agreement is updated, the update will be published to prompt a cross-domain monitoring & analytics component

		to re-configure monitoring aggregation to align with the updated contract terms (see 5.3.13) for each resource/service that comprises the contract.
Record monitoring measurements	**M**	A contract will define a monitoring aggregation service that has been agreed to be the sole provider of SLA related metrics. At intervals defined in the contract measurements will be submitted to the SLA & Licensing Manager (see section 5.3.13) which will subsequently locate the associated smart contract and notify the smart contract should a breach have a occurred. The smart contract will trigger any subsequent actions as determined by the terms of the agreement. In addition to the measurement, a hash of the monitoring data and the period for which that accounts should be supplied in the request to the SLA & Licensing Manager in order for it to be stored on the ledger to achieve non-repudiation of the submitted raw logs.
Submit licensing action	**M**	Licensing-related actions need to be recorded and validated by Smart Contracts. For example, recording a scaling operation to add an additional VNF instance would need to verify that this action is in-line with the agreed licensing terms. The response would indicate whether the action is valid/invalid based on the terms; see section 5.3.12.
Notes		
'Record monitoring measurement' and 'Submit Licensing Action' may provide interfaces or simply rely on messaging protocols. It will depend upon how communications between the Cross-domain SLA Monitoring & Analytics and eLicensing manager respectively interact with the marketplace. In both these cases, they could be functional elements that subscribe to measurements/actions.		

Table 5-8: Definition of Smart Contract Lifecycle Manager service (cross-domain level)

Service name: **Smart contract lifecycle manager**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Compose and deploy smart contact	**M**	Take an agreed set of terms and deploy the necessary contracts to the DLT
Subscribe to and manage lifecycle events	**M**	Subscribe to contract events such as SLA threshold warning, SLA breach, termination etc.
Notes		
none		

5.3.7 Identity Management and Permissions Management

5.3.7.1 Overview

The goal of Identity Management and Permissions Management is to supply the mechanisms required for generating unique identifiers in 5GZORRO ecosystem, recognising communicating endpoints, identifying and authenticating entities, services, and organizations, and authorising consumer requests to access a preserved services and resources.

In its present form, Identity Management is able to identify providers, consumers, services, resources, organizations, etc., using Decentralised Identifiers (DIDs) associated with DID Documents. What is more, DIDs can also be used for authentication through Verifiable Credential linked to a DID Document. In the case of

Permissions Management, this allows setting up a secure layer that regulates the access to resources, services, and delimited areas using a set of policies and rules. By means of policies and rules, each domain can determine the amount of information exposed, the duration for which that information is shared, what kind of information is shared, limiting resource capabilities, and so on. Therefore, each domain must define its policies and rules based on its criteria such as improving security, usability, availability, and cost-efficiency. In the end, Permissions Management attempts to prevent unauthorised access to services, resources, and data, making access control enforcement as granular as possible.

5.3.7.2 Provided Services

The main services provided by Identity and Permissions Management mechanisms are an appropriate mechanism to identity entities, services, resources, consumers, providers, and organizations, which allows decentralisation of the system without forgetting the security principles, a reliable authentication using Decentralised Identifiers (DIDs), DID Documents, and Verifiable Credentials, and finally, a granular control access mechanism that standardises authorised access to data, resources, and services.

Table 5-9: Definition of identity and permissions management service (domain level)

Service name: **Per-domain Identity and Permissions Management**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Create Identity	**M**	It allows registering new stakeholders, services, and organization in 5GZORRO ecosystem.
Revoke Identity	**M**	It removes a decentralised identifier (DID) from 5GZORRO system.
Authenticate Identity	**M**	It verifies the connection between the identifier and the holder.
Authorise Identity	**M**	It checks whether DID' holder has access to resources, critical actions, secure environments, etcetera.
Create Document	**M**	It allows generating a template where holder introduces information associated with him/her/its.
Replace Document	**M**	This capability enables you to remove a document from the system and replace it with a new one.
Update Document	**M**	It performs an iterative update to the existing document.
Read Document	**M**	It enables to acquire information about an identifier.
Check Operation in Progress	**M**	It returns the current status of an unfinished DID operation.
Check operation	**M**	It gives back the last status of a finished DID operation.
Create Claim	**M**	This capability associates a Verifiable Credential with an identifier.
Verify Claim	**M**	It allows proving the authenticity of a document associated with an owner.
Assert Claim	**M**	It is used to restrict the amount of information and limit the duration in a Verifiable Credential.
Store Claim	**M**	This capability locates claims in one or more subject decentralised repository.
Retrieve Claim	**M**	It is utilised to get information about a Verifiable Credential.
Notes		
none		

Table 5-10: Definition of identity and permissions management service (cross-domain level)

Service name: **Cross-domain Identity and Permissions Management**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Authenticate Identity	**M**	It verifies the connection between the identifier and the holder.
Authorise Identity	**M**	It checks whether DID' holder has access to resources, critical actions, secure environments, etcetera.
Read Document	**M**	It enables to acquire information about an identifier.
Retrieve Claim	**M**	It is utilised to get information about a Verifiable Credential.
Notes		
none		

5.3.8 Trust & Security Management

5.3.8.1 Overview

The Trust & Security Management functional block aims to provide the capabilities and operations required to administrate the trust and security evaluation of internal entities and resources, and the ones of other stakeholders.

A key capability for 5GZORRO architecture is to evaluate the confidence on the infrastructure and activities of other stakeholders in order to decide with which stakeholder commercial relationships will be established based on its security and trust properties, for example buying processing capabilities to a 3rd party in order to enhance the service performance.

Figure 5-1 shows a diagram of the architecture model of the Trust & Security functional block. It is divided into different blocks based on the proposed Trust & Security management process.

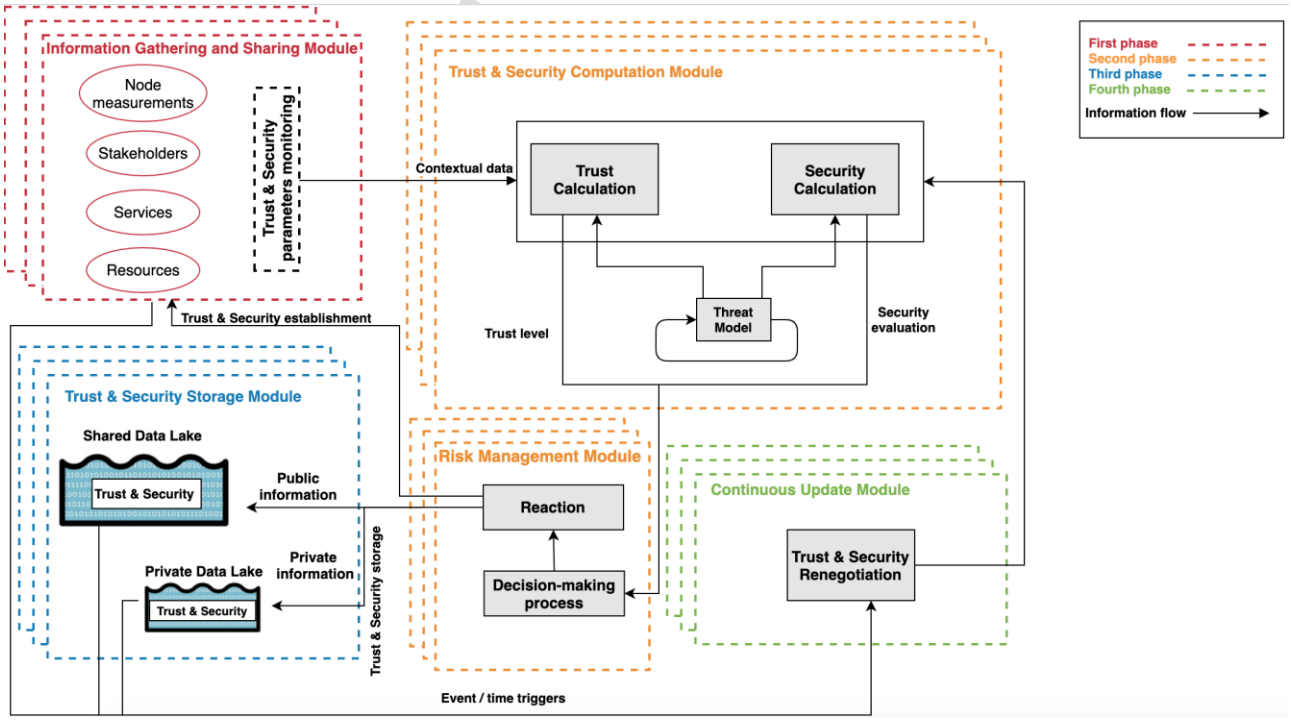

Figure 5-6: Trust & Security architecture model.

Trust & Security management services and interfaces are described separately, differentiating between intra- and inter- domain services. Based on the current design, 5GZORRO will follow a Zero Trust approach for the management of the resources, this is, there is no differentiation or security guarantee based on the resource location, this is, if the resource is inside or outside the stakeholder private network. For this reason, the operations utilised to manage internal and external Trust & Security are the same.

Besides, note that the Trust & Security services can, and should, be deployed and utilised in parallel, combining their capabilities to ensure a correct functioning of the rest of the elements of the architecture.

5.3.8.2 Provided Services

Table 5-11: Definition of trust & security management service (domain level and cross-domain)

Service name: **Intra-domain trust management**		Type: ***Per-domain*** **&** Type: **Cross-domain**
Capabilities	**Support (O|M)**	**Description**
Gather Information	**M**	It collects available information from an interested party from data sources.
Compute Trust Level	**M**	This method calculates trust level of the stakeholder to some internal resource using the previously acquired parameters.
Store Trust Level	**M**	This capability enables to records the previously calculated trust level and the utilized information in the available storage sources.
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 calculation process.
Compute Security Level	**M**	This capability enables to calculate the security level of the internal resource using the previously acquired parameters.
Store Security Level	**M**	This capability enables to records the previously calculated security level and the utilized information in the available storage sources.
Query Security Level	**M**	This capability enables to request the current security level of an internal resource. If there is no value calculated, it triggers the calculation process.
Notes		
Note that the methods utilised in intra and cross-domain to enable the previous capabilities are the same based on the Zero Trust approach followed by the infrastructure.		

5.3.9 Trusted Execution Environment Management

5.3.9.1 Overview

Trusted Execution Environment (TEE) is an isolated processing environment in which services, tasks, and applications can be securely executed irrespective of the rest of the system. Hence, this technology is able to withstand usual software attacks and physical attack such as access-based and contention-based cache attacks. Regarding to their functionality, the TEEs is mainly made of hardware-based solutions, which guarantee a secure memory area, software-based solutions, which protect the kernel and operating system via robust cryptographic methods, and integrated hardware-software solutions. Therefore, TEE solutions ensure both secure isolations and allow remote code and data integrity attestation.

Since trust is an essential capability for 5GZORRO ecosystem, TEEs will enable the code and data loaded within these isolated environments to be protected with respect to integrity and confidentiality.

With respect to the functional view of the 5GZORRO architecture (Chapter 5.2), two main types of TEE are applicable:

1) *Intra-domain TEE*, in which the functional modules that are related to the communication inside a domain are executed securely through the presence of a software-based TEE (Chapter 11.4.5). Additionally, the software-based TEE allows the function modules to exchange data in a protected manner. Each Domain also includes an untrusted area, which has modules that are not executed inside a TEE. The interactions between the intra-domain TEE and the untrusted area of a Domain takes place through secure internal channels to avoid a potential compromise of intra-domain TEE.
2) *Federated TEE*, in which the TEE is protecting:
    a. The execution of dedicated 5GZORRO platform application and services that are available for each Domain, such as the DLT service.
    b. The communication between different domains through tunnelling mechanisms, such as VPN connections, that prevent eavesdropping from malicious entities.

In particular for the architectural view of Chapter 5.2, a potential TEE deployment would be to include the zero-touch management and orchestration functions, the catalogue and the resource/service trading functions inside the intra-domain TEE. This TEE can communicate with the applications and services of the untrusted area (described in Chapter 11.4.5) as well as with the second TEE type i.e. federated TEE. In turn, the federated TEE can leverage the information that are originating from the intra-domain TEE, to perform internal functionalities as well as for interactions with other Domains.

5.3.9.2 Provided Services

The TEE Management service gives to 5GZORRO platform stakeholders the option to set up a trustworthy communication, where stakeholders can launch their tasks or services in a safe and reliable manner, as well as complete this process at any time. The following table depicts the principal TEE intra-domain services.

Table 5-12: Definition of Trusted Execution Environment Management service (domain level)

Service name: **TEE Capabilities Management**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Create TEE connection	**O**	This capability initializes and configures the connection with the available or selected TEE.
Execute command in TEE	**O**	This capability allows to execute commands in the TEE once a connection has been established.
Delete TEE connection	**O**	This capability deletes the connection with the TEE, deleting all the data left in the environment.
Notes		
None.		

5.3.10 Communication Fabrics

5.3.10.1 Overview

The communication fabric provides a set of services that make possible the integration and interoperation of the 5GZORRO services, facilitating the flexibility to allow closed loops automation across domains. The communication fabric allows cross-domain access to the 5GZORRO services' exposure and communication, using management data transport (streaming data to subscribers), management service exposure (service publication, discovery, consumption) and management service control (e.g. authorization, access control).

The cross-domain communication fabric facilitates the reachability of cross-domain services and the access of cross-domain endpoints. This also includes services for the communication between 5GZORRO services, which facilitates the provisioning of metadata to authorized consumers who require them.

5.3.10.2 Provided Services

The provided services by the Communication Fabric component are directly inherited from the ETSI ZSM reference architecture [23] and comprises this set of mandatory capabilities:

- **Service registration**: Manage (create, read, update, delete, list) registrations of 5GZORRO services.
- **Service discovery**: Manage the requests for registered services, providing access to their capabilities and notifications about potential changes in the location or availability of the services.
- **Communication** service: Enables communication between two or more services using a set of communication channels.

5.3.11 Network Slice and Service Orchestration

5.3.11.1 Overview

The Network Slice and Service Orchestration is responsible for the deployment of network slice instances, together with the orchestration of network services composing such network slices, in order to provide the associated communication services. This functional block provides a service that can be offered and consumed at the two different levels considered in 5GZORRO (intra and inter-domain).

5.3.11.2 Provided Services

The Network Slice and Service Orchestration service allows users to create and maintain network slice instances via traditional LCM operations. By using this service, users can establish a logical network that uses infrastructure resources contained inside a single domain or that expands across multiple administrative domains. This function is also responsible for the instantiation, run-time operation and orchestration of services that rely on the abstracted view of the underlying infrastructure that slices provide.

Taking into account the diverse nature of stakeholders targeted in 5GZORRO, in the cross-domain scope two scenarios are contemplated: i) slicing setup with 3rd party resources and ii) slicing setup with 3rd party orchestrated services. The former refers to the case in which a slice creation/extension is done using leased resources (VNF resources included) but the orchestration is only handled by the purchaser domain; while the latter targets the composition of cross-domain slices by concatenating services offered by different domains, each one with its own orchestration deployment.

More details about the service capabilities at domain and cross-domain levels are outlined in Table 5-13 and Table 5-14, respectively.

Table 5-13: Definition of network slice and service orchestration service (domain level)

Service name: **Per-domain Network Slice and Service Orchestration**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Manage slice lifecycle	**M**	Manage the lifecycle of slices that are deployed within the domain. The slice composition in terms of resource technologies is determined by the infrastructure of the considered domain. This capability supports traditional LCM operations (i.e. creation, modification and removal).
Manage services lifecycle	**M**	Manage the lifecycle of network services that are deployed over a given slice within the domain. This capability supports service-specific LCM operations, including instantiation, configuration, scaling and removal.
Provide slice(s) information	**M**	Provide on-demand information regarding slice description (including hosted NFs) and operational status for one or several slices.
Provide notifications about slice changes	**M**	Keep updated information in the data lake about lifecycle changes of slices (including changes on hosted NFs).

Notes
none

Table 5-14: Definition of network slice and service orchestration service (cross-domain level)

Service name: **Per-domain Network Slice and Service Orchestration**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Manage slice lifecycle	**M**	Manage the lifecycle of slices that span across multiple administrative domains by including infrastructure resources from 3rd party domains. This capability supports the same operations as its per-domain counterpart.
Manage services lifecycle	**M**	Manage the lifecycle of services that are deployed over infrastructure resources from 3rd party domains (in a cross-domain slice). This capability supports the same operations as its per-domain counterpart.
Coordinate slice setup with 3rd party orchestrated services	**M**	Interact with other network slice and service orchestration services to setup cross-domain slices containing resources and services orchestrated by 3rd party domains.
Provide slice(s) information	**M**	Provide on-demand information regarding slice description (including hosted NFs) and operational status for one or several slices.
Provide notifications about slice changes	**M**	Keep updated information in the data lake about lifecycle changes of slices (including changes on hosted NFs).
Notes		
none		

5.3.12 e-Licensing Management

5.3.12.1 Overview

The e-License management service is designed for the metric-based control of the proprietary Virtual Functions (VFs) in the different domains. The licensing agreements signed in the smart contracts that grants the use of the VF are translated in the licensing manager to pieces of software to observe the actions that are produced in the tracked VFs that are called watchers. Therefore, every action produced in each domain for every license-controlled VF will be tracked and evaluated for the licensing fulfilment.

5.3.12.2 Provided Services

In order to make possible this metric control, the licensing manager should request the agreement for each VF to evaluate the potential licensing terms associated and generate the licensing watchers lied to these licensing terms. Once the watcher tied to a VF is activated due to an instantiation, scale, termination, user connexion request, etc. The e-Licensing Manager should be capable to launch the action record request. For supporting these capabilities, it needs to expose the services in Table 5-15 and Table 5-16.

Table 5-15: Definition of e-Licensing Management service (domain level)

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

Notes
none

Table 5-16: Definition of e-Licensing Management service (cross-domain level)

Service name: **e-Licensing Management**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Licensing control trigger	**M**	Capability of the Network Slice and Service Orchestrator to notify the eLicensing Manager that licensing control needs to be checked in some software resources.
Action notification	**M**	Manage ACK/ERROR notifications of the licencing actions recording to the stakeholders.
Notes		
none		

5.3.13 Service & Resource Monitoring

5.3.13.1 Overview

Monitoring data is provided by each Operator for the resources and services controlled by that Operator. This data is stored in the Data Lake and is used to keep track of resource usage and to predict/track SLA violations. As monitoring data is provided over time, it is aggregated and made available in a suitable manner to perform the desired analytics. Monitoring data may be saved in either or both a private Data Lake and the cross-domain Data Lake.

5.3.13.2 Provided Services

The *Post Monitoring Data* service is called by Operators to store monitoring data in the Data Lake. The data must be channelled to *Aggregate Monitoring Data*, from where the results are eventually channelled to an SLA monitoring component.

Table 5-17: Definition of Service &Resource Monitoring service (cross-domain level)

Service name: **Service & Resource Monitoring**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Post Monitoring Data	**M**	Provide Monitoring data for selected resources and services to be saved in the Data Lake (and to be provided to the aggregator).
Aggregate Monitoring Data	**M**	Take monitoring data provided for some resource over time and aggregate it, saving result back in the Data Lake. This aggregated data must eventually find its way to an SLA monitoring component.
Notes		
none		

5.3.14 Intelligent SLA monitoring & breach prediction

5.3.14.1 Overview

This functional entity consists of two main services:

1. *SLA Monitoring* service which collects, and analyses aggregated monitoring data in order to detect violations in SLAs. The Marketplace (Section 5.3.1) can be notified about SLA violations and the appropriate smart contract is subsequently notified for re-calculating SLA status. This service will be

provided by the Smart Contracts Lifecycle Management functional block (Section 5.3.6) and works closely with the DLT and Smart Contracts to realise trustworthy SLA governance.

2. *SLA Breach Prediction* service which collects, and analyses aggregated monitoring data using AI techniques in order to predict possible breaches in SLAs and detect anomalies. The *Intelligent Network Slice and Service Orchestration* functional block (Section 5.3.15) could use this information in order to proactively allocate additional resources before the SLA violation to effectively prevent it. This service operates off-chain in the Data Lake.

Both services subscribe to receive aggregated monitoring data from other 5GZORRO functional entities, such as the Service and Resource Monitoring (Section 5.3.13). Then, the SLA Monitoring uses this data to detect SLA violations and inform the related smart contracts, while the SLA Breach Prediction uses historical and current aggregated monitoring data to predict SLA violations. Other 5GZORRO functional blocks, such as the Intelligent Network Slice and Service optimization (Section 5.3.15), can subscribe to the SLA Breach Prediction service for specific SLAs/Contracts in order to receive SLA breach predictions and anomaly detection indicators.

5.3.14.2 Provided Services

The 5GZORRO SLA Monitoring operates when a new contractual agreement is signed. Initially, it receives the SLAs of the new contract from the Marketplace (Section 5.3.1). Then, the SLAs of all the active contracts are kept internally and periodically the SLA Monitoring service requests resource metrics from the Monitoring Data Aggregator module (Section 5.3.13). Marketplace informs the service for any changes in SLAs (e.g. terminations, updates). It examines SLA metrics (SLIs) such as availability or response time by retrieving aggregated monitoring data representing the overall service levels from Monitoring Data Aggregator module. SLA Monitoring analyses the monitoring data and compares the metrics with the thresholds in SLAs in order to detect SLA violations. The Marketplace can subscribe to receive notifications about SLA violations and such notifications are subsequently propagated to the smart contract for the purposes of re-calculating SLA status. The types of considered violations are described in Section 5.3.6. Table 5-18 lists the SLA monitoring services, which are also part of the functionalities of the Smart Contracts Lifecycle Management.

Table 5-18: Definition of SLA Monitoring service (domain level)

Service name: **SLA Monitoring**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Start SLA Monitoring process	**M**	The service starts to receive and analyse resources monitoring data from the *Monitoring Data Aggregator* service for specific contract SLAs using AI techniques in order to detect violations in SLAs. SLA Monitoring process could be started by the marketplace after a new contractual agreement.
Update SLA Monitoring process	**M**	This capability is used when there is a need to change the characteristics of an SLA Monitoring Process. It can be used when the contracts SLAs have been changed and when there is a need to add/remove SLAs or modify Start/End Times.
Terminate SLA Monitoring process	**M**	This capability is used to stop receiving and analysing resources monitoring data for specific contract SLAs.
Return active SLA Monitoring Processes	**M**	This capability is used to create a list of active SLAs Monitoring Processes, their descriptions, their SLA violations and their subscribers.
Return SLA violations of a specific contract	**O**	This capability is used to return information on SLA violations of a specific contract in a specific time period. The information includes the SLOs that are violated, when these violations occurred, the durations of the violations and to what extent the SLOs are violated.

Return current values of SLA Monitoring parameters	**O**	This capability is used when a module should receive SLA Monitoring parameters values (SLIs) for specific contract.
Subscribe to SLA Monitoring Process	**M**	This capability is used when a module should start receiving SLA violations for specific SLAs/contracts.
Publish Notifications for SLA Violations	**M**	This capability is used to send to the subscribed modules SLA violations as they are occurring. The violations include the SLOs that are violated, when these violations occurred, the durations of the violations and to what extent the SLOs are violated.
Unsubscribe from SLA Monitoring Process	**M**	This capability is used to stop receiving notifications for SLA violations for specific SLAs/contracts.
Notes		
none		

The SLA Breach Prediction collects and analyses resource monitoring data from the Monitoring Data Aggregator module (Section 5.3.13) using AI techniques in order to predict possible breaches in SLAs and detect anomalies. Anomaly Detection refers to the problem of finding instances or patterns in data that deviate from normal behaviour. This service is critical because anomalies often indicate useful, critical, and actionable information (for example problems in the provisioning of a resource or an intrusion which exhausts a resource).

The SLA Breach Prediction service keeps the current status of all active SLAs and predicts whether the resource metrics violate the SLA parameter values as well as the exact time that this happens. 5GZORRO modules can subscribe to SLA Breach Prediction to different topics (different SLAs/Contracts) in order to receive estimations about when these SLAs could be violated. Then, the service notifies the subscribed modules with predictions for SLA violations.

The predictions include:

- SLA metrics that could be violated,
- exact time that these violations could occur,
- extent that the SLA metrics will be violated and
- level of certainty for SLAs violations.

The modules that are subscribed into the 5GZORRO SLA Breach Prediction can decide how much earlier they would like to be informed of the time of a possible SLA violation in order to be proactive in securing the additional resources that are needed to continue to comply with SLAs. Table 5-19 provides an overview of the SLA breach prediction services.

Table 5-19: Definition of SLA Breach Prediction service (cross-domain level)

Service name: **SLA Breach Prediction**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Start SLA Breach Prediction	**M**	The service starts to receive and analyse resources monitoring data from the *Monitoring Data Aggregator* service for specific contract SLAs using AI techniques in order to predict possible breaches in SLAs and detect anomalies. SLA Breach Prediction process could be started by the marketplace after a new contractual agreement or by the *Intelligent*

		Network Slice and Service Orchestration service in order to estimate the future needs for resources.
Update SLA Breach Prediction	**M**	This capability is used when there is a need to change the characteristics of an SLA Breach Prediction Process. It can be used when the contracts SLAs have been changed and when there is a need to add/remove SLAs or modify Start/End Times.
Terminate SLA Breach Prediction	**M**	This capability is used to stop receiving and analysing resources monitoring data for specific contract SLAs.
Return active SLA Breach Predictions	**M**	This capability is used to create a list of active SLA Breach Predictions, their descriptions, their breach predictions and their subscribers.
Subscribe to SLA Breach Prediction	**M**	This capability is used when a module should receive SLA Breach Predictions for specific SLAs/contracts.
Publish Notifications for SLA Breach Predictions	**M**	This capability is used to send to the subscribed modules predictions for SLA violations. The predictions include the SLOs that will be violated, when these violations will occur, to what extent the SLs will be violated and the level of certainty for SLAs violations.
Unsubscribe from SLA Breach Prediction	**M**	This capability is used to stop receiving notifications for SLA breach predictions for specific SLAs/contracts.
Configure Machine Learning Algorithms	**M**	Select and Configure the machine learning algorithms based on the types of SLA metrics to be monitored and the associated requirements.
Notes		
none		

5.3.15 Intelligent Network Slice and Service optimization

5.3.15.1 Overview

Intelligent Network Slice and Service optimization functional element receive monitoring data (e.g., resource utilization, SLA violations, etc.) on periodic time intervals from 'Service & Resource Monitoring' function element. This data is stored (i.e., within private data lake) for some time (e.g., two months) to perform intelligent data analysis using AI techniques/models (e.g., machine learning, deep learning) that can predict specific behaviours (e.g., low CPU/PRB resources in a VNF or network slice) before it can occur. . These AI models are evaluated regularly based on their predicted actions, and if needed, the models will be retrained.

5.3.15.2 Provided Services

Based on such data analysis, certain proactive actions (e.g., VNF or network slice scaling) will be triggered towards 'Network Slice and Service Orchestration' functional element which tries to perform those actions using resources from the same domain. If unsuccessful, 'Intelligent Network Slice and Service optimization' through 'Intelligent 3rd party resource selection' functional element discovers 3rd party resources from other domains and informs 'Network Slice and Service Orchestration' functional element to perform proactive multi-domain network slice scaling actions.

Table 5-20: Definition of Intelligent Network Slice and Service Orchestration service (domain level)

Service name: **Intelligent Network Slice and Service Orchestration**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Predict resource for network slice	**M**	Predict resource requirements of a network slice ahead of time using machine/deep learning models. The prediction accuracy depends on a number of factors including available datasets, chosen model hyper-parameters, etc.

		Notify the Network Slice and Service Orchestration functional element about the same.
Predict resource for network service	**M**	Predict resource requirements of a network service (composed of chained VNFs) ahead of time using machine/deep learning models. Notify the Network Slice and Service Orchestration functional element about the same.
Notes		
none		

Table 5-21: Definition of Intelligent Network Slice and Service Orchestration service (cross-domain level)

Service name: **Intelligent Network Slice and Service Orchestration**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**
Discover 3rd party resource initiation	**M**	Through Smart Resource and Service discovery function element, discover all 3rd party resources offered in the 5GZORRO architecture.
Select 3rd party resource initiation	**M**	Through Intelligent 3rd party resource selection functional element, select the best available 3rd party resource w.r.t cost and QoS among all the discovered resources.
Resource agreement setup initiation	**M**	Through Smart Contracts Lifecycle Management functional element, setup the proposed resource agreement via smart contract.
Proactive multi-domain network slice lifecycle initiation	**M**	Proactively notify Network Slice and Service Orchestration functional element to initiate multi-domain network slice extension to satisfy the network slice SLA.
Notes		
none		

5.3.16 Virtual Resource Management and Control

5.3.16.1 Overview

The Virtual Resource Management and Control functional element serves as a proxy between the Network Slice and Service Orchestration functional element and the domains NFV MANO and SDN controller components. It is responsible for controlling and managing the NFV infrastructure such as compute, storage, and network resources together with lifecycle management of VNFs in edge, core, and cloud domains.

5.3.16.2 Provided Services

Main services of Virtual Resource Management and Control functional element include allocation, release, upgrade of NFVI resources, maintaining a repository of NFVI hardware (compute, storage, networking) and software resources (hypervisors), managing software images, supporting VNF forwarding graphs by assigning virtual links, subnets, and ports, and collecting performance and fault monitoring data. Additionally, it also includes instantiation, scaling, updating, and termination of VNFs based on the monitored KPI data. Therefore, Virtual Resource Management and Control is also responsible for forwarding the collected monitoring data from their managed entities to the Intelligent Network Slice and Service Optimization functional element. The southbound interfaces of a Virtual Resource Management and Control interact with software-defined network controllers to carry out certain functions that are described before.

Table 5-22: Definition of Virtual Resource Management and Control service (domain level)

Service name: **Virtual Resource Management and Control**	Type: ***Per-domain***

Capabilities	Support (O|M)	Description
Manage NFVI resources and VNFs in Edge/cloud slice subnet	**M**	Manage the lifecycle of edge/cloud NFVI resources and VNFs deployed in a slice. This is done by interacting with VIM, VNFM and NFVO components of the domain.
Provide NFVI resource and VNF performance statistics	**M**	Push monitoring data about the NFVI resource usage and VNF performance/fault statistics to Private data lake within the Intelligent Network Slice and Service Optimization functional element.
Manage transport network slice subnet	**M**	Manage, configure, and optimize traffic flow control within the network according to the forwarding policies defined in the domains SDN controller.
Provide transport network statistics	**M**	Push monitoring data about the transport network to Private data lake within the Intelligent Network Slice and Service Optimization functional element.
Notes		
none		

5.3.17 Radio Resource Management & Control

5.3.17.1 Overview

The Radio Resource Management & Control entity is responsible for managing the shared radio resources offered in the 5GZORRO architecture. The radio resources encompass not only the shared licenced spectrum but also the RAN infrastructure. For a given service or network slice, the Radio Resource Management & Control associates a RAN slice to it, which includes the base stations among those available in the platform (Wi-Fi, LTE, 5G), to deploy in a given geographical area and the specific radio resources, including the operation frequency and bandwidth.

5.3.17.2 Provided Services

The Radio Resource Management & Control is responsible for the deployment and maintenance of RAN slice sub-nets, which includes the automatic configuration of the base stations and their coordination therein. This coordination requires a global scheduler, which distributes the traffic loads among slices based on SLA, base station load conditions or interference levels.

Diverse radio resource providers are expected to co-exist in the 5GZORRO architecture. However, each Radio Resource Management & Control will be responsible to configure the radio resources (infrastructure) within its own domain. Therefore, no cross-domain Resource Management & Control is assumed. In case a service of network slices requires to be mapped to more than one RAN slice, they will have to communicate with each domain Resource Management & Control independently.

More details about the Radio Resource Management & Control service capabilities at domain level are outlined in Table 5-23.

Table 5-23: Definition of Radio Resource Management & Control service (domain level)

Service name: **Radio Resource Management & Control**		Type: ***Per-domain***
Capabilities	**Support (O|M)**	**Description**
Manage RAN slice sub-nets	**M**	Manage the lifecycle of RAN slice sub-nets that are deployed within the domain. The slice composition in terms of resource technologies is determined by the SLA and infrastructure availability of the considered domain.

		This capability supports traditional LCM operations (i.e. creation, modification, adaptation, and removal)
Provide RAN slice sub-net statistics	**M**	Provide regular information regarding the status of a RAN slice sub-net. This information shall be stored in the data lake, although digested notifications or statistics can be sent to the RAN slice consumer
Notes		
none		

5.3.18 DLT platform

5.3.18.1 Overview

A key focus of 5GZORRO is to ensure that the Marketplace remains DLT agnostic by encapsulating a series of abstract interfaces described previously e.g. Smart Contract Lifecycle Management, Catalogue etc. DLT 'Drivers' will be developed for Corda to demonstrate the full capabilities of the 5GZORRO marketplace.

The Corda DLT platform facilitates the building of distributed applications (CorDapps) to extend the capabilities of the ledger, including the specification of inter-node flows, data types and smart contracts to verify transactions.

5.3.18.2 Provided Services

R3 Corda provides the following key services for realising the DLT network and building distributed applications atop [24]:

Table 5-24: Definition of Corda services (cross-domain level)

Service name: **Corda**		Type: **Cross-*domain***
Capabilities	**Support (O|M)**	**Description**
States	**M**	States are classes that encapsulate on-ledger facts. Over time, states are marked as consumed/historic as generated outputs from transactions are consumed as inputs to subsequent transactions. Each DLT node has a vault where it stores all states relevant to itself.
Contracts	**M**	A transaction is contractually valid if all its input and output states are acceptable according to the contract. Contracts are deterministic classes that encapsulate logic that perform verification over a proposed update to the ledger.
Flows	**M**	Flows are a means of programmatically defining multi-step, multi-party processes for agreeing a ledger update; flow steps can be paused/check-pointed and resumed. Communication between nodes is point-to-point and only occurs in the context of flows. Corda also provides built in flows for common tasks such as finalising a transaction with the notary.
Service Hub	**M**	An API accessible from Flows that provides an interface to node functions such as: • Network map cache – node information within the network • Identity service – for Identity resolution • State/transaction access • Contract upgrade services
Transactions	**M**	A transaction in Corda is a proposal to update the ledger. A transaction will be committed to the ledger if it doesn't contain double spends, is contractually valid and is signed by the required parties. Verification of transaction input and output states is performed by Smart Contracts and final checks for double spends, transaction timestamps and - optionally - transaction validation are

		performed by a notary pool, a cluster of nodes that provide the point of transaction finality in the system.
Identity	**M**	Provides services for resolving and exchanging both well-known and confidential identities to facilitate the data/transaction privacy needs of the application.
RPC Operations	**M**	A node's owner interacts with their node solely via Remove Procedure Calls (RPC) and does not have access to the afore-mentioned Service Hub. Key operations available are: • Vault querying • List and execute available flows on the node • Get node info • Get information about other network participants
Vault Querying	**M**	A node's Vault stores all the states pertinent to it. It has been built from the ground up to support proven query frameworks. There are various mechanisms for querying the vault, but a VaultService provides a flexible mechanism for querying the vault and satisfies most use cases.
Persistence	**M**	Transaction state stored in the Vault is indexed for the purposes of executing queries. However, Corda also offers the ability to expose some or all of a contact state to an Object Relational Mapper (ORM) to be persisted in a relational database. This gives rise to the potential for relational joins and hierarchical data structures.
Contract Constraints	**M**	Contract constraints serve to control and agree upon platform upgrades, whilst mitigating any attack vector that could be exploited by a bad actor.
Testing	**M**	A range of tooling is provided within the Corda ecosystem such that networks can be mocked and easily interrogated which makes contract & flow development and testing significantly easier.
Notes		
none		

5.3.19 Data Lake Platform

5.3.19.1 Overview

Using the services of a Data Lake usually includes several coordinating services, which can be thought of as a pipeline. In 5GZORRO, we have the following specific example:

1. Provide and store raw monitoring data,
2. Monitoring Data Aggregator,
3. Perform analytics to predict violation of SLA,
4. Perform actions upon detection of SLA issues.

The implementation and deployment of the stages of the pipeline must be coordinated. The output from one stage often serves as the input for another stage.

5.3.19.2 Provided Services

When an Operator joins cross-domain Data Lake services (by calling *Register Operator*), it must first register and authenticate. Upon registration, data storage location is designated for the Operator's monitoring data. The Data Lake must also set up the infrastructure to run a Monitoring Data Aggregator and SLA Breach Predictor (and perhaps other entities).

Table 5-25: Definition of Data Lake Platform service (cross-domain level)

Service name: **Data Lake Platform**		Type: ***Cross-domain***
Capabilities	**Support (O|M)**	**Description**

Register Operator	**M**	Operator registers with Data Lake. Data Lake designates location to store monitoring data and sets up other entities to perform monitoring data aggregation, SLA breach prediction, etc.
Notes		
none		

5.3.20 5G Network Virtualization Platform

5.3.20.1 Overview

The 5G Network Virtualization platform is the underlying abstraction for managing the virtualized infrastructure. This functional entity encompasses several infrastructure controllers that are employed in 5GZORRO in collaboration with Virtual Resource Management and Control and Radio Resource Management & Control blocks, without any design or development by the consortium beyond its integration in the platform.

The 5G Network Virtualization platform will offer a set of tools that will be particularly useful for slice sharing among several operator's infrastructure, activating and isolating resources on demand based upon requests. These tools are required to ensure the computing and storage of the requested resources and services in different domains, that could be potentially purchased in the 5GZORRO marketplace creating the also the transport network.

The infrastructure controllers foreseen at this stage are:

- The NFV-MANO: This block is responsible for the inter-domain NFV Management and Orchestration that requires the synergy of several functional blocks and collaborating through specified reference points. The NFVO has two main functions, the NS Orchestration (NSO) and the Resource Orchestration (RO), which implement the lifecycle management of the Network Functions that will compose the Network Slice Subnet Instances and the orchestration of the NFVI resources across multiple domain VIMs respectively.
- The Virtual infrastructure Manager (VIM) is responsible for controlling and managing the NFVI resources within a single domain, leveraging on hypervisors for the control of the computation/storage resources respectively.
- SDN controllers, that provides the networking between the instantiated resources that composes the shared slices and services.

At this stage, the decision about the supported tools in the 5G Virtualization Platform is under discussion. Regarding the cloud-native nature of the project and the maturity of the opensource tools available, the most popular valuable tools are Opensource MANO [113] as NFV-MANO, Kubernetes [105] as MANO/VIM, OpenStack[112] as VIM and ONOS[114] as SDN controller, but this decision will be taken in further iterations considering the developments in WP3 and WP4.

5.4 5GZORRO Information Elements

Some initial information elements for the 5GZORRO functional entities discussed above are detailed in this section. The presented models are not exhaustive of the entire set of data and internal elements which are needed to implement the related functionality. They rather aim at giving a preliminary indication for implementation of core information items to be considered. Further specification and formal representation of these information elements are expected to be included in future releases of the 5GZORRO reference architecture, in order to incorporate design and implementation decisions for the 5GZORRO Platform.

5.4.1 5GZORRO DIDs

As depicted in Section 11.4.2, Decentralized Identifiers (DIDs) are global and unique identifiers that do not need a centralized registration authority since they are created and/or recorded using cryptographic proofs. In spite of there are several DID methods available, 5GZORRO is going to generate and register a new method named "*5g-zorro*" in order to build a specific representation that contains the necessary attributes, properties, and claims. Besides, this new DID method will utilize the DLT deployed in the ecosystem which was specially designed to meet DIDs related requirements. Finally, DIDs also provide services that are a communication mechanism with the DID's owner (subject) such as discovery services or agent services. Nevertheless, it should be pointed out that 5GZORRO Services are different from DID Service since 5GZORRO Services are communication services offered to end-user customers that are built on top of 5GZORRO Resources.

The 5GZORRO DID syntax is compliant with W3C DID syntax by using a unique method name like "5g-zorro". As such, an example for a 5GZORRO DID for some 5GZORRO resource traded in the Marketplace would look like: ***"did:5g-zorro:123456789abcdefghi".***

5.4.1.1 5GZORRO DID Subjects

The following 5GZORRO DID Subjects have been identified:

Table 5-26: 5GZORRO DID Subjects

Subject type	description
5GZORRO Stakeholder	The legal entity that operates 5GZORRO administrative domains as defined in D2.1 including Resource Providers, Resource Consumers, Regulators, ...
5GZORRO Platform	The software system implementing 5GZORRO functionalities e.g. the Marketplace node, that has is deployed and operated by a 5GZORRO stakeholder.
5GZORRO Resource	The available resource or set of resources managed by a Resource Provider through the 5GZORRO Framework which is/are deployed and enabled to be traded in the Marketplace.
5GZORRO Service	A 5G Service or set of 5G Services ready for utilizing in the Marketplace and which is/are delivered by a 5G Service Provider.
5GZORRO Stakeholder Individual	An Individual working for a 5GZORRO Stakeholder e.g. in the Legal department to define the legal terms to be used in the Marketplace Smart Contracts.
5GZORRO Business Agreement	The Business Relationship among 5GZORRO Stakeholders to support the trade of 5GZORRO Resources and Services in the 5GZORRO Marketplace with specific SLA requirements that are controlled with DLT Smart Contracts.

5.4.1.2 5GZORRO DID Documents

Associated to 5GZORRO DIDs the following DID Documents have been identified:

Table 5-27: 5GZORRO DID Documents

Document type	Subject	Description	Services	Verifiable Credentials
Stakeholder Document	Stakeholder	Metadata about 5GZORRO stakeholder including associated services and verification methods used to authenticate the stakeholder, to verify	Identity and Permissions Management	stakeholderVC, governanceVC

		stakeholder assertions like the stakeholder role and to give permissions to the stakeholder *(capabilityInvocation property),* e.g. to publish offers in the marketplace. To be generated from the Governance Portal.		
Platform Document	Platform	Metadata about 5GZORRO platform including all deployed functionalities with associated versions. To be generated from the deployment process	Identity and Permissions Management	platformVC
Resource Document	Resource	Metadata about 5GZORRO platform including Resource type. For security reasons, most service data are only available in its associated Verifiable Credential. To be generated by the Resource Management Control or other component managing the Resource Offer life-cycle.	Identity and Permissions Management, (Virtual / Radio) Resource Management & Control	resourceVC
Service Document	Service	Metadata about 5GZORRO service but, for security reasons, most service data is only available in its associated Verifiable Credential. To be generated by the Network Slice and Service Orchestration or other component managing the Service Offer life-cycle.	Identity and Permissions Management, Network Slice and Service Orchestration	serviceVC
Stakeholder Individual Document	Stakeholder Individual	Metadata about 5GZORRO stakeholder individual including associated services and verification methods used to authenticate the individual, to verify individual assertions (e.g. name, role) and to give permissions to the stakeholder *(capabilityInvocation property)* e.g. agreement negotiation.	Identity and Permissions Management	
Business Agreement Document	Business Agreement	Metadata about 5GZORRO Business Agreement. For security reasons, most data are only available in its associated Verifiable Credential. To be generated by Smart Contract Life-cycle management.	Identity and Permissions Management, Smart Contract Life-cycle Manager, Service and Resource	governanceVC, agreementVC

	Monitoring Service

5.4.1.3 Verifiable Credentials

Regarding Verifiable Credentials (VC) [56], which were briefly explained in Section 11.4.2, we need to determine criteria in order to decide what type of data are declared in the DID Document or in the VC. One criterion could be visibility due to VCs would store private data, that includes more sensitive information, whilst the DID Document would register public attributes, parameters, or properties. Another criterion could be accessibility since a DID Document can be retrieved from the Registry / Blockchain as soon as you know its DID without having to contact its controller/agent while the VC can only be retrieved from the Holder Agent. Additionally, the Verifiable Credentials also has a data model, just like DIDs, with specific syntax and sematic defined by the W3C standard.

The following Verifiable Claims have been identified for 5GZORRO:

Table 5-28: 5GZORRO Verifiable Claims

Name	Claims Description	Holder / Subject	Issuer	Verifier
stakeholderVC	Claims about 5GZORRO operational software packages including the different plataformVCs previously issued at deployment time. Claim about type of 5GZORRO role including Regulator, Resource Provider, Resource Consumer, Service Provider, and Service Consumers. Claims about Resource types to be provided or consumed	5GZORRO stakeholder	5GZORRO Bootstrap Agent	Marketplace Governance
platformVC	Claims about the different 5GZORRO operational software packages including its endpoints.	5GZORRO Platform	DID User-Agent	Marketplace Governance
governanceDecisionVC	Claim about a certain governance decision including. ▪ To accept or reject a new 5GZORRO member.	5GZORRO stakeholder or 5GZORRO Business Agreement	Marketplace DID UA	zero-touch Service Management and Orchestration DID UA

	▪ Resolutions about SLAs disputes			
resourceVC	Claim about rights over a certain 5GZORRO Resource or set of 5GZORRO Resources. Claim that Resource is deployed and ready to be consumed by consumers and a certain behavior is expected. Claim that data generated by the Resource namely monitoring data, is trustworthy	5GZORRO Resource	zero-touch Service Management and Orchestration DID UA	Marketplace Identity Management
serviceVC	Claim that service is deployed and ready to be consumed by service consumers and a certain behavior is expected. Claim that service can be extended to other domains, by defining descriptors to be used and VNF registries end-points to be used from where images can be downloaded and deployed in the new domain.	5GZORRO Service	zero-touch Service Management and Orchestration DID UA	Marketplace Identity Management
agreementVC	Claim that identify the stakeholders involved in the agreement and the different roles played by each other including the identification of Resources and Services provided in the context of the agreement.	5GZORRO Business Agreement	Marketplace DID UA	zero-touch Service Management and Orchestration DID UA + Analytics and Intelligence DID UA

5.4.1.4 Proposed Structure for DID Documents

In order to clarify what is the common structure of a DID Document, we are going to declare a generic schema which contains mandatory and optional properties:

5GZORRO DID Document

```
{
   "@context": [
       "https://www.w3.org/ns/did/v1",
            "https://w3id.org/security/v1"
    ], //MANDATORY
   "id": "", //MANDATORY
   "controller": "", //OPTIONAL
   "service": {
         "id": "",
         "type": "",
         "serviceEndpoint": "",
         }
   }, //OPTIONAL
   "verificationMethod": [{
         "id": "",
         "type": "",
         "controller": ""
  }], //OPTIONAL
   "authentication": [{
         "id": "",
         "type": "",
         "controller": "",
  }], //OPTIONAL
   "assertionMethod": [{
         "id": "",
         "type": "",
         "controller": "",
  }], //OPTIONAL
         "keyAgreement": [{
         "id": "",
         "type": "",
         "controller": "",
  }], //OPTIONAL
         "capabilityInvocation": [{
         "id": "",
         "type": "",
         "controller": "",
  }], //OPTIONAL
         "capabilityDelegation": [{
         "id": "",
         "type": "",
         "controller": "",
  }] //OPTIONAL
}
```


Examples of DID Document

So far six possible DID Document templates have been identified in 5GZORRO, one for **stakeholder document**, one for **platform document**, one for **resource document, one for service document, one for stakeholder individual document, and one for business agreement document.** Now, we show an example of how DID Document templates apply to stakeholder documents. The functionality of some properties should be highlighted, such as the ***id*** of the DID Subject, the ***controller*** that identifies the authorized entity who can make changes to DID Document, the ***service*** that is related to the available Functional Entities (see the previous Services table), the ***verificationMethod*** that provides a set of methods in order to check the relationship between DID subject and DID Document, the ***authentication*** to prove that an entity it is also the DID subject, the ***assertionMethod*** indicates that a statement was performed by a DID Subject, the ***keyAggrement*** is utilized to express a verification relationship which an entity can use to engage in key agreement protocols on behalf of the DID subject, the ***capabilityInvocation*** is used to invoke capabilities as the DID subject, and finally, the ***capabilityDelegation*** is utilized to grant capabilities as the DID subject or on behalf of the DID subject to other capability invokers.

5GZORRO DID Document

```
{
   "@context": [
       "https://www.w3.org/ns/did/v1",
           "https://w3id.org/security/v1"
    ],
   "id": "did:5g-zorro:123456789abcdefghi",
   "controller": "did:5g-zorro:123456789abcdefghi",
   "service": {
         "id": "did:5g-zorro:123456789abcdefghi#notification1",
         "type": "NotificationService",
         "serviceEndpoint": "https://example.5g/notifications/stakeholder#1"
   },
   "verificationMethod": [{
         "id": "did:5g-zorro:123456789abcdefghi#key1",
         "type": "VerifiableCredentialService",
         "controller": "https://example.5g/verifiablecredential/key#1"
  }],
   "authentication": [
                 //Option 1, this method can be used to authenticate as did:...fghi
                 "did:example:123456789abcdefghi#keys-1",
         {
                 //Option 2, this method is only authorized for authentication, it may
not
                 // be used for any other proof purpose, so its full description is
embedded
                 // in controller URL rather than using only a reference.
         "id": "did:5g-zorro:123456789abcdefghi#key2",
         "type": "VerifiableCredentialService",
         "controller": "https://example.5g/verifiablecredential/key#2"
  }],
   "assertionMethod": [{
         "id": "did:5g-zorro:123456789abcdefghi#key2",
         "type": "VerifiableCredentialService",
         "controller": "https://example.5g/verifiablecredential/key#2"
  }],
         "keyAgreement": [{
         "id": "did:5g-zorro:123456789abcdefghi#key2",
         "type": "VerifiableCredentialService",
         "controller": "https://example.5g/verifiablecredential/key#2"
  }],
         "capabilityInvocation": [{
         "id": "did:5g-zorro:9876543210abcdefghij#key3",
         "type": "VerifiableCredentialService",
         "controller": "https://example.5g/verifiablecredential/key#3"
  }],
         "capabilityDelegation": [{
         "id": "did:5g-zorro:9876543210abcdefghij#key3",
         "type": "VerifiableCredentialService",
         "controller": "https://example.5g/verifiablecredential/key#3"
  }]
}
```

Proposed Structure for Verifiable Credentials

In this subsection, the basic concepts associated with the 5GZORRO structure of Verifiable Credentials will be described as a combination of mandatory attributes, additional attributes, and optional attributes, all of them associated with each Verifiable Credential template. In addition, a few examples of Verifiable Credentials are provided.

Verifiable Credentials may be expressed using JSON. A VC is typically composed of the following properties: the ***context*** of the VC, the ***id*** is an identifier that others must use in order to express statements covered by this verifiable credential, the ***type*** means the kind of information represented in the VC, the ***credentialSubject*** represent who is the subject or subjects of claims, the ***issuer*** is the person or entity that allocates this VC, the ***issuanceDate*** indicates when VC was emitted and validated, the ***expirationDate*** depicts

when VC will not be considered useful, the ***credentialStatus*** is utilized for knowing information about the current status of a VC, the ***termOfUse*** depicts the conditions under a VC was issued, and it is normally used by an issuer or a holder in order to communicate them, the ***nonTransferable*** , and finally, the ***proof*** contains all details necessary to assess this VC is authentic.

The proposed structure for Verifiable Credential is as follows:

5GZORRO VerifiableCredential

```
{
   "@context": [
       "https://www.w3.org/2018/credentials/v1",
       "https://www.w3.org/2018/credentials/examples/v1"
    ], //MANDATORY
   "id": "", //MANDATORY
   "type": "", //MANDATORY
   "credentialSubject": {
         "id": "",
         "5GZORROResource": [{
                "id": "",
                "type": "",
                "platformId": ""]
         }
   }, //MANDATORY
   "issuer": {
         "id": "",
         "name": ""  }, //MANDATORY
   "issuanceDate": "",  //MANDATORY
   "expirationDate": "", //MANDATORY
   "CredentialStatus": {
        "id": "",
        "type": ""
        }, //MANDATORY
   "termsOfUse": "", //OPTIONAL
   "nonTransferable": "", //OPTIONAL
   "proof": {
        "type": "",
        "created": "",
        "proofPurpose": "",
        "verificationMethod": "",
                "jws": ""
        } //MANDATORY
}
```

5.4.2 General offer information model

As mentioned in section 3.2, in 5GZORRO we use the resource, service and product offer information models proposed by TMForum [108][109][110] as the reference models to describe the 5GZORRO assets available in the market. In TMForum a resource describes a traceable or inventoried asset, which in 5GZORRO will usually refer to the resource description (i.e. VNFD, Spectoken, NSD, etc). These resources are usually attached to services that in general terms describe how a resource can be instantiated, deployed or used. In 5GZORRO we use services to describe VNFs, network services and network slices, while RAN, spectrum and computing assets remain as simple TMForum resources. Both services and resources can be exposed to 3rd parties in order to be acquired, by means of product offers. In the resource information model, depicted in Figure 5-7 and Figure 5-9, a resource is made available to a catalogue using a ResourceCandidate and its associated ResourceSpecification, which defines common attributes and features. The information model of the service, depicted in Figure 5-8 and Figure 5-10, in a similar way to resources, uses a ServiceCandidate and a ServiceSpecification. The information model of the product offer is depicted in Figure 5-11.

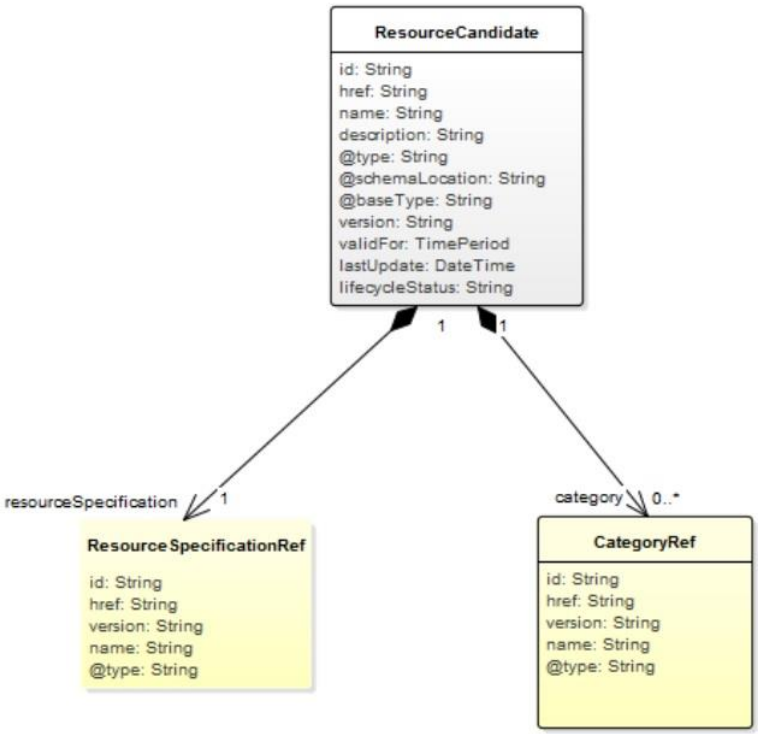

Figure 5-7: TMForum ResourceCandidate information model [108]

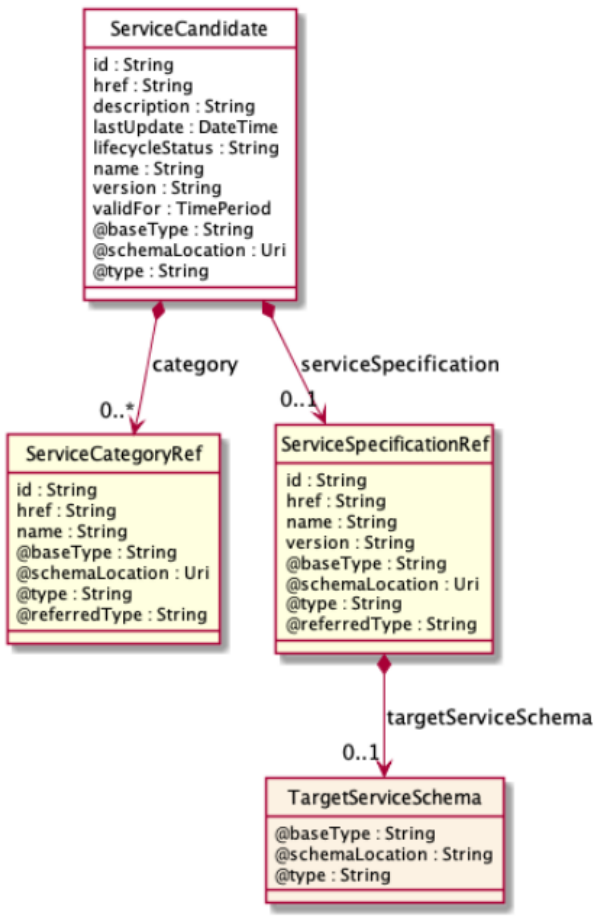

Figure 5-8: TMF ServiceCandidate information model [109]

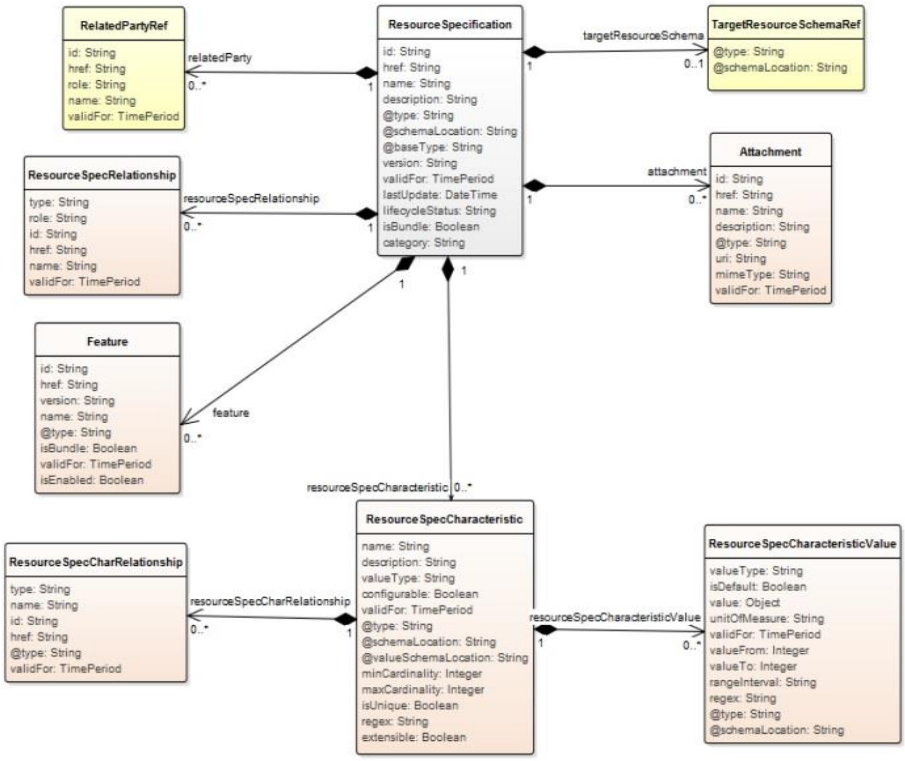

Figure 5-9: TMF ResourceSpecification information model [108]

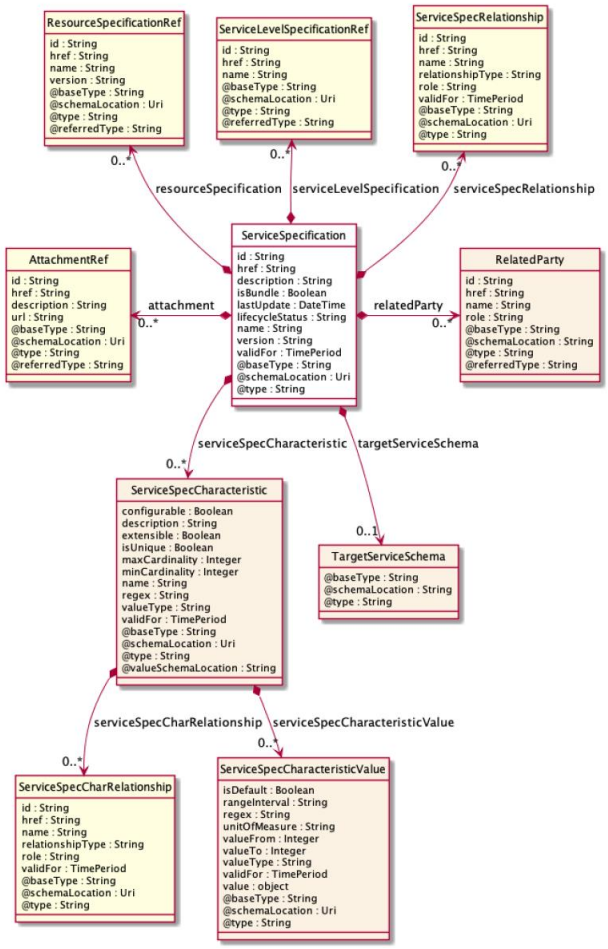

Figure 5-10: TMF ServiceSpecification information model [109]

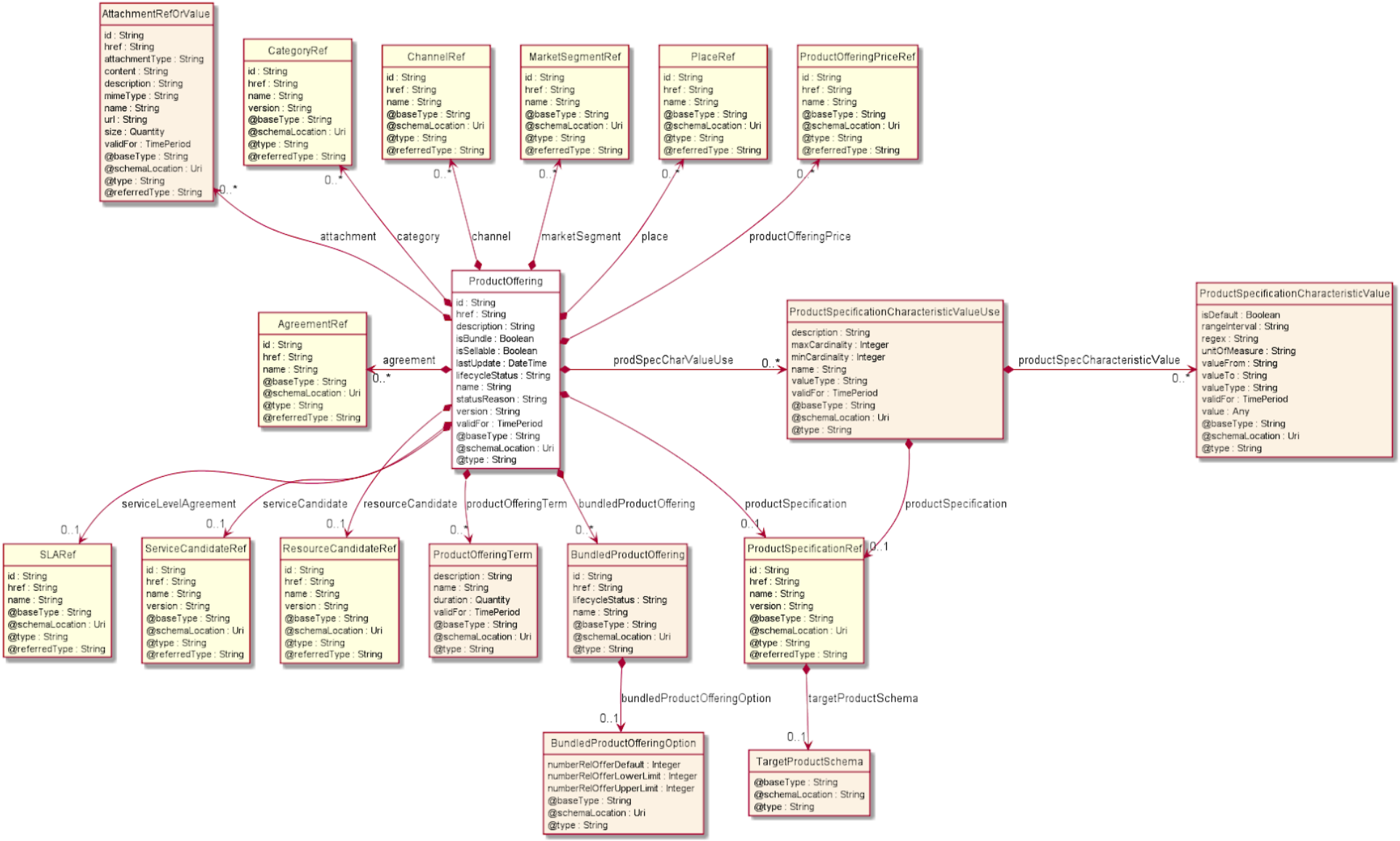

Figure 5-11: TMForum product offering information model [110]

The following subsections detail the specific extensions to the models for the different 5GZORRO resources.

5.4.3 Spectrum (licensed & non-licensed) offer information model

The spectrum offer information model is used to indicate the physical frequency resources and the geographical location of a licensed spectrum sharing offer. Table 5-29 and Table 5-30 address the main parameters of the Spectrum ResourceCandidate and ResourceSpecification IM, respectively, and their fundamental fields. Table 5-31 and Table 5-32 define the resource specification characteristics of a spectrum resource.

Table 5-29: Spectrum resource ResourceCandidate Information Model

Parameter	Type	Description
id	String	Unique identifier in the catalogue
href	DID	Distributed identifier of the spectrum resource candidate
name	String	A name given to the spectrum resource candidate
resourceSpecification	A list of ResourceSpecificationRef [108] elements	A reference to the spectrum resource ResourceSpecification (see Table 5-30)

Table 5-30: Spectrum resource ResourceSpecification Information Model

Parameter	Type	Description
id	String	Unique Service Specification id in the catalogue
href	DID	Distributed identifier of the spectrum resource specification
name	String	The name of the spectrum resource specification
resourceSpecCharacteristic	List of ResourceSpecCharacteristic [108]	Description of the spectrum resource key features that are relevant for the definition of the applicable spectrum range, i.e. pairs of central frequency and bandwidth Information Models (see Table 5-31 and Table 5-32)

Table 5-31: Central frequency ResourceSpecCharacterstic Information Model

Parameter	Type	Description
name	String	centralFrequency
description	String	The operation central frequency of the spectrum resource
configurable	Boolean	false
validFor	Time period [108]	Date of expiration of the current description
isUnique	Boolean	true
resourceSpecCharacteristicValue	List of objects	A list of central frequency values
valueType	String	Numeric
value	Frequency object	A central frequency value
unitOfMeasure	String	Megahertz [MHz]

Table 5-32: Bandwidth ResourceSpecCharacterstic Information Model

Parameter	Type	Description
name	String	bandwidth
description	String	The bandwidth of the spectrum resource
configurable	Boolean	false
validFor	Time period [108]	Date of expiration of the current description
@type	String	Bandwidth class type
isUnique	Boolean	true
resourceSpecCharacteristicValue	List of objects	A list of possible bandwidth values
valueType	String	Numeric
value	Frequency object	Bandwidth value
unitOfMeasure	String	Megahertz [MHz]

Leveraged by the recently-introduced resource candidates and specifications Information Models, spectrum offers are created to expose the licensed spectrum resources and to be shared by the SRP within the 5GZORRO platform. Relevant fields of the *ProductOffering Information Model* [110] used for spectrum resources are shown in Table 5-33.

Table 5-33: Spectrum resource Product Offering Information Model

Parameter	Type	Description
id	String	Unique internal id for the spectrum resource offer in the catalogue
href	DID	Distributed identifier of the spectrum resource offer
Name	String	Name of the spectrum resource offer
agreement	List of AgreementRef [110]	Agreements related to the spectrum offer (e.g. pricing agreements, etc.)
place	List of PlacRef [110]	Geographical location for which the spectrum resource offer is valid
productOfferingPrice	List of ProductOfferingPriceRef [110]	Reference to the pricing models available for the spectrum resource offer
resourceCandidate	ResourceCandidateRef [110]	Reference to the spectrum resource candidate (see Table 5-29)
productOfferingTerm	List of productOfferingTerm [110]	Business conditions in which the spectrum resource offer is valid
serviceLevelAgreement	List of SLARef [110]	Service Level Agreements available for the spectrum resource offer

5.4.4 RAN (active & passive) offer information model

The RAN offer information element is used to indicate the RAN infrastructure elements the RAN infrastructure provider is sharing in the 5GZORRO architecture. Table 5-34 and Table 5-35 address the main parameters of the RAN *ResourceCandidate* and *ResourceSpecification* IM, respectively, and their fundamental fields. Table 5-32, Table 5-36 and Table 5-37 define the resource specification characteristics of a RAN element.

Table 5-34: RAN ResourceCandidate Information Model

Parameter	Type	Description
id	String	Unique identifier in the catalogue
href	DID	Distributed identifier of the RAN resource candidate

name	String	A name given to the RAN resource candidate
resourceSpecification	A list of ResourceSpecificationRef [108] elements	A reference to the RAN *ResourceSpecification* (see Table 5-35)

Table 5-35: RAN ResourceSpecification Information Model

Parameter	Type	Description
id	String	Unique Service Specification id in the catalogue
href	DID	Distributed identifier of the spectrum resource specification
name	String	The name of the spectrum resource specification
resourceSpecCharacteristic	A list of ResourceSpecCharacterstic [108] elements	Description of the RAN resource parameters. Each RAN resource specification characteristics contains multiple values to show the possible configurations. This list can only contain the following characteristics: • *operationBand* (Table 5-36) • *bandwidth* (Table 5-32) • *quota* (Table 5-37)

Table 5-36: Operation band ResourceSpecCharacterstic Information Model

Parameter	Type	Description
name	String	operationBand
description	String	In case of a cellular base station or Wi-Fi access point, the supported operation band (3GPP) or channel (Wi-Fi). Will contain as many values as supported
configurable	Boolean	false
validFor	Time period [108]	Date of expiration of the current description
@type	String	RAN operation band class
isUnique	Boolean	true
resourceSpecCharacteristicValue	List of objects	A list of operation band values
valueType	String	Numeric
value	Frequency object	The band/channel number
unitOfMeasure	String	Integer

Table 5-37: Quota ResourceSpecCharacterstic Information Model

Parameter	Type	Description
name	String	quota
description	String	The percentage of the passive resources shared (e.g. backhaul link capacity, baseband processing capacity, etc.)
configurable	Boolean	true

validFor	Time period [108]	Date of expiration of the current quota description
@type	String	RAN quota class
isUnique	Boolean	true
resourceSpecCharacteristicValue	List of objects	A list of quota values
valueType	String	Numeric
value	Frequency object	Quota value
unitOfMeasure	String	Percentage [%]

5.4.5 Edge/Core Cloud resources (Iaas, PaaS) offer information model

This section defines the cloud resources information model which will be used to onboard in the 5GZORRO marketplace, making explicit the cloud physical and virtual characteristics allowing the trade with this kind of resource in the 5GZORRO architecture.

Table 5-38: IaaS Information Model

Parameter	Type	Description
id	String	Internal id in the catalogue
href	DID	DID of the resource owner
href	DID	DID of the resource
resourceName	String	Name of the resource
productOfferingPrice	List of ProductOfferingPriceRef [110]	Pricing models available for the offer
productOfferingTerm	List of productOfferingTerm [110]	Terms of the offer: Duration, conditions, etc
resourceType	String	"core"/"edge"
locationRef	List of PlacRef [110]	Geographic placement of the cloud resource
resourcePhysicalCapabilities	List of objects	A list of operation band values
cloudId	String	Unique identifier of the cloud
datacenterId	String	Unique identifier of the datacenter
nodeId	String	Unique identifier of the node
hardwareCapabilities	List of objects	Descriptor of HW capabilities: GPU, SDD storage, FPGA, ASIC, NIC, high performance network uplinks, etc…
hardwareCapKey	String	Type of capability
hardwareCapValue	float	Value of capability
hardwareCapUnit	String	Unit of measure: Gb, MHz, etc…
hardwareQuota	float	Quota of offered resource
SLARef	List of SLARef [110]	Service level agreement associated reference
resourceVirtualCapabilities	List of objects	A list of operation band values
cloudId	String	Unique identifier of the cloud
datacenterId	String	Unique identifier of the datacenter
nodeId	String	Unique identifier of the node
isMaster	Boolean	Identifies the master or worker node
Type	String	"openstack"/"kubernetes"/ "openshift"/etc
SLARef	List of SLARef [110]	Service level agreement associated reference

5.4.6 VNF/CNF offer information model

The VNF or CNF resource is described in the resource catalogue, according to [108], using a ResourceCandidate and a ResourceSpecification. The resourceCandidate defines the DID and name of the resource, as shown in Table 5-39: VNF/CNF ResourceCandidate Information Model, and the ResourceSpecification is used as describe in Table 5-40.

Table 5-39: VNF/CNF ResourceCandidate Information Model

Parameter	Type	Description
id	String	Unique identifier in the catalogue (e.g. UUID)
href	DID	Distributed identifier of the VNF/CNF resource
name	String	The name of the VNF/CNF resource
resourceSpecification	ResourceSpecificationRef [108]	A reference to the VNF/CNF ResourceSpecification

Table 5-40: VNF/CNF ResourceSpecification Information Model

Parameter	Type	Description
id	String	Unique identifier in the catalogue (e.g. UUID)
href	DID	Distributed identifier of the VNF/CNF package
name	String	The name of the VNF/CNF package
resourceSpecCharacteristic	List of ResourceSpecCharacteristic [108]	Description of characteristic features of a resource specification, in this case VNF/CNF parameters (e.g. package format, descriptor format, see Table 5-41)

The *ResourceSpecCharacteristic* should be used to define additional features to the *ResouceSpecification*. An example of a *ResourceSpecCharacteristic* for VNF/CNF *ResourceSpecification*, following the information model of *ResourceSpecCharacteristic* in [108], is described in Table 5-41.

Table 5-41: VNF/CNF ResourceSpecCharacteristic & resourceSpecCharacteristicValue example

Parameter	Description	Example Value
name	This field contains the name of the specification	PackageFormat
description	This field describe the content of the specification	The VNF package structure format
configurable	This field specifies if the value is configurable for the specification	False
isUnique	This field specifies if the value is unique for the specification	False
resourceSpecCharacteristicValue		
valueType	A type of value of the characteristic	String
value	The value of the characteristic	CSAR

In case of VNF and CNF, the resource is associated to a service, defined through a *serviceCandidate* and a *ServiceSpecification*, which are used as described in Table 5-42 and in Table 5-43. Similar to

resourceSpecCharacteristic, *serviceSpecCharacteristic* should be used to define additional features to the *serviceSpecification*.

Table 5-42: VNF/CNF ServiceCandidate Information Model

Parameter	Type	Description
id	String	Unique identifier in the catalogue
href	DID	Distributed identifier of the VNF/CNF service
name	String	The name of the VNF/CNF service
serviceSpecification	A list of ServiceSpecificationRef [109]	A reference to the VNF/CNF ServiceSpecification

Table 5-43: VNF/CNF ServiceSpecification Information Model

Parameter	Type	Description
id	String	Unique Service Specification id in the catalogue
href	DID	Distributed identifier of the service specification
name	String	The name of the service
resourceSpecificationRef	List of ResourceSpecRef [108]	A list of references to the VNF/CNF resource specification available with this service specification
serviceSpecCharacteristic	List of ServiceSpecCharacteristic [109]	Description of characteristic features of a service specification such as VNF/CNF developer, VNF/CNF owner, and other configurable parameters of the VNF/CNF

The information model of the product offer is described in Table 5-44, according to the general product offer proposed in [110].

Table 5-44: VNF/CNF product offer information model

Parameter	Type	Description
id	String	Unique internal id for the VNF/CNF offer in the catalogue
href	DID	Distributed identifier of the VNF/CNF offer
Name	String	Name of the VNF/CNF offer
agreement	List of AgreementRef [110]	Agreements of the VNF/CNF provider with Network Operators, VNF/CNF certifications, special VNF pricing agreements
place	List of PlacRef [110]	Geographical places for which the VNF/CNF offer is valid
productOfferingPrice	List of ProductOfferingPriceRef [110]	Reference to the pricing models available for the VNF/CNF product offer.
serviceCandidate	ServiceCandidateRef [110]	Reference to the VNF/CNF service candidate (see Table 5-42)
productOfferingTerm	List of productOfferingTerm [110]	Business conditions in for which the VNF/CNF offer is valid
serviceLevelAgreement	List of SLARef [110]	Service Level Agreements available for the VNF/CNF offer

5.4.7 Network Slice and Network Service Offer information model

As previously mentioned, offers for network services and network slices are modelled in a standardised manner based on the TMForum Information Framework. Following the TMForum service categories, network slices in 5GZORRO are mapped to Resource Facing Services (RFS) that reference to infrastructure resource objects (modelling compute or RAN elements) required to establish the considered network slice.

On the other hand, network services are abstracted as Customer Facing Services (CFS) that can be acquired and consumed by other 3rd party domains. These service entities are supported by RFS including network slices and VNF/CNF services.

Table 5-45 and Table 5-46 outline the main parameters of a network slice in the service catalogue [109], via the *ServiceCandidate* and the *ServiceSpecification* IM, which is adopted as (reusable) core building block to assemble service catalogue and inventory entries in the 5GZORRO marketplace. Similarly, Table 5-47 and Table 5-48 depict the *ServiceCandidate* and the *ServiceSpecification* of network services, respectively.

Table 5-45: Network slice ServiceCandidate Information Model

Parameter	Type	Description
id	String	Unique identifier in the catalogue
href	DID	Distributed identifier of the network slice candidate
name	String	The name of the network slice candidate
serviceSpecification	A list of ServiceSpecificationRef [109]	A reference to the network slice ServiceSpecification (see Table 5-46)

Table 5-46: Network slice ServiceSpecification Information Model

Parameter	Type	Description
id	String	Unique Service Specification id in the catalogue
href	DID	Distributed identifier of the network slice specification
name	String	The name of the network slice specification
resourceSpecificationRef	List of ResourceSpecRef [108]	A list of references to the associated infrastructure (Edge, Cloud and/or RAN elements) resource specifications.
serviceSpecCharacteristic	List of ServiceSpecCharacteristic [109]	Description of the network slice characteristics representing key features that are relevant for network services to be supported by the slice. Note that attributes here included (values, configurability, etc.) are inherited from the resources specifications associated to the slice.

Table 5-47: Network service ServiceCandidate Information Model

Parameter	Type	Description
id	String	Unique identifier in the catalogue
href	DID	Distributed identifier of the network service candidate
name	String	The name of the network service candidate
serviceSpecification	A list of ServiceSpecificationRef [109]	A reference to the network service ServiceSpecification (see Table 5-48)

Table 5-48: Network service ServiceSpecification Information Model

Parameter	Type	Description
id	String	Unique Service Specification id in the catalogue
href	DID	Distributed identifier of the network service specification
name	String	The name of the network service specification
serviceSpecCharacteristic	List of ServiceSpecCharacteristic [109]	Description of the network service characteristics representing key features that are relevant for customers obtaining this service via product offers.

By using the aforementioned service candidates and specifications, product offers are created to expose network slices and network services available for discovery and purchase in the 5GZORRO Marketplace. Relevant fields of the *ProductOffering* Information Model [110] used for network slices and network services offers are shown in Table 5-49 and Table 5-50, respectively.

Table 5-49: Network slice ProductOffering information model

Parameter	Type	Description
id	String	Unique internal id for the network slice offer in the catalogue
href	DID	Distributed identifier of the network slice offer
Name	String	Name of the network slice offer
agreement	List of AgreementRef [110]	Agreements of the network slice provider with network slice customers, for instance, service providers
place	List of PlacRef [110]	Geographical places for which the network slice offer is valid
productOfferingPrice	List of ProductOfferingPriceRef [110]	Reference to the pricing models available for the network slice offer
serviceCandidate	ServiceCandidateRef [110]	Reference to the network slice service candidate (see Table 5-45)
productOfferingTerm	List of productOfferingTerm [110]	Business conditions in which the network slice offer is valid
serviceLevelAgreement	List of SLARef [110]	Service Level Agreements available for the network slice offer

Table 5-50: Network service ProductOffering information model

Parameter	Type	Description
id	String	Unique internal id for the network service offer in the catalogue
href	DID	Distributed identifier of the network service offer
Name	String	Name of the network service offer
agreement	List of AgreementRef [110]	Agreements of the network service provider with network service customers

place	List of PlacRef [110]	Geographical places for which the network service offer is valid
productOfferingPrice	List of ProductOfferingPriceRef [110]	Reference to the pricing models available for the network service product offer
serviceCandidate	ServiceCandidateRef [110]	Reference to the network service candidate (see Table 5-47)
productOfferingTerm	List of productOfferingTerm [110]	Business conditions in for which the network service offer is valid
serviceLevelAgreement	List of SLARef [110]	Service Level Agreements available for the network service offer

5.4.8 Smart Contract information model

In the tables below we define smart contract information models associated with the creation and management of Resource/Service offers and Agreements.

5.4.8.1 Resource & Service Offers

Resource and Service Offers will be governed by Smart Contracts. Below summarises the general operations, events and information model.

Table 5-51: Generic Offer Smart Contract

Parameter	Description
Id	DLT specific Identifier
Owner	Owning Party (Resource/Service Provider)
Owner HREF	DID of Resource/Service Provider
Allocated To	Consuming Party (Resource/Service Consumer)
Allocated To HREF	DID of Resource/Service Consumer
HREF	DID of the Resource/Service Offer
Offer Terms HREF	Location of legal prose document
Offer Terms Hash	Hash of legal prose document to support non-repudiation
Licence Terms HREF	Location of licence terms document
Licence Terms Hash	Hash of the licence terms document to support non-repudiation
Status	The status of the offer e.g. available, reserved, in use etc.

A Resource Offer will extend the generalised 'Offer' Smart Contract summarised above to encapsulate logic/properties specific to certain categories of resource e.g. Edge Resources, Spectrum etc.

Service Offers will extend the more general 'Offer' contract to encapsulate one or more resources as per the below. Operations and Events related to

Table 5-52: Service Offer Smart Contract (extends General Offer SC)

Parameter	Description
Resource Offers	Map of comprised Resource Offer (reference).

For each offer, available operations and events are:

- **Operations**
  - *Service Offer*
    - Register
    - Update
    - Remove
    - Allocate
    - De-allocate
    - Add Resource Offer
    - Remove Resource Offer
  - *Resource Offers*
    - Register
    - Update
    - Remove
    - Allocate

        - De-allocate
- **Events**
    - Created
    - Updated
    - Removed
    - Allocated
    - De-allocated
    - Resource offer Added (Service Offer only)
    - Resource offer Removed (Service Offer only)

5.4.8.2 Agreement Smart Contract

Having agreed terms between Provider and Consumer, a smart contract agreement will be deployed that encapsulates the agreement on the DLT. The table below summarises the operations, events and information model expected for this contract.

Table 5-53: Agreement Smart Contract

Parameter	Description
Id	DLT Specific Identifier
Provider	Party
Provider HREF	DID of Resource/Service Provider
Consumer	Party
Consumer HREF	DID of Resource/Service Consumer
Monitoring Service HREF	DID of the monitoring service agreed between provider & consumer to provide SLO measurements
Agreement Terms HREF	Location of legal prose document
Agreement Hash	Hash of legal prose document to support non-repudiation
Status	A flag to indicate the status of the agreement e.g. Created, Active, Reconciled, Terminated etc.
Start Date/Time	Agreement Start Date
End Date/Time	Agreement End Date
Resource Offers	Map of Resource Offers that comprise the agreement
Service Offers	Map of Service Offers that comprise the agreement
SLOs	Mapping of SLOs that comprise the contract – based on the Resource Offers
Pricing	Agreed pricing for the contract term

For each offer, available operations and events are:

- **Operations**
    - Create/Deploy
    - Update
    - Terminate
    - Add SLO
    - Remove SLO
    - Submit Licensing Action
    - Get Status (I.e. status of all SLOs)
- **Events**
    - Created
    - Updated
    - Terminated
    - Completed
    - Licensing Action Result

5.4.8.3 SLO Smart Contract

Smart contracts will be developed to manage the registration of SLOs against an agreement, receive a and verify a violation claim from the SLA Manager.

Table 5-54: SLO Smart Contract

Parameter	Description
Id	SLO Identifier
Parent Identifier	Identifier of the parent SLA contract
Resource/Service Offer	DID

Frequency	The temporal frequency on which a measurement should be taken
Penalty Map	A Map of any Penalties (SC's) that have been associated with the SLO

A Smart Contract will be created for each type of concrete measurement (SLO) to manage the specifics of the measurement and how violations are to be handled. Example instances might be Uptime. Average Response Time, Throughput etc. each having properties to record the current status of the SLO and logic to re-calculate violation/threshold status on receiving new measurements.

For each offer, available operations and events are:

- **Operations**
  - Create
  - Add Penalty
  - Remove Penalty
  - Post violation claim
  - Get Status (e.g. valid, violated)
- **Events**
  - Added
  - Violated

5.4.8.4 Penalty Smart Contract

Should a violation occur, an SLO may have one or more Penalties associated with it that encapsulate the compensation rules as defined in the agreement.

Table 5-55: Penalty Smart Contract

Parameter	Description
Id	Penalty Identifier
Compensator	Identifier of the party being penalised
Compensator HREF	DID of the compensating party
Compensatee	Identifier of the party being compensated
Compensatee HREF	DID of the party being compensated
Type	The type of penalty e.g. Credit | Debit
Unit	The unit of remuneration e.g. Service Credits | GBP | EUR etc.
Amount	The number of units to be remunerated

Table 5-56: Penalty Smart Contract

Parameter	Description
Violating Party	Party in violation
Recipient Party	The party to be remunerated in-line with the SC
Type	Credit | Debit
Penalty Unit	The unit of the penalty e.g. GBP, Service Credits
Penalty Value	The number of units to be Credited/Debited from the concerned party

6 Operational patterns

In this section, some relevant operations patterns for the 5GZORRO architecture are introduced, which highlight the role and service of the different Functional Entities defined in Section 5.

6.1 Resource Provider Onboarding in 5GZORRO marketplace

The Onboarding of new stakeholders in the 5GZORRO marketplace enables new Resource Providers and Service Providers to be enrolled into the 5GZORRO eco-system and begin trading resources or services, with other 5GZORRO Marketplace members. In order to proceed, the candidate has to deploy the 5GZORRO framework and to generate the required certificates (W3C VerifiableClaims) that will be used by the consortium governance model to take a decision about the new member candidate. Once onboarded the new member can begin advertising/consuming resources/services based on their assigned roles & permissions.

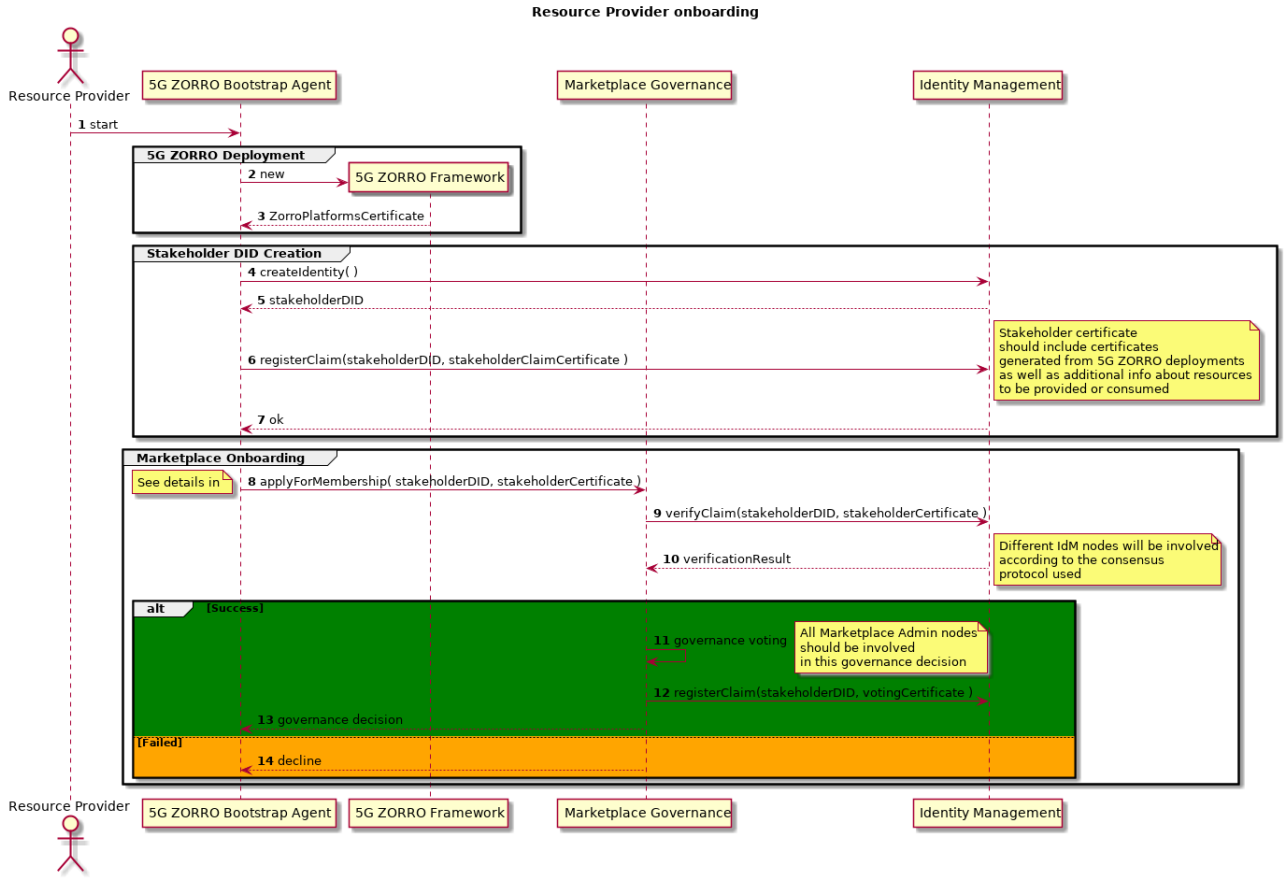

Figure 6-1: Resource Provider Onboarding in 5GZORRO marketplace Operational Pattern

Step 1 to 7: The Resource Provider executes the 5GZORRO Bootstrap Agent (e.g. script) that should perform all required steps to have the new Stakeholder onboarded in the Marketplace with minimum human intervention. The Bootstrap Agent deploys the 5GZORRO sub-systems and if successful, the deployment process automatically generates associated DIDs and registers Certifiable Claims that should include the different endpoints addresses.

Step 8 to 11: The Bootstrap Agent creates Resource Provider's 5GZORRO stakeholder DID and registers at the Identity Management the *stakeholderCertificate* that should include previously generated 5GZORRO *platformCertificates* plus additional information about resources to be provided or consumed in the Marketplace.

Step 12 to 18: The Bootstrap Agent applies to be a Member of 5GZORRO Marketplace passing its *stakeholderDID*. The Marketplace Governance 1st verifies if the Resource Provider has successfully deployed the 5GZORRO framework and then initiates the process to take a decision to accept or reject the request according to the Marketplace Governance Model. As soon as a decision is taken a new Verifiable Claim is generated and registered in the IdM with all the data about the decision. If accepted this claim is a stakeholder certificate that defines all permissions granted to the new 5GZORRO Resource Provider. The Governance decision is notified to the candidate by using the 5GZORRO notification service.

6.2 Publishing a Spectoken Resource Offer

This operational pattern describes the sequence of operations involved in the creation of a spectoken. It is worth noting that, in our vision, spectokens are referred to a set of spectrum resources in a licensed band. Radio transmissions over a licensed frequency band is restricted primarily to the spectrum license holder, typically a national operator which won a spectrum license following a spectrum auction or beauty contest organised by the National Regulatory Authority (NRA). The licensed spectrum owner might seek to share to lease underutilized spectrum either to enhance his position in the market or as a result of obligations imposed by the NRA. The 5GZORRO architecture provides fast, reliable, and secure spectoken trading among the Spectrum Resource Provider (SRP) and the Spectrum Resource Consumer (SRC). Spectrum trading cannot be opaque to the NRA, which must acknowledge the radio resources to be shared.

Before the SRP publishes a spectrum offer in the 5GZORRO Marketplace, the NRA must announce the SRP's spectrum capabilities by issuing a spectrum certificate. This certificate determines that the SRP is the original owner of the licensed spectrum bands included in the certificate. Typically, SRPs gained their operational privileges after a spectrum management procedure with the Regulator (spectrum auction or beauty contest).

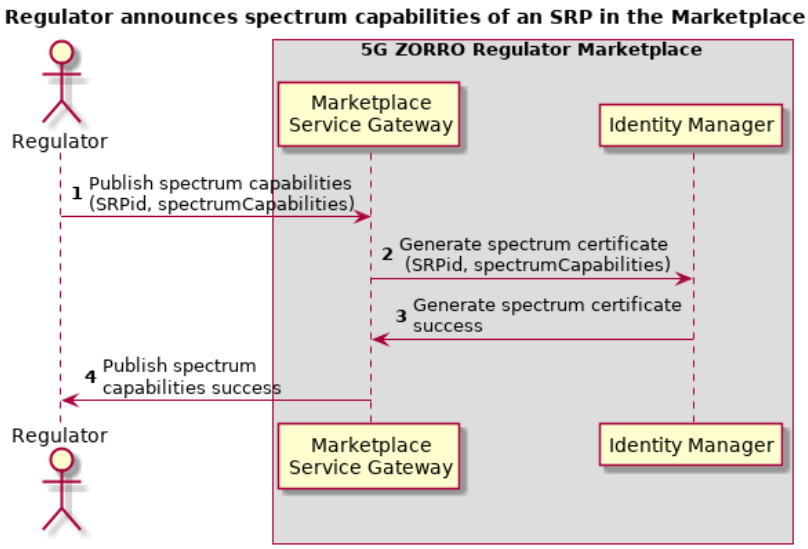

Figure 6-2: Spectrum certificate generation workflow

Figure 6-2: Spectrum certificate generation workflow illustrates the messages involved in the generation of the spectrum certificates by the Regulator. Firstly, the Regulator wants to generate a spectrum certificate relating an SRP to its spectrum capabilities in terms of frequency bands and available bandwidth and geographical area coverage. Then, the Marketplace Service GW sends a request to the Identity Manager to generate the spectrum certificate for the SRP with the given characteristics. The Identity Manager acknowledges the creation of the spectrum certificate to the Marketplace Service GW, which notifies the Regulator that the spectrum certificate has been issued and registered.

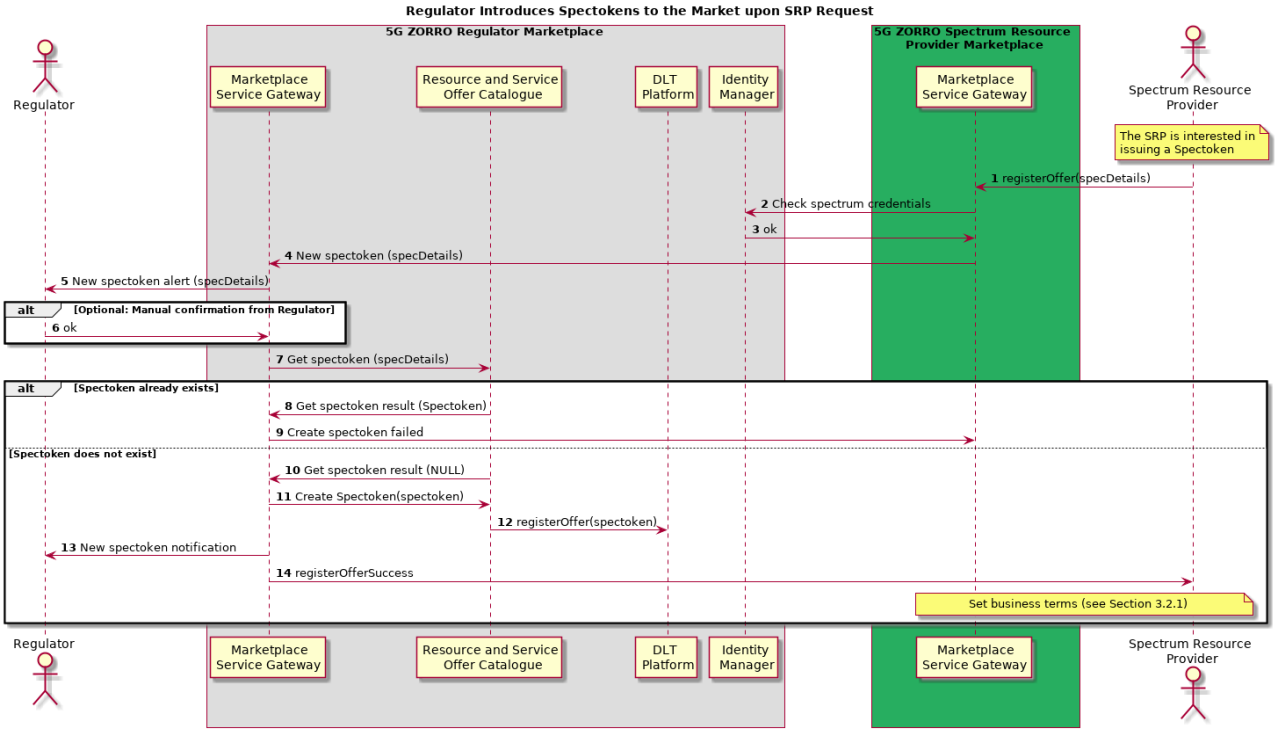

Figure 6-3: Spectoken Resource Offer Publishing workflow

Figure 6-3 illustrates the workflow to request the issuance of a spectoken offer in the 5GZORRO marketplace.

Step 1: The SRP decides to share a certain a subset of its licenced spectrum resources in a given geographical area. The SRP enters the 5GZORRO marketplace portal (UI) in its domain and enters the spectrum offer information (e.g. central frequency, bandwidth, price, etc.)

Step 2: The Marketplace GW request the Identity Manager to validate if the spectrum capabilities of the SRP (spectrum certificate) is compatible with the spectrum details of the spectoken to be generated

Step 3: Assuming that the SRP owns the claimed spectrum, the Identity Manager will confirm so and the spectoken generation procedure starts

Step 4: The Regulator receives a request to create a spectoken with the spectrum offer details provided by the SRP in step 1

Step 5: The Regulator gets a notification that an SRP is asking for a spectoken to be created

Step 6: Optionally, the spectoken generation may require some elevated validation by the Regulator. Once done, and if the SRP claim on the spectrum is correct, the Regulator can trigger the generation of the spectoken

Step 7: The Regulator must check that a similar spectoken for the same radio resources and geographical area does not exist. To do this, the Regulator sends a get spectoken request to the 5GZORRO Catalogue with the spectrum information

Step 8: If the Catalogue answers with a spectoken, the generation of the new spectoken cannot continue

Step 9: If the spectoken creation fails, the Marketplace GW notifies the SRP that a spectoken for a similar frequency range or geographical area already exists

Step 10: But in case no duplicity in the spectoken is found, the Catalogue answers the message in 8. with no spectoken information

Step 11: The Regulator issues the spectoken with the Catalogue

Step 12: The Catalogue registers the spectoken in the DLT.

Step 13: Once the spectoken creation and registration has been concluded, the Regulator is notified

Step 14: Then, the SRP is also notified that the spectoken creation and registration has been concluded

Finally, the generated spectoken is associated with a smart contract. Note that the SRP may have decided to put the spectoken for sale in the market (providing a price for it) or, instead, the SRP may have decided to operate in the shared spectrum and have the spectoken for its own. Whichever the SRP's choice is, the smart contract will determine who the owner of the spectoken is: no one (for sale) or the SRP (not selling).

6.3 Trustworthy Resource Discovery

Discovery will be facilitated by a Resource Catalogue and Service Catalogue services, each exposing endpoints for querying resources and services respectively and supporting both simple and smart selection methods. Requesters shall be able to make requests based on 'simple' criteria, simply querying the catalogue's off-chain storage for available resources or services meeting a basic set of requirements. The requester will then be able to apply any domain-specific intelligence they wish to further filter results. For more complex, 'smart' queries, the catalogue will utilise a Smart Discovery module provided by Cross Domain Monitoring and Analytics. Results will be filtered/prioritised using smart selection techniques based on machine learning. Catalogue data and metrics will be provided by the public data lake for smart selection to be based on.

Resources and services will be universally identifiable and trustfully discoverable by means of Distributed Identifiers (DIDs) and Verifiable Claims incorporating associated cryptographic material, verification methods and service endpoints, allowing the provider (controller) to prove control over the resource and present any associated claims, and third-party statements to discover and validate endpoints.

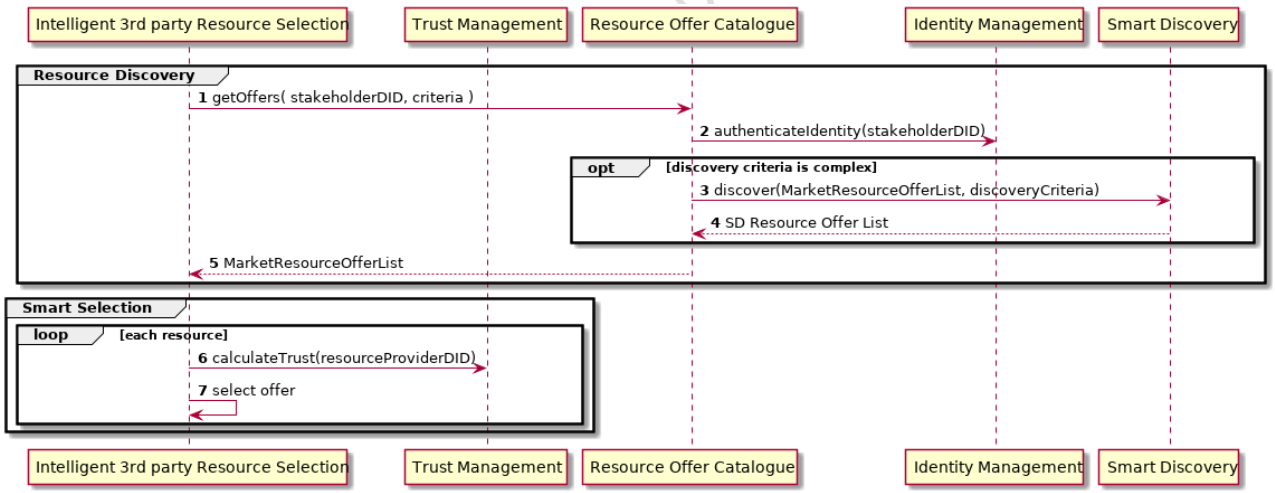

Figure 6-4: Trustworthy Resource Discovery workflow

Steps 1-5: The Resource Offer Catalogue is queried by the Intelligent 3rd Party Resource Selection about available resources fulfilling certain criteria. Permissions to perform such query are validated by the Identity and Permissions Management functionality. Optionally, and in case the discovery criteria are complex, the Resource Catalogue is assisted by the machine learning powered algorithms from a Smart Discovery module.

Steps 6-7: The Intelligent 3rd Party Resource Selection consumer assesses trust on each offer to make a final selection by applying domain-specific intelligence.

6.4 Trustworthy Smart Contract Setup for spectrum

Communication Service Providers (CSPs) may have the ability to extend their radio coverage using the 5GZORRO platform. To that end they must acquire spectokens previously provided by Spectrum Resource Providers (SRPs) and currently available at the 5GZORRO marketplace.

The other entity involved is the Regulator, which must approve the ownership transaction of the spectokens between the Marketplace and the CSP. Figure 6-4 depicts the main steps required in this process described as follows.

Step 1: CSP authentication at the marketplace

Step 2: Indication of the requirement/claim, in this case to acquire spectokens

Steps 3 – 4: The verification of the claim must be conducted and authorized

Steps 5 - 7: The CSP issues a request to obtain the available spectrum at the defined location

Steps 8 – 11: The available resource offers are retrieved from the DLT with attached legal statements of the Smart Contracts

Steps 12 – 13: The CSP chooses a suitable resource offer and signs the terms of the Smart Contract with the intent to acquire spectokens

Steps 14 – 15: The Marketplace transmits to the Spectrum Resource Provider (SRP) the request from the CSP to acquire spectrum for authorization

Step 16: In cases when only is possible to request the Regulator's approval manually, the Marketplace registers the CSP intent to acquire the spectokens

Steps 17 – 18: The Regulator is notified to approve the CSP request and perform necessary arrangements to accommodate the required spectrum

Step 19: Once the request is approved by the Regulator, the spectokens become owned by the CSP

Step 20: In cases when an automatic approval is possible (e.g. The CSP agreed to all terms imposed by the Smart Contract), the Marketplace immediately issues the transaction of request spectokens to be owned by the CSP

Steps 21 – 22: Here, the regulator merely gets a notification about the spectokens transaction between the Marketplace and the CSP

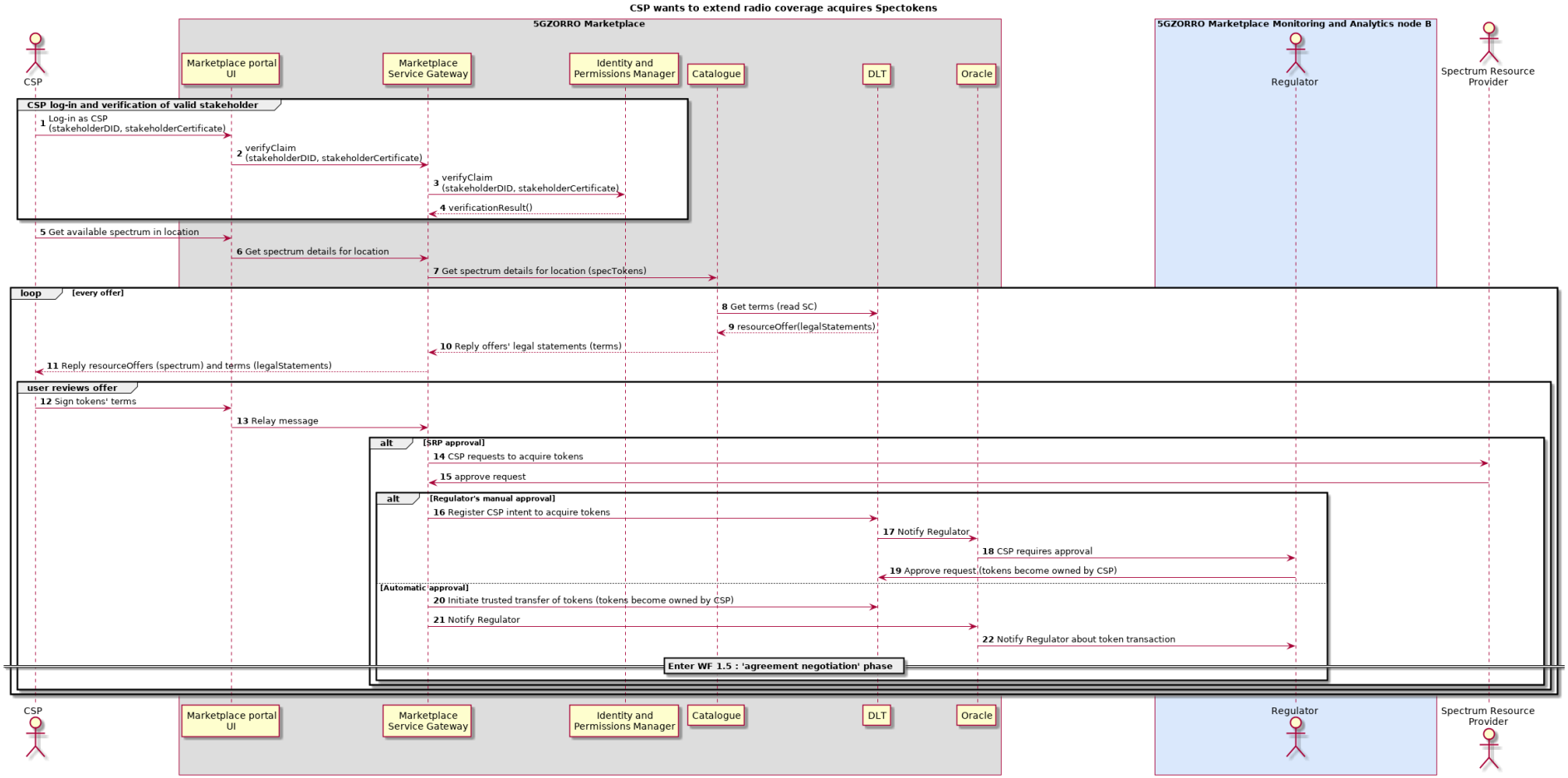

Figure 6-4: CSP who wants to extend radio coverage acquires spectokens

6.5 Trustworthy Smart Contract Setup for edge computing

The current operational pattern describes the processes involved in the automatic, trustworthy resource agreement setup, which is vital for completing the dynamic leasing of edge compute resources. As illustrated in Figure 6-5: Workflow for Trustworthy Resource Agreement Setup, the agreement is realised through an SLA Smart Contract. In this Figure, Domain A has already selected a portion of the resources offered by Domain B. This means that Domain B assumes the role of the Resource Provider, while Domain A the one of the Resource Consumer. The functional elements that participate in this process are the Resource and Service Offer Catalogue, the Smart Contract Lifecycle Management, the Intelligent Network Slice and Service Optimization, the Virtual Resource Management and Control, and the DLT Platform, which are described in Chapters 5.3.2, 5.3.6, 5.3.15, 5.3.16, and 5.3.18, respectively.

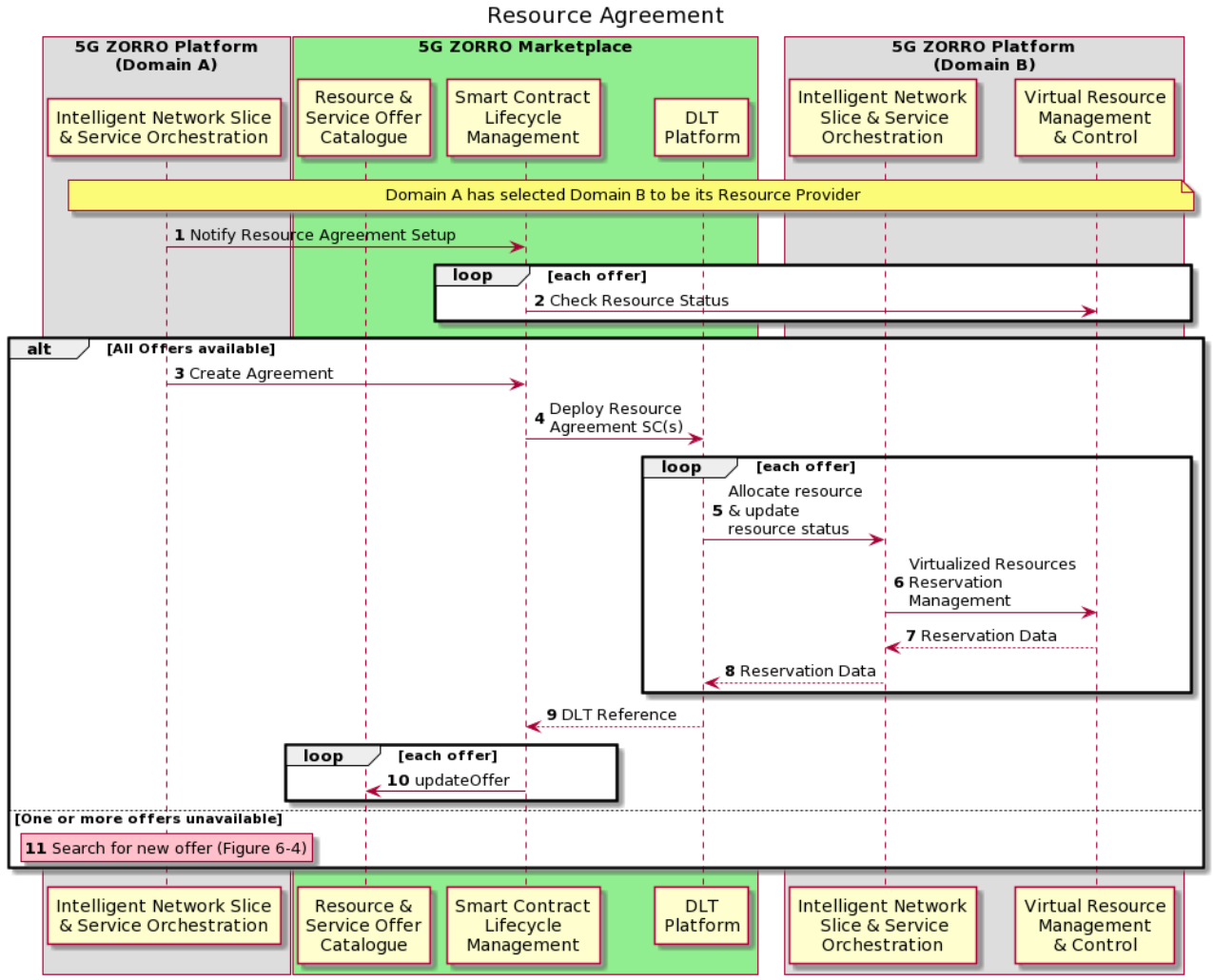

Figure 6-5: Workflow for Trustworthy Resource Agreement Setup

More specifically, the steps presented in Figure 6-5 can be analysed as follows:

Step 1: The *Intelligent Network Slice and Service Orchestration* module of Domain A (Domain Intelligence A) proposes an agreement for the selected resources to the Marketplace, through the *Smart Contract Lifecycle Management*.

Step 2: Following this request, the *Smart Contract Lifecycle Management* module queries the relevant *Virtual Resource Management and Control* in Domain B about resource availability. The queries' results should be returned to Domain Intelligence A.

Steps 3 – 10: If all resources are available then the contract negotiation can begin. In these steps, the agreement is finalized through the Marketplace, the Smart Contracts are signed from involved parties and consensus is reached at the DLT network (steps 3 – 4). Then, the provider's *Intelligent Network Slice and Service Orchestration* is informed, through Oracles, about each resource request and it updates the requested resource status by giving appropriate orders to the *Virtual Resource Management and Control* module (steps 5 – 8). Similarly, each resource must get updated in the Catalogue, too, in order to reflect its new status (steps 9 – 10) and the agreement status is returned to Domain A.

Step 11: In case that one or more requested resources are no longer available from Domain B, the resource selection process must be repeated as explained in Section 0.

Once the resource agreement is successfully established, Domain A is able to leverage on resources provisioned by Domain B. As a consequence, an end user that was initially served by Domain A, may be redirected to Domain B, according to load balancing policies, and get served by the second domain Figure 6-6 describes the redirection of a User Equipment (UE) to the Resource Provider's Domain. The new functional elements that appear in the next diagram are the *Network Slice and Service Orchestration* and the *Service and Resource Monitoring*, which are described in Chapters 5.3.11 and 5.3.13, respectively.

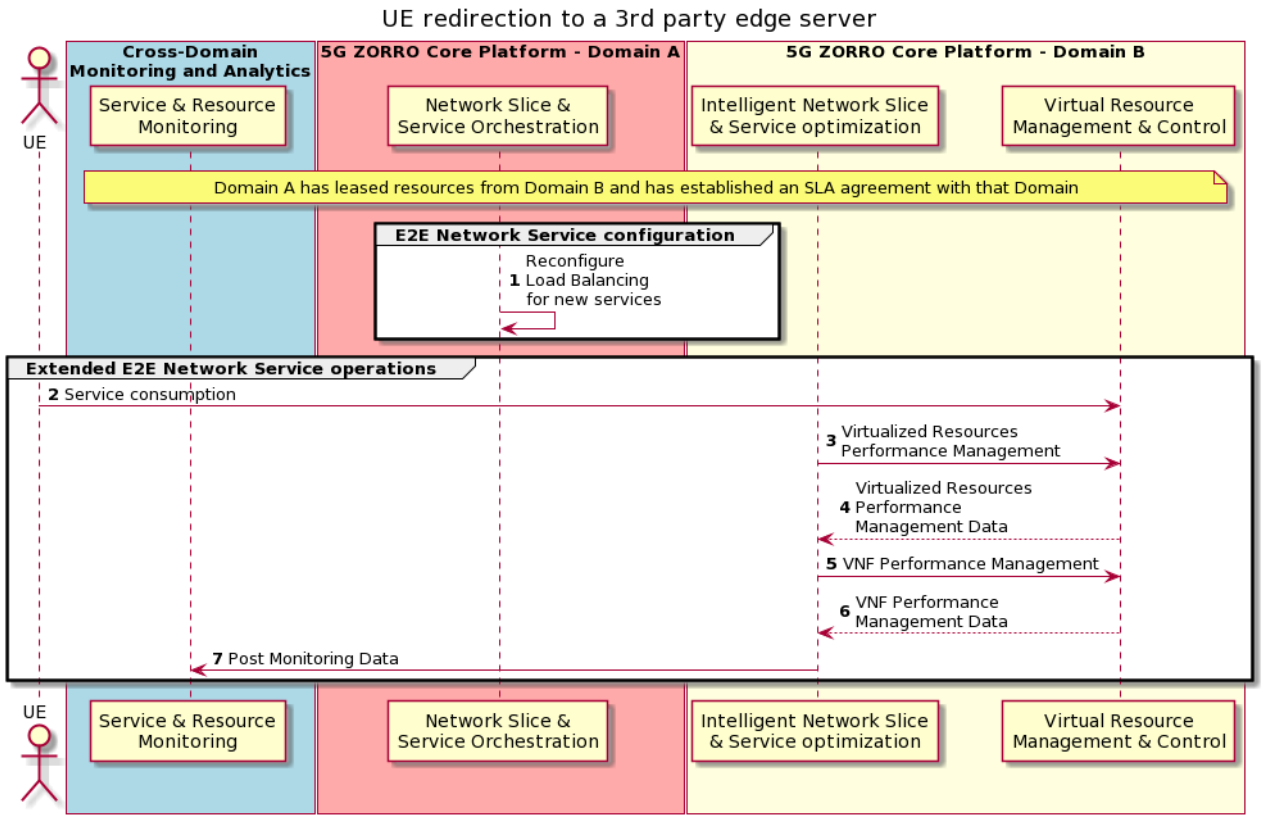

Figure 6-6: Workflow for UE redirection to a 3rd party edge server

More analytically, Figure 6-6 illustrates the case where a User Equipment (UE) had been using a service provided by Domain A, when Domain A predicted the need for additional resources. In this example it is assumed that Domain A couldn't allocate resources from its own infrastructure. Thus, Domain A searched for 3rd party resources and completed the Smart Contract establishment and the Slice extension towards Domain B's resources. This is the status at the beginning of the workflow in Figure 6-6. Then, based on load balancing decisions, the UE may be redirected to the 3rd party edge server and get served by Domain B. The steps of the sequence diagram are the following:

Step 1: After the slice extension to Domain B resources, the service level Load Balancing module is reconfigured in order to take into consideration the newly allocated resources.

Steps 2 – 7: At this point, some users will be served from Domain A, while others will be served from domain B. Figure 6-6 illustrates the case where a UE is redirected to Domain B. Particularly, the User Equipment gets served by Operator B (step 2). At the same time, Domain B collects VNF and VIM monitoring data and sends them to the Cross-domain Monitoring and Analytics module, which, in turn, aggregates them and uses AI techniques in order to make predictions of the service performance for the near future (steps 3 – 7).

6.6 Trustworthy Slice setup with 3rd party resources

We distinguish two cases for the Resource Providers. In the first case, described in this section, the operator provides only compute resources and does not maintain any orchestration components, while in the second case, explained in the next section, the operator supports a MANO framework. Figure 6-7 details the steps taken by a Resource Consumer (Domain A) and a Resource Provider (Domain B) for the extension of Domain A's slice into domain B's offered resources. The participating modules are the *Identity Management and Permissions Management*, the *Network Slice and Service Orchestration*, the *eLicensing Management*, the *Intelligent Network Slice and Service Optimization* and the *Virtual Resource Management and Control*, analysed in Sections 5.3.7, 5.3.11, 5.3.12, 5.3.15 and 5.3.16, respectively.

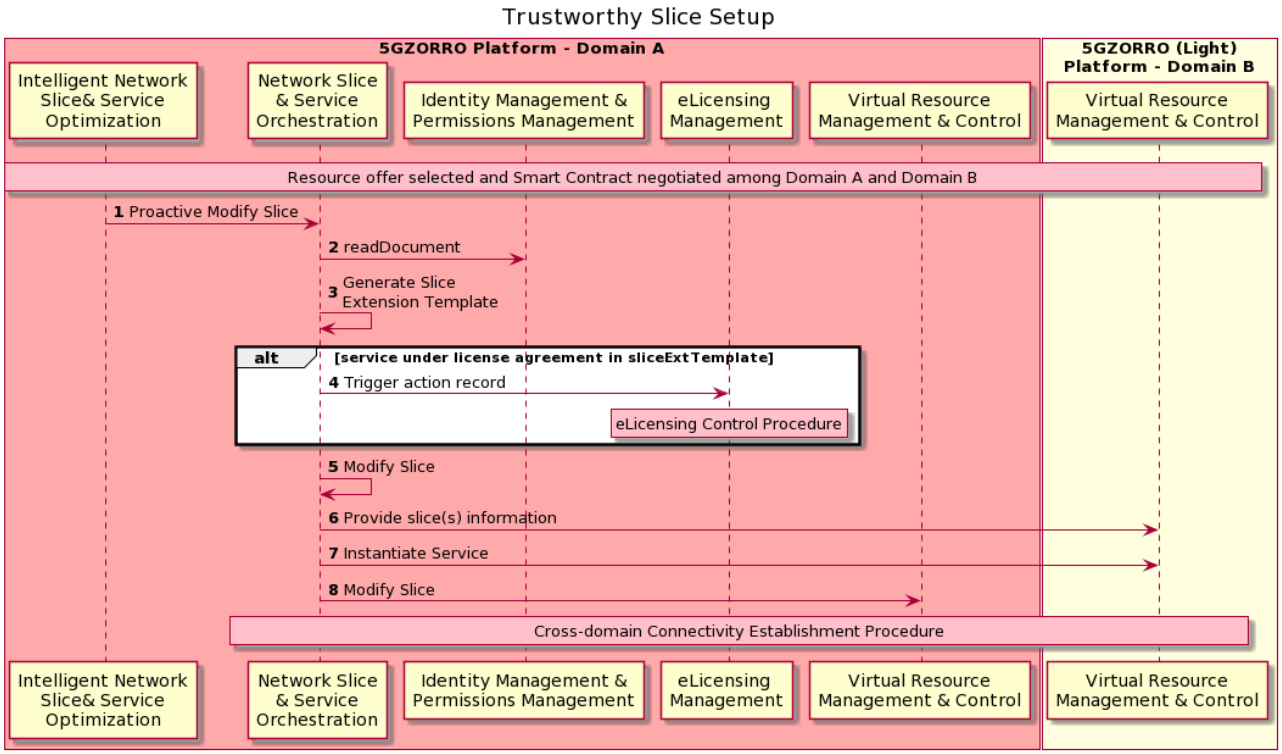

Figure 6-7: Workflow for trustworthy Slice setup (Domain B as a pure Resource Provider)

Prior to slice extension setup, it is assumed that Domain A has already selected Domain B as Resource Provider and an agreement has been established between these two parties (as detailed in previous sections). Then, the next steps are realised:

Step 1: The *Intelligent Network Slice and Service Optimization* at Domain A requests the slice extension to *the Network Slice and Service Orchestration* (*NSSO A*).

Step 2: Upon this request, *NSSO A* retrieves the external resource Access Point from Domain A's *Identity Management and Permissions Management* module.

Step 3: *NSSO A* handles the creation of the slice extension template which describes how the slice should incorporate resources from Domain B. This template includes also the components required for the establishment of secure connectivity across domains.

Step 4: In case that the slice deployment includes a service that is associated with some license agreement, *NSSO A* should trigger the eLicensing control in the *eLicensing Management* of Domain A.

Steps 5-8: Then, *NSSO A* orchestrates the slice extension towards domain B. Particularly, at step 6, *NSSO A* provides the *Virtual Resource Management and Control* of Domain B with any service and slice information required for the service instantiation. Then, the service is instantiated (step 7) and Domain A updates its local slice in order to enable the establishment of cross-domain connectivity (step 8). At this point, the two domains complete the establishment of the cross-domain secure network connectivity.

6.7 Trustworthy Slice setup with 3rd party orchestrated services

This operational pattern describes the sequence of operations involved in the setup of a cross-domain slice containing 3rd party orchestrated services. As mentioned before, this scenario conceives the extension of a slice by concatenating a service offered and orchestrated by another domain.

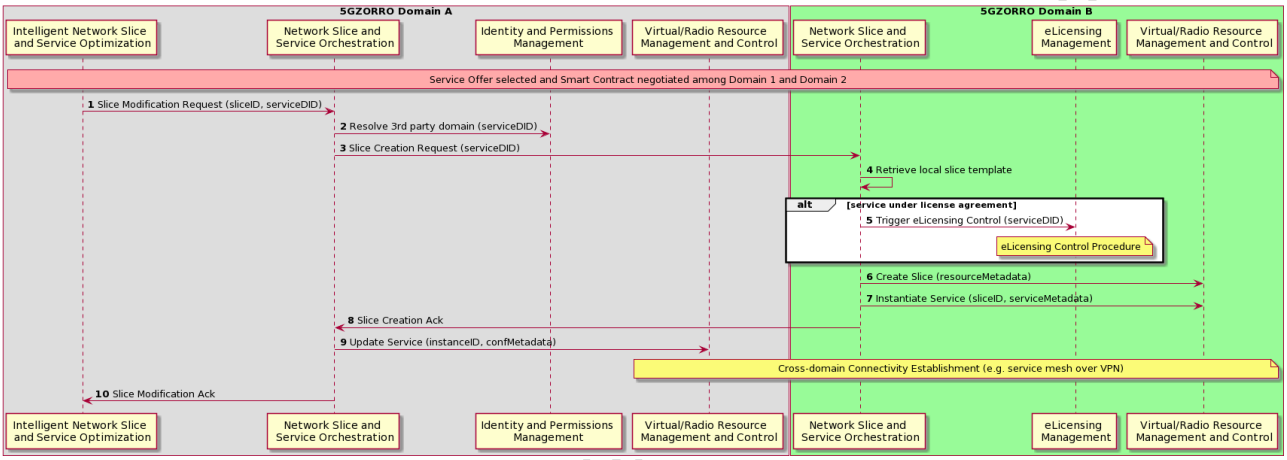

Figure 6-8: Trustworthy slice setup with 3rd party orchestrated services

Figure 6-8 depicts the sequence diagram considering two involved domains (A and B). The workflow is as follows:

Step 1: Once a service offer has been selected and a smart contract has been negotiated among Domains A and B, a slice modification request is triggered towards the Network Slice and Service Orchestration in Domain A. As previously outlined in Subsection 5.3.11 this request is analysed by the invoked Network Slice and Service Orchestration in order to identify the type of modification to be performed with respect to involved domains. In the illustrated example, the *serviceDID* parameter from another domain indicates an integration with 3rd party orchestrated services.

Step 2: Consequently, the identity of the corresponding 3rd party domain is resolved via the Identity and Permission management. With the received information, the Network Slice and Service Orchestration in Domain A is able to reach its peer in Domain B.

Step 3: A slice creation request is next sent towards Network Slice and Service Orchestration in Domain B indicating the specific service to be orchestrated.

Step 4: After receiving this call, the invoked Network Slice & Service Orchestration identifies as own the given *serviceDID,* which is translated into a slice template. With such template, the slice creation and service orchestration requests are next handled in a per-domain manner.

Step 5: To ensure the fulfilment of licensing agreement (if any), a request is triggered towards the eLicensing Management regarding the involved service. More details about this procedure are next described in Sec. 0.

Step 6: The Network Slice and Service Orchestration issues the slice creation request to the corresponding Virtual/Radio Resource Management and Control using the *resourceMetadata* retrieved in step 4. As result, such managers conduct the required resource configuration and slice provisioning.

Step 7: Likewise, by using the *serviceMetadata* of the considered template, the Network Slice and Service Orchestration proceeds with the service instantiation over the corresponding slice. As part of this step, a periodic polling is started to check the instantiation state of the network service.

Step 8: After a correct instantiation is completed, the Network Slice and Service Orchestration in Domain B acknowledges the successful slice extension to its peer in Domain A.

Step 9: In Domain A, a service update request is issued to enable the establishment of cross-domain connectivity. This request is relied to the Virtual/Radio Resource Management and Control for service (re-)configurations that may include the allocation of security credentials and/or load balancing updates, among others. As result, the establishment of the secure cross-domain network is achieved between the two domains.

Step 10: The Network Slice and Service Orchestration acknowledges the successful slice extension to the Intelligent Network Slice and Service Optimization. This step also is used to update the data lake about the performed slice extension.

6.8 Trustworthy e-licensing control

In this section is detailed the operational pattern that describes the sequence of operations involved in the e-licensing management, previously introduced in section 3.5 and specified in Section 5.3.11. Figure 6-1 illustrates graphically the operations and all steps are explained below.

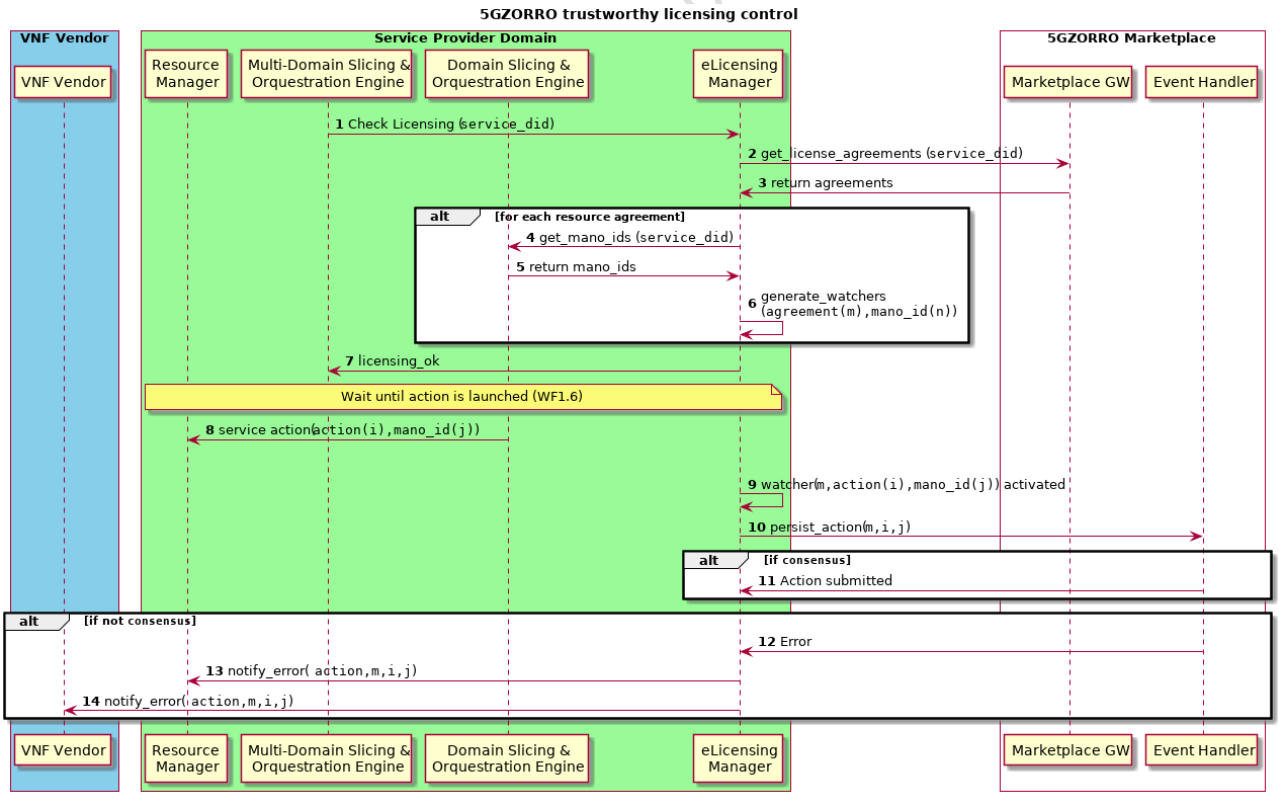

Figure 6-9: Trustworthy licensing control

Step 1: MD-Slicing and Orchestration Engine triggers the licensing checking in the service that is going to be created/modified.

Steps 2-3: The eLicensing Manager requests to the marketplace the related agreements.

Step 4-5: If the service has software components with licensing constraints associated, requests the identifiers used by the Slicing and Orchestration Engine of the virtual functions.

Step 6: Once retrieved the agreements and the mano identifiers, the eLicensing Manager creates the watchers. These watchers observe the MANO and hook the actions performed by the Slicing and Orchestration Engine in the virtual components that have licensing agreements associated. Each eLicensing Manager is responsible for the reporting of the actions in its domain, so will only observe its own infrastructure manager.

Steps 7: ACK licensing OK

Step 8-9-10: When the Service provider Slicing and Orchestration Engine performs some action in the Resource Manager, the watcher will trigger the alert and launch the transaction for the implied agreement, action and *mano_id* to the Marketplace.

Step 11: The Marketplace perform the actions to callout to the involved stakeholders' Resource Managers to sign the transaction. All these steps are performed under the DLT procedure to add an entry in the blockchain and will return an ACK.

Steps 12-13-14: The action was not persisted in the DLT, a notification is sent to the RM and to the VNF Vendor

6.9 Intelligent SLA monitoring & breach prediction

This operational pattern describes the sequence of operations involved in the *Intelligent SLA Monitoring and Breach Prediction* functional element which was introduced in Section 3.6 and analysed in Section5.3.14. In particular, Figure 6-10 and Figure 6-11 present the workflows for the SLA monitoring and the SLA breach prediction, respectively. Starting with Figure 6-10, it illustrates the sequence of operations for monitoring the Resource Agreement (SLA), while ensuring that the generated data are trustworthy.

In Figure 6-10, it is assumed that the trustworthy slice setup has been completed, as indicated in Sections 6.6 and 6.7, and the SLA Smart Contract agreement has been established or updated. Given these, the next process is followed:

Step 1: When a new contract is created or updated, the *Service and Resource Monitoring* is configured according to the requirements specified in the agreement.

Steps 2 – 5: As long as Domain B provides resources to Domain A, both provider and consumer carry out the data monitoring process and they send this information directly to the shared Data Lake, through the *Service and Resource Monitoring*. Data sent by *Virtual Resource Management and Control* modules are accompanied by access rights, so that only authorized parties can read them.

Step 6: Then, the process of performance analysis and monitoring data aggregation would begin, verifying at the same time that the metrics and data sent are correct. The data aggregation process is realised using methods agreed in the Smart Contract and taking into account data provided by both Domains (Resource/Service Provider and Consumer).

Step 7: Additionally, the respective Smart Contracts are updated with the new aggregated monitoring data.

Step 8: For each Smart Contract, the received measurements are analysed and compared with the SLA requirements in order to find out whether an SLA violation has occurred. The Smart Contract is also responsible for specifying the actions that must be taken in case of an SLA breach.

Steps 9 – 12: In the event that the data received by *Service and Resource Monitoring* cannot be verified, a dispute will be generated which must be resolved via governance consensus. Then, if an agreement is reached, the procedure explained in steps 6 to 8 will be performed.

Steps 13 – 16: When the Smart Contract expires, the service is torn down and the status of marketplace resources as well as the cross-domain monitoring functional element are updated.

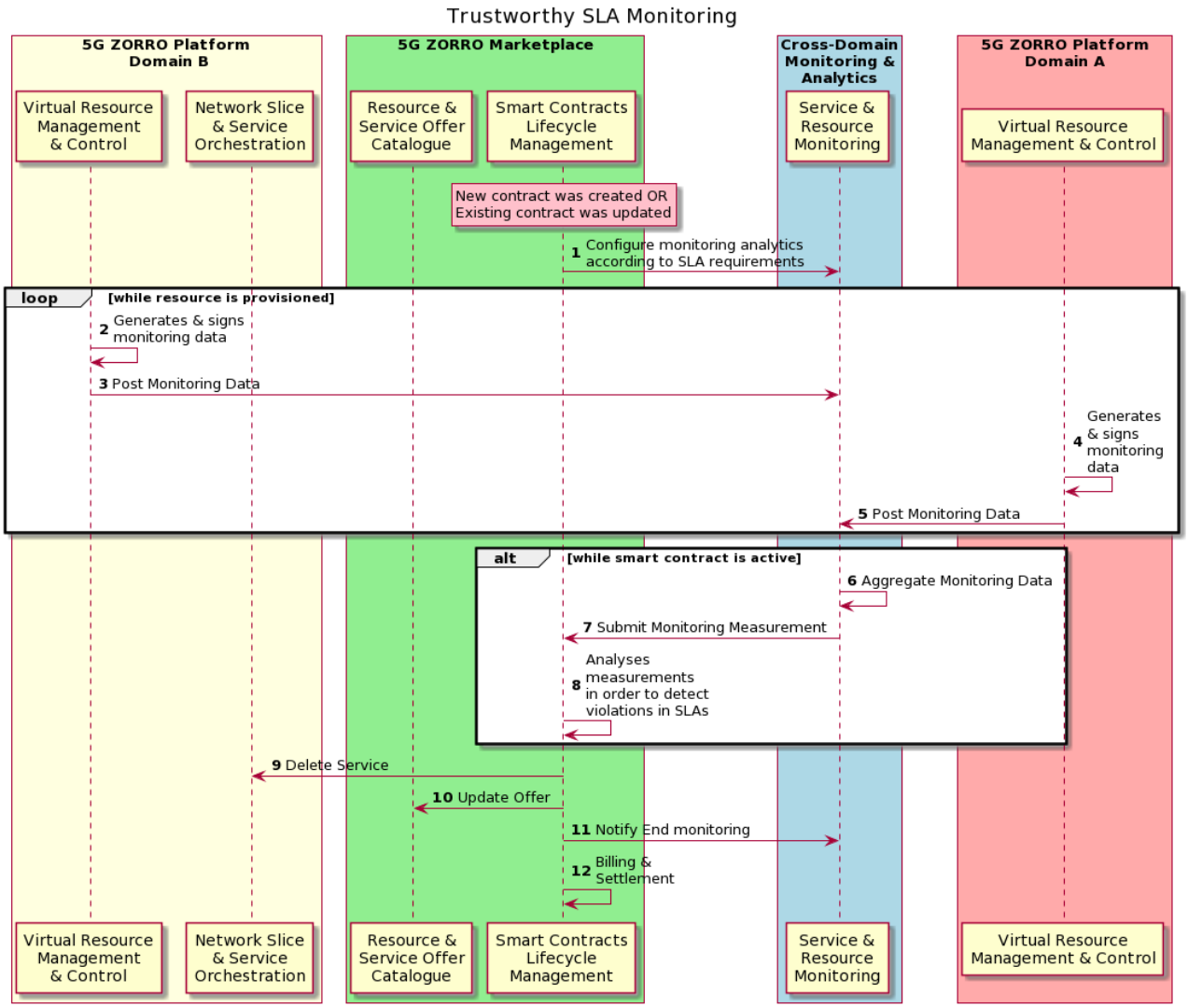

Figure 6-10: Workflow for Trustworthy SLA Monitorin

Figure 6-11 shows the Workflow for the SLA Breach Prediction, which is analysed in the following steps:

Step 1: The *Intelligent Network Slice and Service Optimization* of the Resource Provider requests from the *Intelligent SLA Breach Prediction* functional element to start the algorithms for the SLA Breach Prediction.

Step 2: The *Intelligent SLA Breach Prediction* module subscribes to *Service and Resource Monitoring* in order to get the data needed for running the SLA Breach Prediction Algorithm.

Steps 3 – 6: Monitoring data from both parties is recorded in the Data Lake, through the *Service and Resource Monitoring*.

Step 7: In turn, the *Service and Resource Monitoring* module analyses and aggregates the ingested data according to specifications defined in the Smart Contract.

Steps 8 – 9: *Service and Resource Monitoring* periodically publishes the aggregated monitoring data, which can then be retrieved by the *Intelligent SLA Monitoring and Breach Prediction* module in order to train the Machine Learning (ML) model.

Step 10: The ML model is executed at certain time intervals.

Steps 11 – 12: In case that an SLA Breach is predicted, the *Intelligent Network Slice and Service Optimization* of the Resource Provider is informed accordingly and takes actions according to predefined rules.

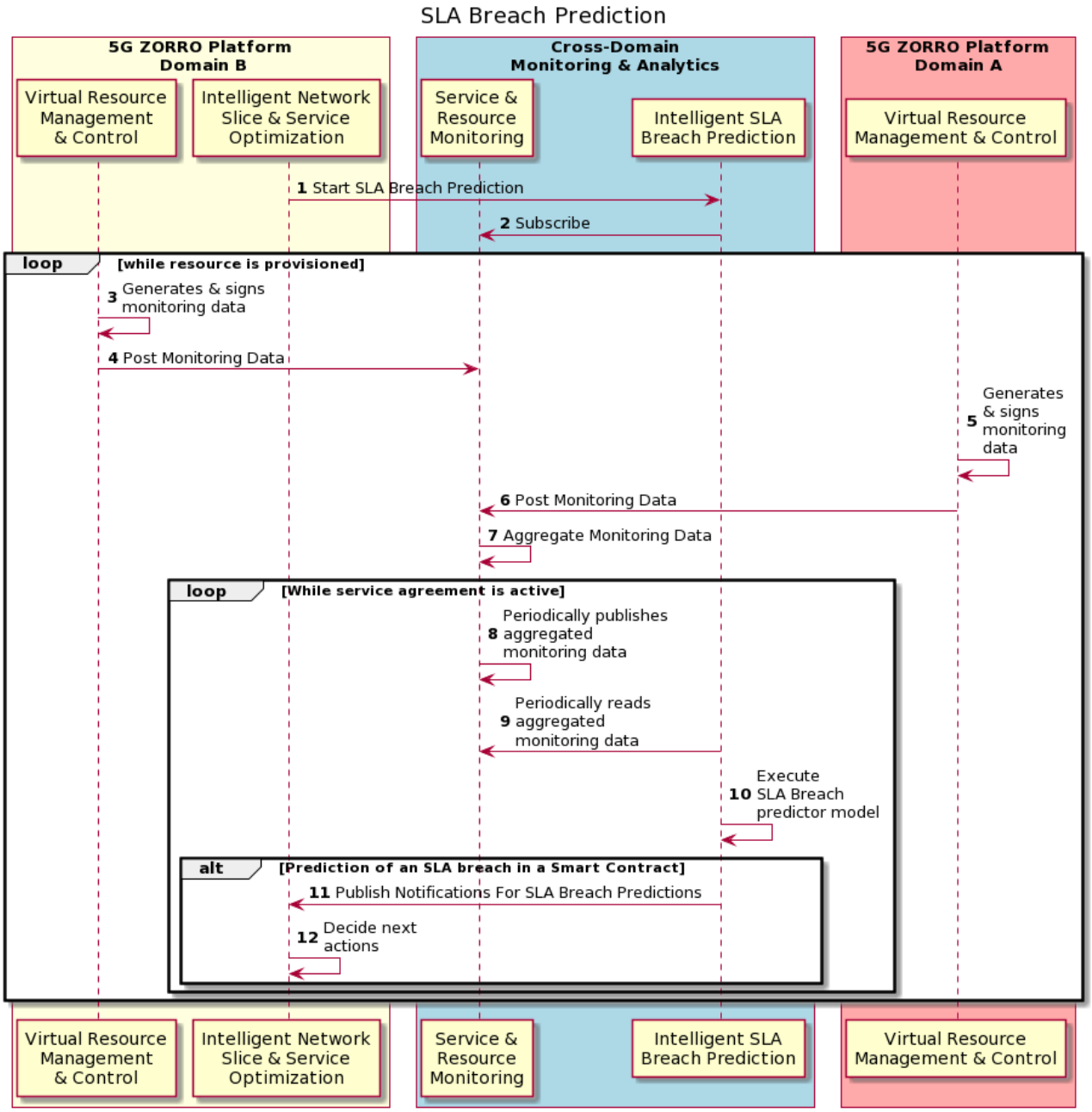

Figure 6-11: Workflow for SLA Breach Prediction

6.10Intelligent Network Slice and Service optimization

In Figure 6-12, we show the workflow for Intelligent Network Slice and Service Optimization. Once trust is established between different 5GZORRO entities across multiple domains, the following steps occur in sequence. The steps (1-3) indicate that both radio and virtual resource managers monitor their managed entities and push this monitored data to Intelligent Network Slice and Service Optimization functional element on a periodic interval. Also, Intelligent Network Slice and Service Optimization functional element receives SLA breach prediction information from Intelligent SLA monitoring & breach prediction functional element. The step (4) prepares data for building machine learning models that can proactively perform operations (e.g., VNF scaling) to optimize the network performance. The steps (5-8) try to perform VNF

scaling/domain slice extension operation within the same domain using Network Slice and Service Orchestration (single-domain) functional element. If previous step is unsuccessful, steps (9-12) performs trustworthy resource discovery from other domains and establishes trustworthy resource agreement setup through the Marketplace Network. Once the resource agreement is complete, steps (13-15) performs VNF scaling/multi-domain slice extension using Network Slice and Service Orchestration (multi-domain) functional element. The steps (16-18) evaluate the performance of the machine learning model and if necessary, re-trains the model with new monitoring data. Finally, steps (19-20) perform trustworthy resource agreement monitoring and resource agreement breach prediction through Intelligent SLA monitoring & breach prediction and Intelligent Network Slice and Service Optimization functional elements.

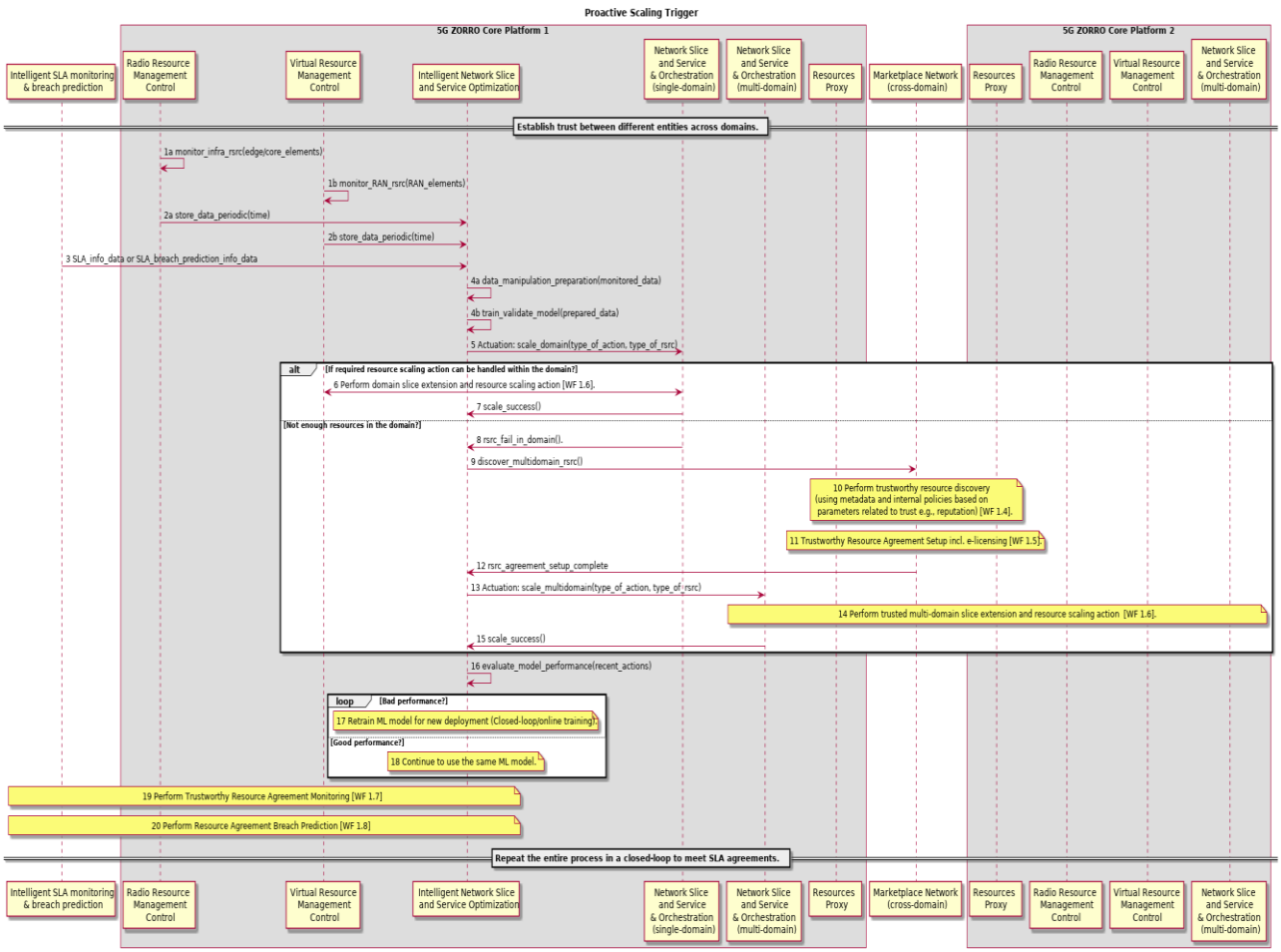

Figure 6-12: Intelligent Network Slice and Service Optimization

In the workflow represented in Figure 6-13, it is assumed that a Vertical (e.g CDN) can proactively request for resource reservation from the Communication Provider (CSP) according to the traffic that it anticipates for a future time. Particularly, the service that is hosted in Domain A's infrastructure expects that is going to need additional resources for a specific time period and/or for a specific geographic area. Moreover, we assume that vertical maintains an SLA Smart Contract with every CSP that hosts its services.

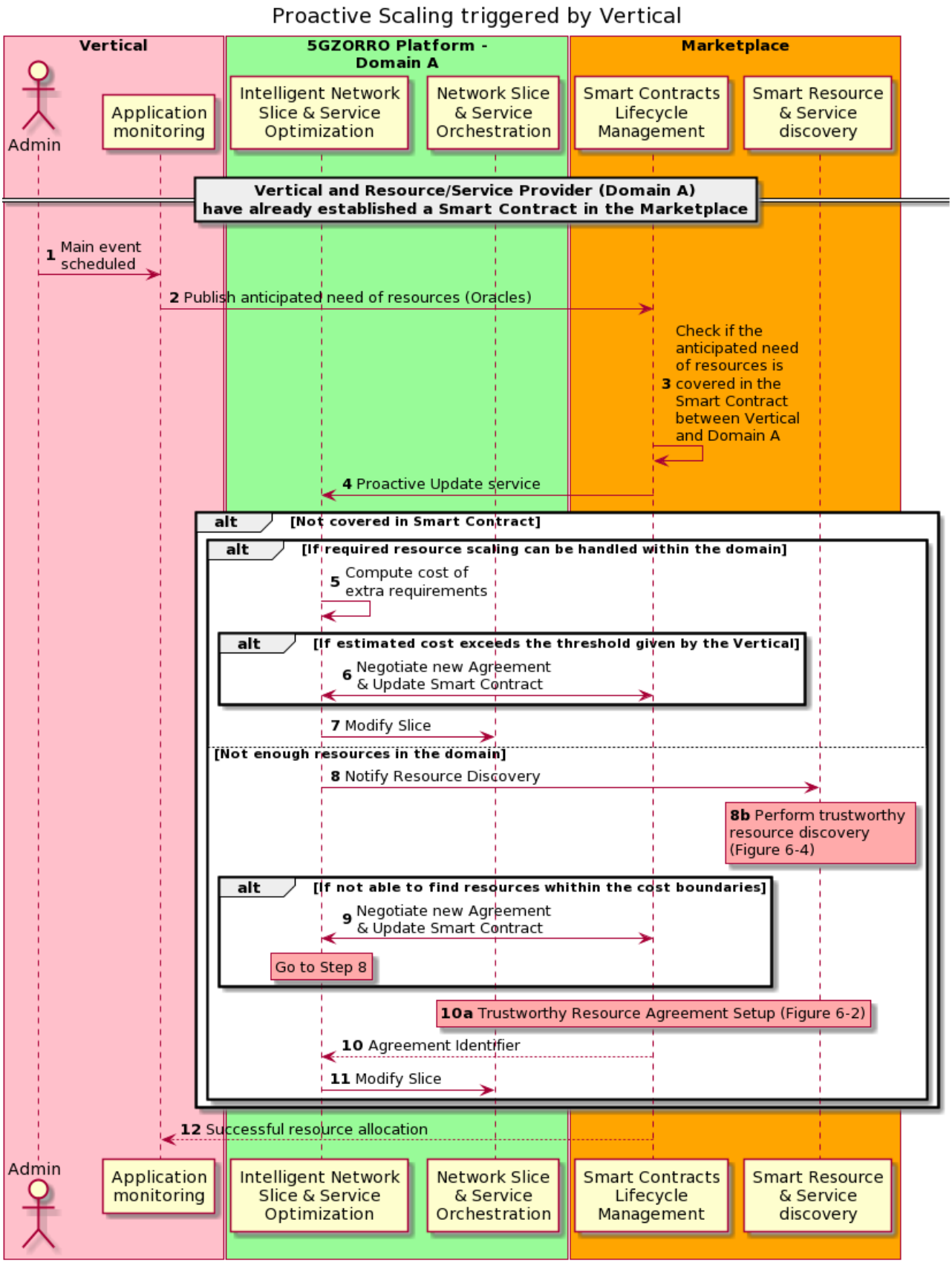

Figure 6-13: Workflow for Proactive Scaling triggered by Vertical

The sequence of operations presented in Figure 6-13 is analysed as follows:

Steps 1: The Vertical receives an input that indicates that a big amount of resources will be required at a specific time period and geographic area.

Step 2: Then, the Vertical publishes the anticipated need for extra resources to the Marketplace though Oracles. This request must entail the Smart Contract Identifier in order to specify which Domain should be responsible for reserving the new resources. In the example shown in Figure 6-13 that is Domain A.

Step 3: The Marketplace checks, through the *Smart Contract Life-cycle Management*, if the resource request is already covered in the Smart Contract between Vertical and Domain A.

Step 4: Domain A must be informed (through the Smart Contract Life-cycle Management) about the anticipated high resource consumption, so that it will be prepared. In case the Agreement already covers the anticipated need of resources, there is no need for updating the contract between CDN and CSP.

Steps 5 – 11: If the need of resources exceeds the agreement's terms, then the Smart Contract between Vertical and Domain A must be updated, as the Vertical should cover the cost of the extra resources. Alternatively, a new contract may be created which will be valid only for the extra resources and for the time that it is needed. These steps are similar to the above workflow (Figure 6-12) with the additional consideration of updating the Smart Contract between Vertical and Domain A. More specifically, in the case of intra-domain scaling (steps 5 – 7), the *Intelligent Network Slice and Service Optimization* module of Domain A estimates the cost of providing the extra resources and if the cost exceeds the threshold defined by the Vertical, then Domain A's and Vertical's Marketplace Nodes should renegotiate their agreement. If an agreement is reached, then either a new Smart Contract is established or the old one is updated. Finally, the *Network Slice and Service Orchestration* engine schedules the resource allocation and the slice instantiation for the specified time. On the other hand, if the Domain A deems necessary to expand to 3rd party resources (Steps 8 – 11), the steps are similar to the ones presented on Figure 6-12, with the exception that the process of resource discovery will be repeated in case that Domain A cannot reach an agreement with any Resource Provider with the given cost threshold defined by the Vertical-Domain A contract. In this case, a new negotiation will be needed between Domain A and the Vertical, so that the later will increase the cost threshold.

Step 12: The last step shows that the *Smart Contrast Life-cycle Management* should return to the Vertical the status of the resource allocation process (success or fail).

7 5GZORRO Platform design

In this section, it is described how the principles discussed in Section 5.1 have been applied to design the 5GZORRO platform.

7.1 Platform design principles and architectural patterns

The base idea is to design the platform as set of modules encompassing one or more of the Functional Elements exposed in Section 5.3. Each module, and its own functionality exposed, represents a 5GZORRO Platform Service and is characterized by the following properties:

- **Loosely coupled**, i.e. is implemented minimizing dependencies between services
- **Highly maintainable and testable**
- **Independently deployable** (as much as possible) from other services and **scalable** if needed
- **Configurable at runtime**

Two main architectural patterns consider service as a key concept: *Service Oriented Architecture* (SOA) and *Microservice Architecture* (MSA). Both patterns present pros and cons and, although from a certain point of view they could be overlapped, they are characterized by several important differences.

In the SOA, the platform is structured as a collection of few services. Each service implements a complete (and even complex) functionality and all services communicate by means of an Enterprise Service Bus (ESB), which enable discovery, connectivity and routing between services.

In a similar way, in the MSA, the platform is still structured as collection of services, but each service, called "micro-service", implements usually one simple functionality and exposes an interface towards the final users or other services. Hence, differently from SOA, the microservices are typically in the order of hundreds. Complex tasks are implemented making microservices communicate directly with each other though well-defined interfaces using lightweight communication mechanisms, such as REST API, without the need of a central bus. In addition, in MSA, each service should maintain its own database with its data.

A third way is represented by the *Service-Based Architecture* (SBA). To the best of our knowledge, a formal definition of such an architectural pattern does not exist and it is often difficult to highlight the differences with the other service-based patterns. In [116], SBA is described as a middle ground between SOA and MSA. Nevertheless, SBA has been chosen by 3GPP as architectural pattern for the design of the 5G System Architecture [115] and by ETSI for the definition of the ZSM Architecture [12].

In particular, we consider ETSI ZSM as the main reference architecture for the design of the 5GZORRO platform as it is closer to the scope of the platform itself. It offers a guideline for the implementation of a zero-touch platform and, with respect to 3GPP 5G System Architecture, ETSI ZSM does not define any specific technology or protocol to be used, so becoming more suitable as “architectural template” for the aims of 5GZORRO.

7.2 Software Architecture Overview

From software components perspective it is envisaged that the functional entities, introduced in Section 5.3, are deployed in four different software platforms (see Figure 7-1).

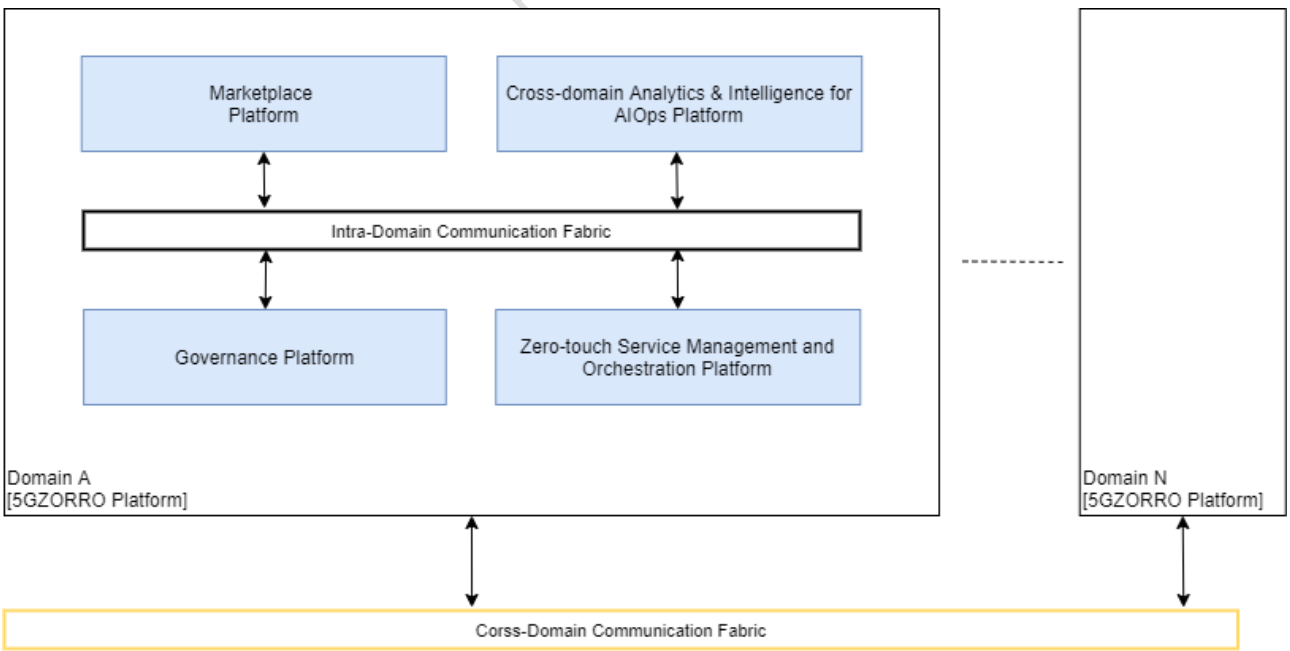

Figure 7-1: 5GZORRO Software Platform overview

The **zero-touch Service Management and Orchestration Platform** is mainly responsible to control 5G Resources including Radio Spectrum resources, Transport Networking resources and Computing resources

(at data centers and at edge computing nodes) as well as existing legacy resource controllers from previous 5G deployments. 5G Resources offered in the 5GZORRO Marketplace are managed through a Resources Manager interface (including Resource offer status management and Monitoring Data Exposure) while the Network slice and service orchestrator manages the life-cycle of slices and associated services at domain level and across different domains. The zero-touch Service Management and Orchestration Platform leverages data lake features (data transformation, analytics and real-time actions) to achieve the automation of some resource management procedures including a proactive scaling mechanism to increase or decrease the infrastructure capacity by using external resources published in the marketplace by Resource Providers. The zero-touch Service Management and Orchestration Platform features a Trust and Security framework to enable trustworthy usage of external resources.

The **Marketplace Platform** leverages DLT technologies including Smart Contracts technologies to enable the trade of 5G resources managed by the zero-touch Service Management and Orchestration. I.e. the 5GZORRO Marketplace is accomplished by a mesh of distributed Marketplace Platforms each one anchored to one Marketplace DLT nodes, and it is envisaged that each CSP has at least one Marketplace Platform instance deployed. The Marketplace Platform features an end-user front-end, a decentralized catalogue for 5G Resource offers and 5G Service offers, as well as the life-cycle management of smart contracts for offers and agreements between providers and consumers, as described in Section 7.5.

The **Cross-Domain Analytics & Intelligence for AIOps platform** mainly comprises the cross-domain Functionalities from the Analytics & Intelligence for AIOps logical layer, described in Section 5.2, i.e. it leverages distributed data lake and AI technologies to provide data persistence, data share and data analytics across domains. It includes functionalities like the ingestion and transformation of monitoring data as well as the automation of complex resource management procedures across-domain. The prediction of SLA breaches and the discovery of the most appropriated resources available in the marketplace are two examples of such resource management procedures automation. Permissions to publish resource data and to read aggregated shared data or cross-domain analytics are managed by the Governance Platform.

The **Governance Platform** is operated by stakeholders with permissions to take decisions according to the Marketplace Governance Model i.e. stakeholders playing the Governance Administrators business role, as defined in D2.1. The Governance Platform also features the decentralized management of global (cross-domain) identifiers (stakeholder identifiers and 5GZORRO resource identifiers) according to Self-sovereign Identity principles and by leveraging DLT technologies. It supports the creation, verification and revocation of certificates as well as authentication and authorisation of identities across all 5GZORRO domains.

The **Intra-Domain Communication Fabric** and **Cross-Domain Communication Fabric** are the two different types of Communication Fabrics in the 5GZORRO Platform; the former allows modules to communicate with other modules inside the same domain, and the latter allows modules to communicate with modules in other domains. They implement Communication fabrics functional block (see Section 5.3.10) and, according to ETSI ZSM, the communication can be based on both publish/subscribe and request/response paradigms. In this sense, the 5GZORRO Platform design forecasts the implementation of both communication paradigms, by using the publish/subscribe, as much as possible, for the interaction between the service implemented from scratch in the platform, and the request-response paradigm for all those services which require it (e.g. NFVO).

7.3 zero-touch Service Management and Orchestration

The Zero-Touch Service Management and Orchestration is a logical group of different software modules, all related to zero-touch capability and service management and orchestration, and it comprises:

- The **Virtual Resource Manager**, which is the interface to all resource controllers in the infrastructure. It implements the following functional blocks:
    - Virtual resource management and control (see Section 5.3.16)
    - Service & resource monitoring (see Section 5.3.13)

- Radio Resource Management & Control (see Section 5.3.17)

These functionalities could be implemented and replicated directly in all of the services that need to access, deploy and monitor resources, avoiding the use of a unified centralized manager. For instance, instead of create a software module with service and resource monitoring functionalities, the Network Slicer could implement these functionalities, collecting monitoring data directly from resources and storing them into the Data Lake.

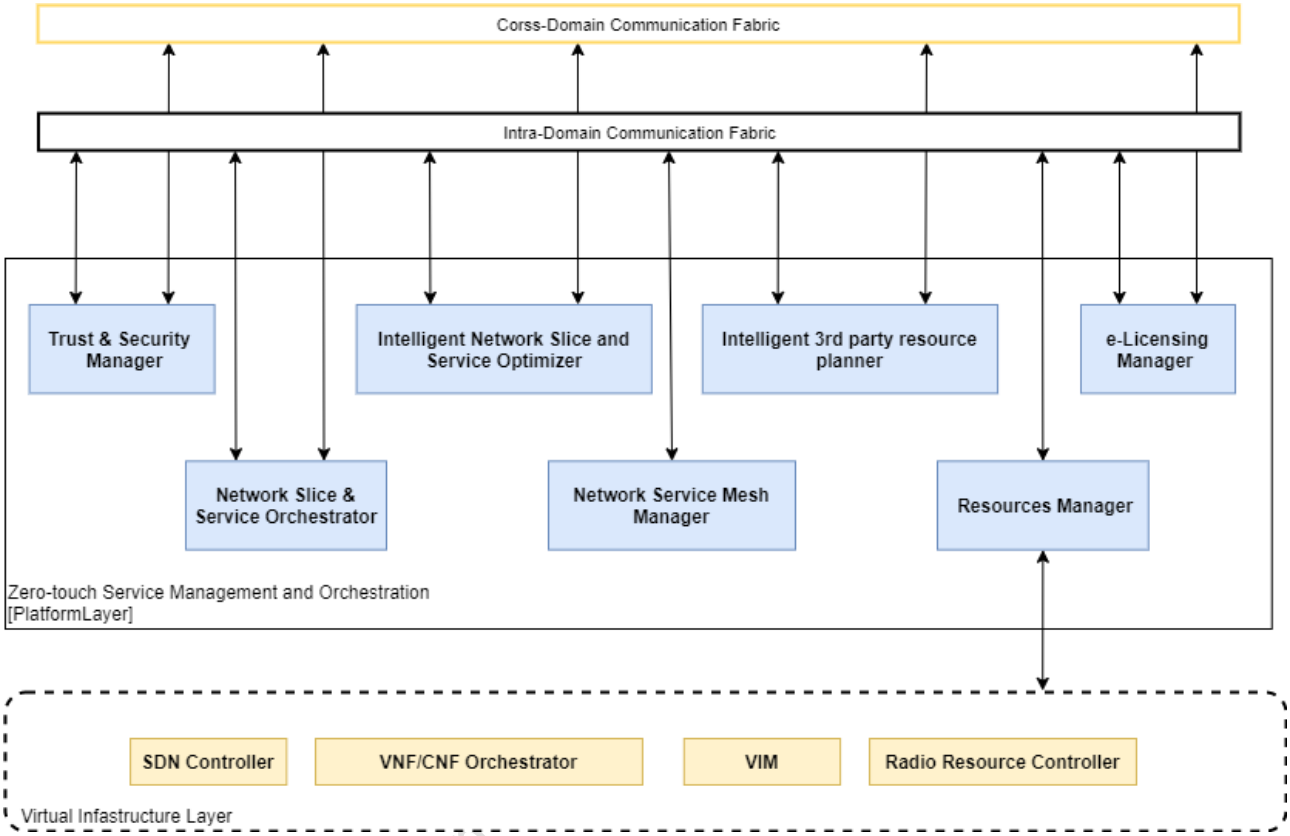

Figure 7-2: Zero-touch Service Management and Orchestration platform

- The **Network Slice and Service Orchestrator**, which is the software module responsible for creating and managing, in an intelligent and automated way, network slice and network service instances in the intra-domain or inter-domain environment. It implements the following functional blocks:
  - Network slice and service orchestration (see Section 5.3.11)
  - Intelligent network slice and service optimization (see Section 5.3.15)
- The **Network Service Mesh (NSM) Manager**, which is the software module that handle the service meshes, detailed in Section 11.4.6, that can be used for providing connectivity between different network services. It could be realized as a distinct software module, implementing some functionalities of the Network slice and service orchestration (see Section 5.3.11) related to service meshes, or these functionalities could be implemented in the Network Slicer module, previously described, because the management of service meshes is strictly related to the management of network slices and their network services.
- The **Trust & Security Manager**, which is the module responsible for evaluating the Trust and Security of a stakeholder or resources; it implements Trust and security management (see Section 5.3.8).

- The **e-licensing Manager**, which is the module responsible for controlling licensing terms of a service and resources published in the Marketplace and instantiated in the domain; it implements the E-licensing Management (see Section 5.3.12).

Outside the Zero-Touch Service Management and Orchestration there is the virtual infrastructure, which abstracts the real 5G network elements, such as SDN Controllers, VNF/CNF Orchestrator, VIMs and radio Resource Controllers.

7.4 Governance applications

At this stage it is envisaged that the Governance Platform is mainly comprised by four main components, as discussed below.

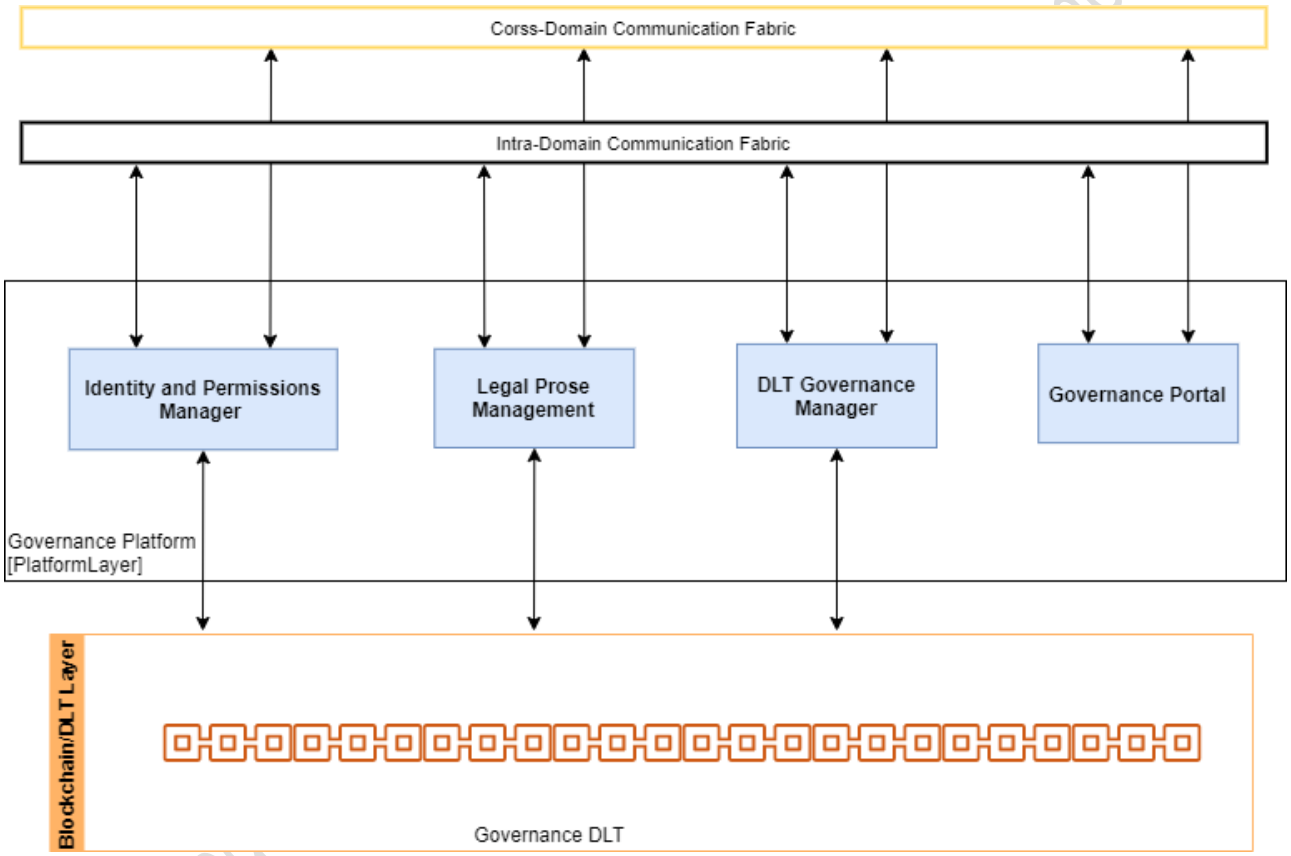

Figure 7-3: Governance Platform architecture

Governance Manager: it implements the Governance Manager functional element as defined in Section 5.3.1, i.e., it provides functionalities to support a consortium governance model for 5GZORRO marketplace. In this way, decisions like admittance, revocation of membership and dispute resolution is managed in accordance with a mutually agreeable governance model. Any Governance decision should be issued as a Verifiable Claim to be associated to some 5GZORRO Subject (Stakeholder or Business Agreement)

Legal Prose Manager: it implements the Legal Prose Repository functional element and its interfaces as defined in Section 5.3.3 i.e. it provides a shared repository of parameterised legal statement templates that can subsequently be associated with a given resource or service by providers. It is envisaged that such parameterised legal statement templates would be Verifiable Claims data schemas that are registered in the Governance DLT.

Identity and Permissions Manager: it implements the server side functionalities for Identity Management and Permissions Management functional element and its interfaces as defined in Section 5.3.7 i.e. it provides appropriate mechanisms to identify entities, services, resources, consumers, providers, and organizations, which allows decentralisation of the system without forgetting the security principles, a reliable authentication using DIDs, DID Documents, and Verifiable Credentials, and finally, a granular control access mechanism that standardises authorised access to data, resources, and services. All the other 5GZORRO platforms (i.e., Marketplace, zero-touch Service Management and Orchestration and Cross Domain Analytics & Intelligence for AIOps platform) should feature a DID Agent component to interact with the Identity and Permissions Manager component, as DID Subjects with Verifiable Credential Holder roles i.e. it should feature a secure storage for the private keys and Verifiable Credentials.

Governance Portal: provides a web user interface to enable the usage of the Governance features by end-users including management of legal proses, tools to take governance decisions and management of Identities and Permissions.

In order to make the Governance Platform as much as possible agnostic of the Governance DLT implementation, there should be a Driver component (or library) that exposes an intent based API with a high level of abstraction of the DLT technology.

7.5 Trustworthy Marketplace applications

At this stage it is envisaged that the Marketplace Platform is mainly comprised by four main components, as described below.

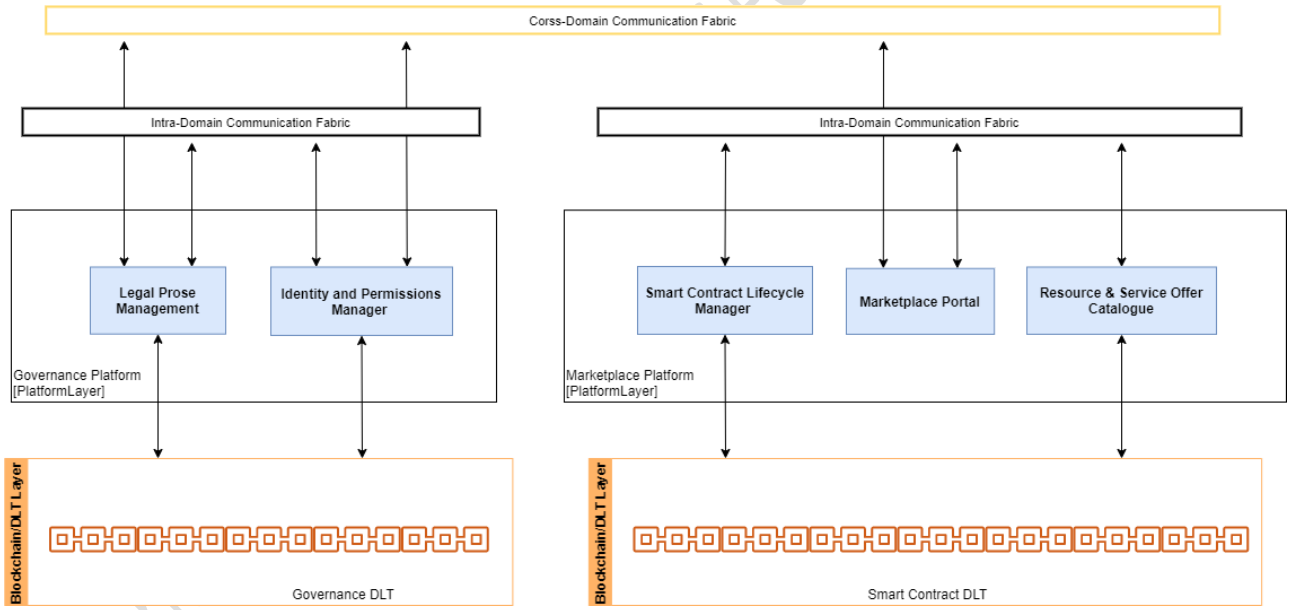

Figure 7-4: Marketplace Platform architecture

Resource & Service Offer Catalogue: it implements the Resource & Service Offer Catalogue functional element and its interfaces as defined in Section 5.3.2, i.e., it provides functionalities to publish and manage resources or services offers into 5GZORRO Marketplace, list the active resource and service offers, modify or remove a resource or service offer, and make an offer for a specific resource or service.

Smart Contract Lifecycle Manager: it implements the Smart Contract Lifecycle Manager functional element and its interfaces as defined in Section 5.3.6, i.e., it provides functionalities to manage Marketplace business contracts (DLT Smart Contracts) throughout their lifecycle, from agreement negotiation and instantiation through to termination.

Marketplace Portal: provides a web user interface to enable the usage of the Marketplace features by end-users including the discovery of offers, management of offers and the management of business agreements.

For non-Governance Administrators stakeholder players-, the Marketplace Portal will interface with external Governance Platforms, via the cross-domain Communication Fabric, to access the Identity and Permissions Manager and the Legal Prose Manager features. Thus, these external service endpoints should be discoverable from Cross-domain Communication Fabric registry.

7.6 Cross-domain Analytics & Intelligence for AIOps

As discussed above, 5GZORRO services and functional entities are implemented as microservices. Each component is packaged in a container that can be independently deployed. The component typically expects the availability of some kind of input data and provides some kind of output data. The component is typically triggered by the occurrence of some event (possibly a message on a message queue) and may trigger some other event to have its output data consumed. Many of these services run in the environment of a (private or shared) Data Lake, which provides the basic infrastructure to coordinate the microservices.

Using the services of a Data Lake usually includes several coordinating services, which can be thought of as a pipeline. The general cycle of operations is:

1. Gather data.
2. Perform some analysis on the data to obtain insights.
3. Perform some action based on the result of the analysis.

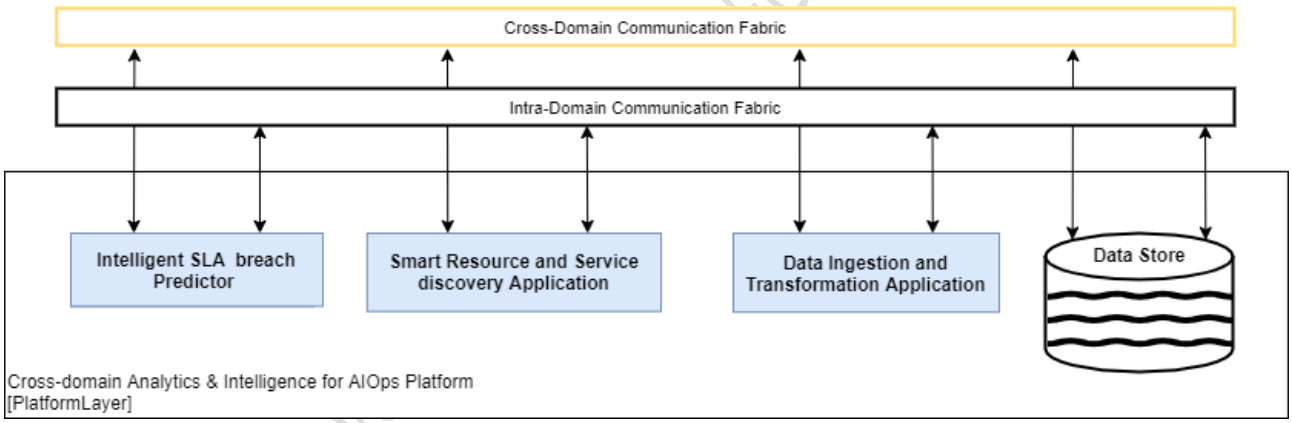

Figure 7-5: Cross-domain Analytics & Intelligence for AIOps platform

In 5GZORRO, we have a number of specific examples, all of which can be modelled with the same pipeline idea. These include:

- Intelligent SLA monitoring & breach predictor,
- Intelligent Network Slice and Service optimizer,
- Intelligent 3rd party resource planner,
- Data Ingestion & Transformation.

For example, for the Intelligent SLA monitoring & breach predictor, we have the following specific pipeline stages.

1. Provide monitoring data.
2. Aggregate the monitoring data.
3. Using the monitoring data, evaluate SLA satisfaction.
4. Perform some analytics to predict violation of SLA.

5. Raise an event for actions to be performed upon prediction of SLA violation.

Similar pipelines need to be defined for all the other analytics workflows.

The implementation and deployment of the stages of the pipeline must be coordinated. The output from one stage typically serves as the input for another stage. The information transferred between stages may be the actual data upon which to operate, or it may simply be a notification that the previous processing stage has completed with an indication of where to find the data in the data store.

We propose a general framework to allow easy connectivity between services defined in a Data Lake. (See [111] for a concrete realization of a pipeline architecture for Kubernetes.)

- Register the service; specify the data that is to be used by the service (some kind of pointer to the data or other type of description of the data).
- Use messaging for communication between services.
- Each service has a channel from which it receives input and has a channel to which it sends output. The input/output might be the actual data, or it may include a pointer to the location of the relevant data to be processed.
- The output produced by a single component could be consumed by multiple consumers. For example, the output of a data aggregator service may be forwarded both to the data store as well as to a service that checks for SLA compliance.

Finally, suppose the analytics (or action) wants to trigger some functionality on the client. We need to define a way for these to interact. This could be encapsulated in a message delivered via an output channel.

The various parameters to set up the microservice are all contained in a (e.g. yaml) configuration file. This configuration file is provided together with the compiled microservice module. Upon registration, variable fields of the configuration file are provided to customise the particular pipeline.

8 Conclusions

This deliverable provides the high-level reference architecture and core design artefacts of the 5GZORRO functional entities which are core inputs to the subsequent development activities for the 5GZORRO Platform.

The architecture presented in this document is the base input for the low-level design and implementation work planned on the 5GZORRO software platform.

A summary of the specific contribution of the presented design elements to the 5GZORRO objectives and related KPIs is provided in Table 8-1 in terms of applicable design artefacts. Future implementation and validation activities will take care of achieving and demonstrating the target KPI metrics.

An update of this specification is planned after the completion of the first implementation cycle in order to include refinements and detailed interface specifications for the various services which incorporate decisions and feedback from implementation.

Table 8-1: D2.2 contribution to 5GZORRO objectives and KPIs.

OBJECTIVE	Target KPIs	Applicable Design Artifact
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).*	See Sec. 5.3.1 for DLT Governance management and Sec. 5.3.10 for Communication Fabric functional blocks.
	• *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. 5.3.19 for Data lake functional block and Sec. 6.9 for Intelligent SLA Monitoring & breach prediction workflow.
	• *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).*	See Sec. 5.3.19 for Smart Resource and Service discovery.
	• *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. 5.3.11 for Network Slice and Service Orchestration, Sec. 6.10 for Intelligent Network Slice and Service optimization, and Sec. 5.3.16 for Virtual Resource Management and Control.
	• *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).*	See Sec. 5.3.18 for DLT platform and Sec. 6.4 for Trustworthy Smart Contract setup for spectrum workflow and in general all the Functional Entities from the trading layer.
OBJ-2. Design and prototype a security and trust framework, integrated with 5G service management platforms, to demonstrate Zero-Day trust establishment in distributed multi-stakeholder environments and automated security management to ensure	• *Provide mechanisms for zero touch trust automation in multi-domain scenarios on top of a 5G service management framework (KPI target: to cover up to 4 different stakeholders as part of the automated trust establishment process and to enable its automatic renegotiation when a stakeholder is joining or leaving the trust link).*	See Sec. 5.3.7 for Identity and Permission management, Sec. 5.3.8 for Trust and Security management and 5.3.9 for Trust Execution Environment management functional blocks.
	• *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*	Addressed by the Security and Trust Management Functional Entity

OBJECTIVE	Target KPIs	Applicable Design Artifact
trusted and secure execution of offloaded workloads across domains in 5G networks	*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).*	
	• *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. 5.3.9 for Trust Execution Environment management functional block.
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. 5.3.3 for Legal Prose Repository functional block, Sec. 5.3.5 for Intelligent 3rd party resource selection, Sec. 5.3.8 for Trust and Security management, and Sec. 5.3.11 for Network Slice and Service Orchestration. See Sec. 6.6 for Slice setup with 3rd party resource workflow.
	• *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 discovery, brokerage and selection, operation with SLA to facilitate workload offloading to 3rd-party resources supporting	• *Automatically discover and "inventorize" various types of resources (i.e. compute, storage, network at core, edge, far-edge), spectrum and services capabilities from different domains and service providers (KPI target: distribution of resource updates and discovery in less than 10 mins).*	See Sec. 5.3.4 for Smart Resource and Service discovery and Sec. 6.3 for Resource discovery workflow.
	• *Implement/correlate technical service configurations and SLA monitoring interactions between multiple parties (KPI target: SLA*	See Sec. 5.3.13 for Service and Resource Monitoring, Sec. 5.3.14 for Intelligent SLA

OBJECTIVE	Target KPIs	Applicable Design Artifact
pervasive computing across multiple 5G domains.	*measurements and validation from at least 3 operators involved in a multi-party service chain).*	Monitoring & breach prediction and Sec. 5.3.19 for Data lake platform.
	• *Support intent-based API to guide the AI-driven resource discovery system (KPI target: open 5GZORRO API specification for resource discovery).*	See Sec. 5.3.5 for Smart Resource and Service discovery and Sec. 5.3.5 for Intelligent 3rd party resource selection.
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).*	See Sec. 5.3.2 for Resource and Service offer catalogue and Sec. 6.2 for Spectoken Resource offer publishing.
	• *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).*	See Sec. 5.3.2 for Resource and Service offer catalogue and Sec 5.3.7 for Identity and Permission management.
	• *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).*	n/a
	• *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. 5.3.17 for Radio Resource Management and Control functional block.
OBJ-6. Realize a cloud-friendly network software licensing framework for location	• *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. 5.3.3 for Legal Prose Repository. See Sec. 5.3.12 for e-Licensing management and Sec. 6.8 for Trustworthy e-License control workflow.

OBJECTIVE	Target KPIs	Applicable Design Artifact
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. 5.3.12 for the definition of the e-Licensing management and Sec. 6.8for Trustworthy e-License control workflow.
	• *Instantiate Network Services with VNFs from diverse providers (KPI target: use eContract to include VNF licensed by at least 3 different providers).*	See Sec. 5.3.12 for the definition of the e-Licensing management, and Sec. 5.4.7 for Network Slice and Network Service offer 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 Sec 4.

9 References

[1] 5GZORRO Consortium, Deliverable D2.1 – "Use Cases and Requirements Definition", May 2020

[2] 5G; Management and Orchestration; Concepts, Use Cases and Requirements (3GPP TS 28.530 Version 15.0.0 Release 15), October 2018. https://www.etsi.org/deliver/etsi_ts/128500_128599/128530/15.00.00_60/ts_128530v150000p.pdf.

[3] GSMA NG.116 - Generic Network Slice Template, V 3.0, 22 May 2020available online: https://www.gsma.com/newsroom/wp-content/uploads/NG.116-v3.0.pdf

[4] ETSI TS 123.501 5G; System Architecture for the 5G System, V15.2.0 June 2018, available online: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.02.00_60/ts_123501v150200p.pdf

[5] 3GPP 5G, Management and orchestration, 5G Network Resource Model (NRM); 3GPP TS 28.541 version 15.4.0 Release 15available online: https://www.etsi.org/deliver/etsi_ts/128500_128599/128541/15.04.00_60/ts_128541v150400p.pdf

[6] http://webfunds.org/guide/ricardian.html

[7] D4.1 First open-source release of the SDK toolset, https://www.5gtango.eu/project-outcomes/deliverables/42-d4-1-first-open-source-release-of-the-sdk-toolset.html

[8] D4.2 Final release of the service validation SDK toolset, https://www.5gtango.eu/project-outcomes/deliverables/71-d4-2-final-release-of-the-service-validation-sdk-toolset.html

[9] D7.3 Final demonstrators and evaluation report, https://www.5gtango.eu/project-outcomes/deliverables/78-d7-3-final-demonstrators-and-evaluation-report.html

[10] ETSI GR NFV-EVE 012; Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework, v3.1.1, December 2017, available online: https://www.etsi.org/deliver/etsi_gr/NFV-EVE/001_099/012/03.01.01_60/gr_nfv-eve012v030101p.pdf

[11] 3GPP TR 28.801 Telecommunication management; Study on management and orchestration of network slicing for next generation network, V15.1.0, January 2018, available online: https://www.3gpp.org/ftp//Specs/archive/28_series/28.801/28801-f10.zip

[12] ETSI GS NFV-SOL 005; Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point, V3.3.1, September 2020, available online: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/03.03.01_60/gs_NFV-SOL005v030301p.pdf

[13]AWS License Manager. AWS. Web: https://aws.amazon.com/license-manager/

[14] Google How Licensing Works. Google. Web: https://support.google.com/a/answer/6309862?hl=en

[15] TM Forum, SLA Handbook [TMF GB917Release 3.1]

[16] 5G TANGO, The Importance of Service Level Agreements in the 5G era, https://5gtango.eu/blog/56-the-importance-of-service-level-agreements-in-the-5g-era.html

[17] Prometheus - Monitoring system & time series database, https://prometheus.io/

[18] RabbitMQ AMQP Message Broker, https://www.rabbitmq.com/

[19] Tang, B. and Tang, M., 2014, September. Bayesian model-based prediction of service level agreement violations for cloud services. In *2014 Theoretical Aspects of Software Engineering Conference* (pp. 170-176). IEEE.

[20] Leitner, P., Michlmayr, A., Rosenberg, F. and Dustdar, S., 2010, July. Monitoring, prediction and prevention of sla violations in composite services. In *2010 IEEE International Conference on Web Services* (pp. 369-376). IEEE.

[21] Hussain, W., Hussain, F.K., Hussain, O. and Chang, E., 2015, November. Profile-based viable service level agreement (SLA) violation prediction model in the cloud. In *2015 10th international conference on P2P, parallel, grid, cloud and internet computing (3PGCIC)* (pp. 268-272). IEEE.

[22] ETSI zero-touch network and Service Management (ZSM), Requirements based on documented scenarios, ETSI GS ZSM 001 V1.1.1, October 2019, available online: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/001/01.01.01_60/gs_ZSM001v010101p.pdf

[23] ETSI zero-touch network and Service Management (ZSM), Reference Architecture, ETSI GS ZSM 002 V1.1.1, August 2019, available online: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf

[24] CORDA 4.5 Documentation – CORDA API Section, https://docs.corda.net/docs/corda-os/4.5.html

[25] ETSI zero-touch network and Service Management (ZSM), Terminology for concepts in ZSM, ETSI GS ZSM 007 V1.1.1, August 2019, available online: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/007/01.01.01_60/gs_ZSM007v010101p.pdf

[26] ETSI Network Functions Virtualisation (NFV), Architectural Framework, ETSI GS NFV 002 V1.2.1, December 2014, available online: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

[27] ETSI Network Functions Virtualisation (NFV), Management and Orchestration, ETSI GS NFV-MAN 001, V1.1.1, December 2014, available online: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf

[28] Experiential Networked Intelligence (ENI); System Architecture, September 2019. https://www.etsi.org/deliver/etsi_gs/ENI/001_099/005/01.01.01_60/gs_ENI005v010101p.pdf.

[29] https://www.tmforum.org/about-tm-forum/

[30] https://inform.tmforum.org/catalyst/2019/05/blockchain-infrastructure-marketplace-enables-pop-networks-fly-business-models/

[31] TM Forum Product Catalog Management API. https://github.com/tmforum-apis/TMF620_ProductCatalog

[32] https://www.tmforum.org/vertical-industry-telcos-federated-dlt-based-marketplace/

[33] “Spectrum trading in cognitive radio networks: A market-equilibrium-based approach” D. Niyato and E. Hossain, IEEE Wireless Communications, vol. 15, no .6, pp 71-80, Dec 2008

[34] "Reconfigurable Radio Systems (RRS); evolved Licensed Shared Access (eLSA); Part 1: System Requirements", ETSI TS 103 652-1 v1.1.1 (2019-02) https://www.etsi.org/deliver/etsi_ts/103600_103699/10365201/01.01.01_60/ts_10365201v010101p.pdf

[35] "Licensed Shared Access (LSA)", ECC Report 205, Feb. 2014, https://docdb.cept.org/download/baa4087d-e404/ECCREP205.PDF

[36] "Automated Frequency Coordination, An Established Tool for Modern Spectrum Management," Dynamic Spectrum Alliance, Mar. 2019, http://dynamicspectrumalliance.org/wp-content/uploads/2019/03/DSA_DB-Report_Final_03122019.pdf

[37] ITU-T (2019) Distributed Ledger Technology Reference Architecture. Technical Specification FG DLT D3.1.

[38] ITU-T DLT reference architecture https://www.itu.int/en/ITU-T/focusgroups/dlt/Documents/d31.pdf

[39] Bitcoin – Developer Guides, https://developer.bitcoin.org/devguide/block_chain.html

[40] Cao, Yu & Sun, Yi & Min, Jiangsong. (2020). Hybrid blockchain–based privacy-preserving electronic medical records sharing scheme across medical information control system. Measurement and Control. 002029402092663. 10.1177/0020294020926636.

[41] MEF 3.0 overview, http://www.mef.net/mef30/overview

[42] MEF LSO Sonata APIs FAQ, v6, June 2020, http://www.mef.net/Assets/Documents/LSO_Sonata_FAQ.pdf

[43] MEF-LSO-Sonata-SDK (Release Candidate 5), https://github.com/MEF-GIT/MEF-LSO-Sonata-SDK

[44] Open Network Automation Platform, https://www.onap.org/

[45]Communications Business Automation Network. https://cban.net/

[46] Communications Business Automation Network (CBAN) Whitepaper version 1.0. https://9db836f3-ccf1-4990-82a6-3036d6f8890a.filesusr.com/ugd/d18226_4bb3582cc7314f649cdaf56811e93d6e.pdf

[47] CBAN Reference Architecture, January 2020. https://9db836f3-ccf1-4990-82a6-3036d6f8890a.filesusr.com/ugd/d18226_545a1ea92d814206957a0d4d41a73a17.pdf

[48] CBAN MVP Definition Data on Demand, January 2020. https://9db836f3-ccf1-4990-82a6-3036d6f8890a.filesusr.com/ugd/d18226_223d1039cc044bd09b1b88ed8b6b20c5.pdf

[49] Service Operations Specification MEF 55. Lifecycle Service Orchestration (LSO): Reference Architecture and Framework. https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_55.pdf

[50] Guo, J., & Chen, R. (2015). A classification of trust computation models for service-oriented internet of things systems. In 2015 IEEE International Conference on Services Computing (pp. 324-331). IEEE.

[51] Sharma, A., Pilli, E. S., Mazumdar, A. P., & Govil, M. C. (2016). A framework to manage trust in internet of things. In 2016 International Conference on Emerging Trends in Communication Technologies (ETCT) (pp. 1-5). IEEE.

[52] Surridge, M., Correndo, G., Meacham, K., Papay, J., Phillips, S. C., Wiegand, S., & Wilkinson, T. (2018). Trust Modelling in 5G mobile networks. In Proceedings of the 2018 Workshop on Security in Softwarized Networks: Prospects and Challenges (pp. 14-19).

[53] S. Burikova et al. (2019). A Trust Management Framework for Software Defined Networks-based Internet of Things. 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2019, (pp. 325-331).

[54] Fernandez-Gago, C., Moyano, F., & Lopez, J. (2017). Modelling trust dynamics in the Internet of Things. Information Sciences, 396, 72-82.

[55] Decentralized Identifiers (DIDs) v1.0. Drummond Reed; Manu Sporny; Markus Sabadello; Dave Longley; Christopher Allen. W3C. 28 July 2020. W3C Working Draft. Available online: https://www.w3.org/TR/did-core/

[56] Sporny, M., Longleyy, D., and Chadwick, D. Verifiable Credentials Data Model 1.0. Expressing verifiable information on the Web. W3C Recommendation 19 November 2019. Available online: https://www.w3.org/TR/vc-data-model/

[57] Hyperledger Indy. Available online: https://www.hyperledger.org/use/hyperledger-indy

[58] Sovrin Fundation. Available online: https://sovrin.org/

[59] Cordentity. Corda Marketplace. Available online: https://marketplace.r3.com/solutions/cordentity

[60] Hyperledger Aries. Available online: https://marketplace.r3.com/solutions/cordentity

[61] SOFIE - Secure Open Federation for Internet Everywhere. Available online: https://cordis.europa.eu/project/id/779984

[62] Siris, V. A., Dimopoulos, D., Fotiou, N., Voulgaris, S., & Polyzos, G. C. (2019). Interledger smart contracts for decentralized authorization to constrained things. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 336-341). IEEE.

[63] ITU-T (2017) ITU-T. Y.3052. Overview of trust provisioning in information and communication technology infrastructures and services.

[64] ITU-T (2018) ITU-T. Y.3053. Framework of trustworthy networking with trust-centric network domains.

[65] ITU-T (2018) ITU-T. Y.3054. Framework for trust-based media services.

[66] Richer, J. & Johansson, L. (2018). Vectors of Trust. RFC 8485. Available online: https://tools.ietf.org/html/rfc8485

[67] ETSI (2020) Electronic Signatures and Infrastructures (ESI); Global Acceptance of EU Trust Services. https://www.etsi.org/deliver/etsi_tr/103600_103699/103684/01.01.01_60/tr_103684v010101p.pdf

[68] ISO (2018) Framework of trust for processing of multi-sourced data. Available online: https://www.iso.org/standard/74844.html

[69] ISO (2020) Security recommendations for establishing trusted connection between device and service. Available online: https://www.iso.org/standard/56572.html

[70] Levin, A., Garion, S., Kolodner, E. K., Lorenz, D. H., Barabash, K., Kugler, M., McShane, N. (2019). AIOps for a Cloud Object Storage Service, 2019 IEEE International Congress on Big Data (BigDataCongress), https://ieeexplore.ieee.org/document/8818188

[71] Cloud Storage as a data lake, https://cloud.google.com/solutions/build-a-data-lake-on-gcp

[72] 4 Data Lake Solution patterns for Big Data Use Cases, https://blogs.oracle.com/bigdata/data-lake-solution-patterns-use-cases

[73] Data Lake Storage for Big Data Analytics, https://azure.microsoft.com/en-us/services/storage/data-lake-storage/

[74] AWS Lake Formation, https://aws.amazon.com/lake-formation/

[75] Cloudera Data Platform, https://www.cloudera.com/products/cloudera-data-platform.html

[76] Data Management with Zaloni, https://www.zaloni.com/

[77] Teradata and Data Lakes, https://www.teradata.com/Cloud/Data-Lake

[78] Impetus Data Lake Architecture, https://www.impetus.com/big-data/data-lake-creation

[79] Open Data Hub, https://opendatahub.io/

[80] Ceph, https://ceph.io/

[81] Apache Spark – Unified Analytics Engine for Big Data, https://spark.apache.org/

[82] Project Jupyter, https://jupyter.org/hub

[83] Prometheus – Monitoring system and time-series database, https://prometheus.io/

[84] Grafana: The open observability platform, https://grafana.com/

[85] Apache Kafka – A distributed streaming platform, https://kafka.apache.org/

[86] Seldon – Machine learning deployment for enterprise, https://www.seldon.io

[87] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3-24.

[88] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, *16*(3), 645-678.

[89] Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237-285.

[90] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(2), 1-19.

[91] Nokia Network Operations Master, https://www.nokia.com/networks/products/network-operations-master/

[92] Ericsson Automated Network Operations, https://www.ericsson.com/en/portfolio/digital-services/automated-network-operations

[93] Acumos AI, https://www.acumos.org/

[94] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283).

[95] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

[96] Microsoft Azure Machine Learning, https://azure.microsoft.com/en-us/services/machine-learning/

[97] Google AI Hub, https://cloud.google.com/ai-hub

[98] Amazon SageMaker, https://aws.amazon.com/sagemaker/

[99] GlobalPlatform Device Technology, ". c. (2010). Tee client api specification. http://www.globalplatform.org/specificationsdevice.asp.

[100] Santos, N., Saroiu, H. R., & A., W. (n.d.). Using ARM TrustZone to build a trusted language runtime for mobile applications. SIGARCH Comput. Archit. News, vol. 42, no. 1, pp. 67–80, Feb. 2014.

[101] Sangorr´ın, D., Honda, S., & H., T. (n.d.). Reliable and efficient dual-os communications for real-time embedded virtualization. Information and Media Technologies, vol. 8, no. 1, pp. 1–17, 2013.

[102] Sechkova, T., Barberis, E., Paolino, M. (n.d.). Cloud & edge trusted virtualized infrastructure manager (VIM)-security and trust in OpenStack. 2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW). IEEE, 2019.

[103] Docker. URL: https://www.docker.com/

[104] RKT. URL: https://coreos.com/rkt/

[105] Kubernetes. URL: https://kubernetes.io/

[106] ISTIO. URL: https://istio.io

[107] NSM, Network Service Mesh. URL: https://networkservicemesh.io/

[108] Resource Catalog Management API REST Specification, TM Forum Specification, TMF634, Release 17.0.1, December 2017.

[109] Service Catalog Management API REST Specification, TM Forum Specification, TMF633, Release 18.5.0, January 2019.

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

[111] Argo Workflows and Pipelines. URL: https://argoproj.github.io/projects/argo

[112] Openstack. Open Source Cloud Software. URL: https://www.openstack.org/

[113] Open Source MANO, ETSI-hosted project to develop an Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV. URL: https://osm.etsi.org/

[114] Open Network Operating System (ONOS) open source SDN controller for building next-generation SDN/NFV solutions. URL: https://www.opennetworking.org/onos/

[115] 3GPP; Technical Specification Group Core Network and Terminals; 5G System; Technical Realization of Service Based Architecture; 3GPP TS 29.500 version 17.0.0 Release 17, September 2020, available online:

[116] N. Ford, “Comparing Service-based Architectures”, available on-line http://nealford.com/downloads/Comparing_Service-based_Architectures_by_Neal_Ford.pdf

[117] 5Growth, Deliverable 2.1 “Initial design of 5G End-to-End Service Platform”, https://5growth.eu/wp-content/uploads/2019/11/D2.1-Initial_Design_of_5G_End-to-End_Service_Platform.pdf, November 2019

[118] SliceNet, Deliverable 7.1 “Cross-Plane Slice and Service Orchestrator”, https://doi.org/10.18153/SLIC-761913-D7_1, May 2020

10 Abbreviations and Definitions

10.1 Definitions

No definition introduced in this deliverable.

10.2Abbreviations

5G IA	5G Infrastructure Association
AIOps	Artificial Intelligence for IT operations
CNF	Cloud Native Function
DoA	Description of Action
DID	Distributed Identifier
DLT	Distributed Ledger Technology
EC	European Commission
IPR	Intellectual Property Rights
LCM	LifeCycle Management
MANO	Management and Orchestration
NFV	Networks Function Virtualization
NFVI	Networks Function Virtualization Infrastructure
NFVO	Networks Function Virtualization Orchestrator
NS	Network Service or Network Slice depending on the context
NSM	Network Service Mesh
PPP	Public Private partnership
SBA	Service Based Architecture
SBI	Service Based Interface
SC	Smart Contract
SDO	Standard Developing Organization
SM	Service Mesh
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
WG	Working group
WP	Work Package
ZSM	Zero Touch Service Management

11 Appendix I – Reference architectures and technologies

11.1 Area intelligent zero-touch management

11.1.1 ETSI ZSM

The increasing management complexity in next generation networks makes it challenging to improve the agility in the deployments and maintenance costs. Besides, another degree of complexity is introduced considering the infrastructure availability, flexibility and performance to provide the expected user interactions with all applications across different interconnected domains. The overarching design goal of ETSI zero-touch Service Management ZSM is to provide a framework that provides enablers for zero-touch automated network and data-driven management algorithms in a multivendor environment.

ZSM scenarios and requirements have been delivered in GS ZSM 001 [22], and based on these requirements, the ZSM framework reference architecture has been defined in GS ZSM 002 [23] and a terminology document GS ZSM 007 [25] enumerates and details commonly used terms in the other ZSM documents to clarify the main concepts.

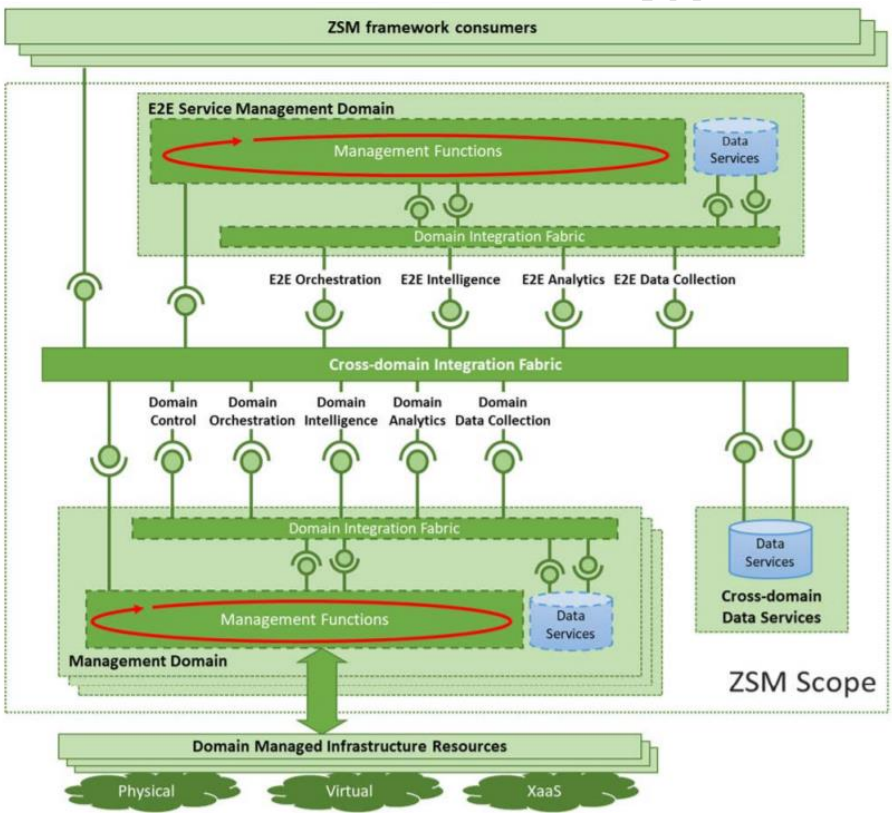

Figure 11-1: ETSI ZSM reference architecture (source [23])

The ZSM reference architecture (see Figure 11-1) proposes a decomposition of complex management services and management functions in fundamental building blocks that are integrated with a set of composition and interoperation patterns.

Management domains are used to create separation of concerns, considering boundaries of administrative, geographical or technological nature. An end-to-end service management domain is a special management domain responsible for the cross-domain management and coordination. Every management domain is composed by different entities:

- Management functions that provides a set of services, by exposing and/or consuming a set of service endpoints.
- Data services enable shared management data access and persistence by authorized consumers across management services within management domains, avoiding the management functions to handle their own data persistence.
- Domain integration fabric is the entity responsible for controlling exposure of services beyond domain boundaries and for controlling access to the management services exposed by the domain.

The cross-domain integration fabric facilitates the provision of services and the accessing of endpoints cross-domain. This also includes services for the communication between management functions, which facilitates the provisioning of "live" management data to consumers who require them. Complementing this, the cross-domain data services provide services to persist data and to access these.

5GZORRO implements concepts of the ZSM architectural reference, like the management domains that will represent the different stakeholders involved in the platform that will offer their own management services exposed by service-endpoints, or the cross-domain integration fabric (see Section 5.3.10), that will enable the provision, communication and coordination of the services. Data services will be also managed in 5GZORRO, sharing operational data produced by different domains to the involved stakeholders through the Cross-domain monitoring and analytics 5GZORRO component.

5GZORRO platform complement solutions for zero-touch automation with a cross-domain Marketplace, based in distributed ledger technology, which is used to establish trust among parties and automate the service and slice resource sharing between different domains. The 5GZORRO architecture is designed to offer network operators and service providers the needed mechanisms to automatically negotiate network slice requests and resource composition with external providers based on the availability and capabilities of the services and resources offered on the Marketplace.

11.1.2 ETSI NFV MANO

The ETSI NFV architecture, illustrated in Figure 11-2 and defined in [26], describes the reference architecture for Network Function Virtualization and it is composed of three main domains: the Virtualised Network Functions (VNFs), which are software implementation of a networking function, the virtualization infrastructure (NFVI), which includes all the computing, storage and networking resources, and the Management and Orchestration (MANO), which is in charge of the management of the Networks Services, VNFs and physical and virtual resources.

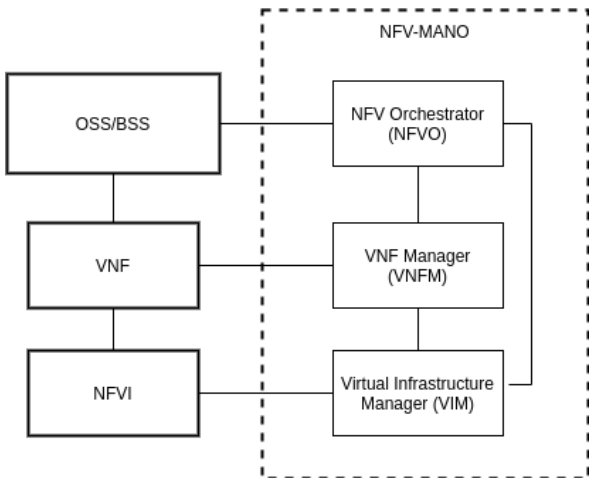

Figure 11-2: Reference architecture for NFV

The architectural framework of ETSI NFV MANO, defined in [27], identifies the functional blocks of the management and orchestration architecture, the interactions between them and their interfaces.

The three functional blocks are:

- *Virtual Infrastructure Manager* (VIM) responsible for controlling and managing the NFVI compute, storage and network resources within one operating infrastructure domain. Some of its functions are the orchestration and management of virtualized resources on top of physical compute, storage and network resources, the management of software images and the collection of performance and fault information of hardware resources.

- *VNF Manager* (VNFM) responsible for the lifecycle management of VNF instances. In details, its functionalities are the VNF configuration and the VNF instantiation, upgrading, scaling and termination.
- *NFV Orchestrator* (NFVO) has two main responsibilities: first it is responsible for the Network Service lifecycle management (e.g., instantiation, scaling in and out, performance measurement, and termination of a Network Service) and then it orchestrates resources across multiple VIMs.

5GZORRO aims to share heterogeneous resources and to handle flexible, on-demand allocation of resources crossing the borders of administrative domains. In order to realize that the MANO, which needs additional resources for its Network Services, should cooperate with the MANO in other administrative domains to instantiate new services or extend existing services. Moreover, VNFs, similar to other software products that can be distributed and deployed, are licensed by their Providers, therefore a VNF has rules and conditions which regulate its usage. A Consumer has to negotiate the usage of VNF with its vendor and usually this negotiation consists of slow legal and commercial agreements, making impossible the on-demand instantiation of new VNFs or Network Services. 5GZORRO uses an e-licensing mechanism that allows Vendors to automate the negotiation process using smart contracts, which contain the license conditions and constraints, and to control that the usage of VNF fulfils the agreements, as described in Section 5.3.12.

11.1.3 ETSI ENI

The ETSI Experiential Networked Intelligence (ENI) Industry Standardization group aims to design a generic Cognitive Network Management architecture to support network service optimization based on AI/ML and context-aware techniques. The final objective is to enable reactive and automated service optimizations based on changes on the overall service context, service requirements, business goals, etc. The proposed architecture uses the information available from the different layers of the MANO platform and from the virtual and physical resources to generate outcomes in the form of: (i) optimization suggestions; (ii) explicit lifecycle management and orchestration actions; (iii) policy updates.

Since the objective is to provide a generic platform which could be applied in different environments and contexts, the specification is purposely designed to be agnostic of the managed system specifics and does not impose any constraint in terms of implementation. In this sense, the whole architectural platform is designed in terms of functional blocks which delimit the functionality of each block, and reference points which describe the information that should be exchanged among the different functional blocks. The main functional blocks of the ENI system are:

- **Data Ingestion**: Handles the appropriate retrieval of the information from the different sources of the assisted system, providing pre and post processing functionalities (including as data correlation, filtering and anonymization capabilities)
- **Data normalization**: Processes the ingested data and transforms it into a format that can be processed by the other ENI functional blocks.
- **Knowledge Management**: Embraces all the process, systems and mechanisms required to support, consume, share and refine the knowledge assets using a consensual knowledge representation.
- **Cognition framework**: This block mimics brain cognitive processes: understands the ingested data, the specific context, the information gathering procedures and determines actions to ensure the achievement of certain goals.
- **Situational Awareness**: This block uses contextual information, and other sources of information (i.e. trend forecasting, prediction, etc) to specify actions based on decisions in order to achieve the desired goals.
- **Model Driven Engineering**: This block transforms the actions specified by the Situational Awareness functional block into policies using a model driven approach.
- **Policy Management**: The policy management block contains the policy repository, and is responsible for deciding and executing the correspondent policies.
- **Denormalization**: Processes information and data produced by the ENI system and translates it into recommendations, commands that the assisted system can understand.

- **Output Generation:** Contains all the processes that are specific to the translate the output of the denormalization to the format required by the assisted system.

The different components of the managed system are seen from the ENI System as external functional blocks. Figure 11-3 illustrates the intended interaction between a typical NFV MANO platform and the ETSI ENI System.

As can be seen from Figure 11-3, the ENI System leverages all the information available from the different layers, and uses the interfaces exposed by the different layers to enforce the optimization outcomes by means of a communication bus.

The architecture from 5GZORRO takes the mentioned centralized communication bus concept, and natively includes the functionality of all the functional blocks identified by the ENI.

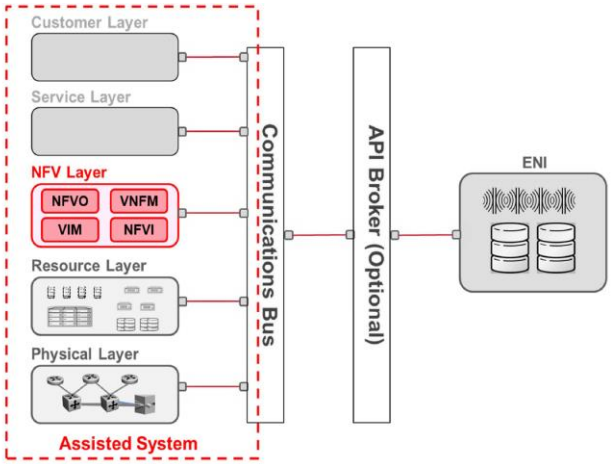

Figure 11-3: Interaction between an NFV MANO platform and the ETSI ENI System (source [28])

In this sense, the 5GZORRO architecture has been designed using the ETSI ENI approach, and therefore it natively supports AI/ML based policy management, orchestration decisions and suggestions.

- **Multi-domain knowledge management:**
- **Situation awareness algorithms for SLA breach prediction**
- **Cross-layer service optimization algorithms**

11.1.4 5GPPP Project Network Slicing approach

5GTANGO [7] (sub-section 2.1.1.6 Slice descriptor) proposed to use Slice Descriptors (SDs), very much like ETSI's Network Service Descriptors (NSDs) and Virtual Network Function Descriptors (VNFDs). Plus, it has proposed a way that those assets would be identifiable before any instantiation (no need for an id that is always connected to a specific Catalogue instance) in any Catalogue. This way of identifying any asset uses a well-known trio of concepts that uniquely identify any NSD, VNFD, SD, etc.:

- Vendor: a string identifying the vendor (or author) of the asset;
- Name: a string identifying the asset (service, functions, slice, etc.);
- Version: a string that uniquely identifies the asset, within the vendor and name;

In this way, every descriptor is identifiable before it is ingested into any catalogue of services, functions, etc., thus allowing for cross-referencing between different asset's descriptors: services may refer to functions, slices to services, etc. In 5GTANGO a slice descriptor has to include at least the following concepts:

- A list of services that will be part of the slice, together with the necessary 'credentials' of each one of those service's service provider: this will allow a slice to be multi-domain, with different services being provided under different circumstances to different 'service consumers';
- A list of interconnections between the above described services, in a similar way the interconnections of multiple VNFs within the same service are described. These interconnections must also take into consideration the necessary credentials to allow different parties to connect different services in the same slice;

5GTANGO's D4.2 [8] gives some concrete examples about Network Slice instantiation, as they were experienced in the several pilots. One of those challenges were the need for (the network slice's or network service's) instantiation parameters: it is not so odd to need the instantiation of the network slice (or one or more instances of the services that comprise the slice) to take into account a specific set of parameters that are only known at instantiation time. It is crucial that the Network Service instantiations processes (see below) in each one of the service providers are able to receive such instantiation parameters, given (at least to some

of) them reasonable default values, give the correct feedback to the 'network slice designer' about those default values (and if any of them cannot be used, etc.).

5GTANGO [9] provides an implementation of network slicing that allows for the sharing of Network Service instances (useful, e.g., when we need a NS that is a singleton, or takes too long to start, consumes too many resources to be multi instantiated, etc.): in this case the shared Network Service instance should not be teared down, of course. Please note that this feature might also be needed for the Network Slice migration feature, described above.

5Growth [117] and SliceNet [118] use similar approach to manage the definition and instantiation of network slices. In practice for the definition these projects use the concept of Network Slice Template (NST) which follows the approach specified by 3GPP for the network slice NRM [5], where a network slice is composed by one or more Network Slice Subnets (NSST). The exact values for the parameters impacting the network slice performance are computed at runtime during the provisioning phase. The NSST composing the NST detail the capabilities and constraints of the individual components of the network slices required to support the end-to-end service logic. Furthermore, NSSTs may include the Network Service Descriptors to be used to deploy the given NSST.

From the management point of view, NSST and NST are created and managed by a Network Slice Provider (NSP). NSTs are offered to the Digital Service Providers (DSPs) in order to be deployed as part of a Digital Services. NSSTs, are also exposed to the DSPs but only as constituent part of the NST, and therefore can not be activated or provisioned as standalone instances.

The Network Slice Instance (NSI) in 5Growth and SliceNet, represents the provisioned slice for a given NST in a NSP administrative domain and follows the same 3GPP NRM [5]. Following the same principle, NSIs can be further decomposed in NSSIs which convey the resource requirements for a certain segments or elements of the given slice instance. These NSIs are created and activated when the DSP issues a NST instantiation request using the id of the requested NST.

11.2Area cross-domain resource & service trading

11.2.1 TMForum Telecom infrastructure marketplace

TM Forum is a neutral and non-profit association "driving collaboration and collective problem-solving to maximize the business success of communication and digital service providers and their ecosystem of suppliers"[29], representing over 850-member companies serving five billion customers across 180 countries. The Association focuses on accelerating CSPs' (Communications Service Providers) digital transformation, shaping their business operations, IT systems and ecosystems to meet the ever-growing need of meeting new highly-demanding requirements brought forth by new communication paradigms and capitalize on new opportunities presented in such a rapidly evolving digital world.

Within TM Forum, "Catalysts" are designed as PoC (Proof of Concept) projects whose goal is to design innovative solutions to new challenges proposed by "Champions" - entities presenting the problem statement to be solved. The outcomes of such may then take the form of white papers, case studies, best practices, lessons learned, API specifications, models, frameworks and reference code. Most recently, in 2019, a Catalyst has been created under the name "Blockchain-based Telecom Infrastructure Marketplace", championed by Orange and Vodafone, two major Telecom Operators, to explore "how blockchain can be used to help the telecoms industry procure infrastructure and assets quickly and cost-effectively to unleash new agile, on-demand business and procurement models."[30] together with other Participants: Infosys, Nokia, IOTA and R3. The produced white paper focuses on CAPEX (Capital Expenditure) challenges CSPs are facing - and will face over the upcoming years -, as well as energy consumption, so that these stakeholders can effectively grow in the most efficient and cost-effective manner when scaling their network capacity and coverage. For this, it is crucial to design a better and more efficient Telecom Infrastructure and Energy Sourcing, deployment and Investment model, as per the white paper. A prime example is the need to acquire,

deploy and operate a massive number of Small Cells in urban areas, whose need for such added capacity and coverage may have even been triggered during a temporary event. Such assets, however, in order to operate as intended, depend on numerous external factors: available passive infrastructure (telecom towers), energy, licensed spectrum, backhaul, municipality authorisation, etc.). The project has then focused on researching and documenting how multiple actors (Service Providers, Asset Providers, Regulators, third-party Sponsors, Infrastructure Provider, Cell Service Provider, etc.) can come together in a trusted manner and enable a common, open and interoperable marketplace facilitating business to happen between such different entities, in an automated fashion.

It is clear then that the work performed under this TM ForumCatalyst is greatly aligned with that of 5GZORRO. Therefore, the consortium has carefully analysed the produced outcomes which have had a direct impact on the definition of the project's architecture. Specifically, the different proposed DLT-based solution architectures have been analysed, proposed by both R3 Corda and IOTA, as well as the produced Data Models and APIs. Based on this, and having in mind 5GZORRO's marketplace - which consists of the onboarding and offering of different asset categories referred to as resources (such as VNFs, NSs and Spectrum, for instance) -, 5GZORRO's designed technical components will take into account the Information Model already described and released as specifications[31], so that the results are aligned with the original vision of TM Forum and to possibly further improve and push for such an interoperable environment. The details of such offerings can be found in section 5.4.2.

Furthermore, prior to this Catalyst, a new one has been created which builds upon the previous work and further enhances it. It is called "Vertical Industry Telcos: a Federated DLT-based Marketplace" which will be working on creating "new open standards for interactions in distributed marketplaces and to offer a variety of practical use cases to cover the new needs of CSPs of any kind" allowing "new business models and value propositions with the main goal to deliver and assure new digital services (network slices) in an agile, cost effective and profitable manner to all partners of the ecosystem"[32]. To this end, 5GZORRO will follow the outcomes of such Catalyst Project, providing that 5GZORRO architecture also builds upon a distributed DLT-based Marketplace.

11.2.2 Licensed spectrum trading

Spectrum is an essential element in the provision of wireless connectivity. Given the ubiquity of wireless networks which can cause interference to each other and the scarcity of spectrum, efficient and effective management of this resource is crucial to ensure that the wireless services provided deliver the expected and necessary quality of service. Spectrum authorization should be mainly achieved via General Authorisation. However, certain applications require a higher level of availability and reliability and therefore need spectrum which is free from interference i.e. individually licensed spectrum. Individually authorized spectrum is normally assigned to Mobile Network Operators who seek to acquire spectrum to offer mobile connectivity over a large geographical area using a public network. Since 5G can also deliver services that either have the capability of connecting massive number of telemetric devices over massive Machine Type Communications (mMTC) or services that require ultra-Reliable Low Latency Connectivity (uRLLC), verticals are now interested in spectrum for their private networks. Usually verticals require a smaller band of spectrum, for a specific geographic area unlike Mobile Network Operators. One-way verticals can access spectrum is by allowing an authorised licensee to trade its underutilized spectrum.

Licensed spectrum trading is a market-driven solution allowing secondary users such as other mobile operators or industry verticals to access spectrum for their public or private networks [33]. There are three types of Spectrum Trading. These are spectrum transfer, spectrum lease and spectrum sharing. Spectrum transfer and lease refer to granting all or some of the spectrum rights for a duration which is equal to the remaining duration of the licence or less respectively. Spectrum Sharing seeks to improve the efficiency of spectrum usage by allowing secondary users to access spectrum without causing interference to the primary user. An NRA adopts different regulatory policies for spectrum transfer, lease or sharing to adequately cater for the different case scenarios that usually vary in terms of geographical coverage, duration of trade and application.

Spectrum sharing can adopt Dynamic Spectrum Access. This can be achieved by deploying a centralized controller that identifies the existing available spectrum for secondary usage. Another solution is decentralized spectrum sharing that relies on Cognitive Radio using Software Defined Radio (SDR) to determine which bands are underutilized and adapt the technical parameters of the SDR accordingly.

The most common implementation of spectrum sharing in Europe is the two-tier Licensed Shared Access (LSA) based on geographic areas and database assist involving an incumbent and a secondary user. Three-tier shared access, combines licensed and opportunistic use such as the Citizen Broadband Radio Service deployed in America and the TV White Space in the UK.

Another important development in spectrum sharing is the work carried out by ETSI on the evolved LSA (eLSA) [34] which is based on the LSA concept [35] but specifically focusing on the needs of the vertical operators defining three geographical zones: allowance zone, restriction zone and protection zone. Unlike LSA, eLSA is frequency agnostic.

The 5GZORRO platform will cater for three phases of licensed spectrum trading. The first phase is spectrum exploration - any licensee which has unused spectrum can publish its offering on the 5GZORRO Marketplace and any entity interested in acquiring spectrum can use the 5GZORRO Marketplace as a catalogue of available spectrum. It has been stated that "the use of databases to coordinate more intensive and efficient spectrum sharing has emerged as a critical regulatory tool" [36]. 5GZORRO takes this a step further relying on DLT instead of databases benefitting from the immutability of the records and the automation afforded by smart contracts.

The second step is the actual trade, whereby an interested party can acquire the spectrum using spectokens and smart contracts. However, this second phase will benefit from AI to determine the correct price of the spectrum that is being traded.

The third phase is spectrum exploitation. Through the use of DLT involving also the participation of the relevant national authorities, 5GZORRO allows such authorities to oversee that shared spectrum still respects the obligations of the individual license. This is achieved through access in real-time to the spectrum trade and therefore allowing for effective monitoring of the spectrum assignment and remedial actions addressing non-compliance. The 5GZORRO will also allow such authorities to carry out the due diligence necessary before authorizing the spectrum trade.

The process proposed is without prejudice to additional steps that regulators might take to safeguard competition and ensure transparent and non-discriminative access to such spectrum.

11.2.3 ITU-T FG DLT

The ITU-T Focus Group on application of Distributed Ledger Technology (DLT) [37] was established in May 2017 to identify and analyse DLT-based applications and services, to draw up best practices and guidance which support the implementation of those applications and services on a global scale, and to propose a way forward for related standardization work in ITU-T Study Groups.

The high-level architecture presented in Figure 11-4 refers to the highly abstract hierarchical architecture of distributed ledgers covering almost all distributed ledgers, including public chains represented by Ethereum and Bitcoin, private chains represented by Hyperledger Fabric and non-blockchain distributed ledgers systems.

Resource and Infrastructure Functions. The infrastructure provides the operating environment and essential components required for the regular operation of the distributed ledger system. The base layer includes network services, storage services, and computing services. The layer is the resource that most software systems rely on and is the foundational support of the distributed ledger system.

Protocol and Compliance Functions. In DLT systems, each node can have its implementations based on the system's technical specifications. The protocol layer is a conceptual layer to serve the technical specification across nodes inside a DLT system. The protocol layer comprises governance/compliance, consensus, ledger

management and messaging. Moreover, governance/compliance includes node management and AAA management. Its function is to support the management of system governance (based on trust endorsement) and AAA functions of other components.

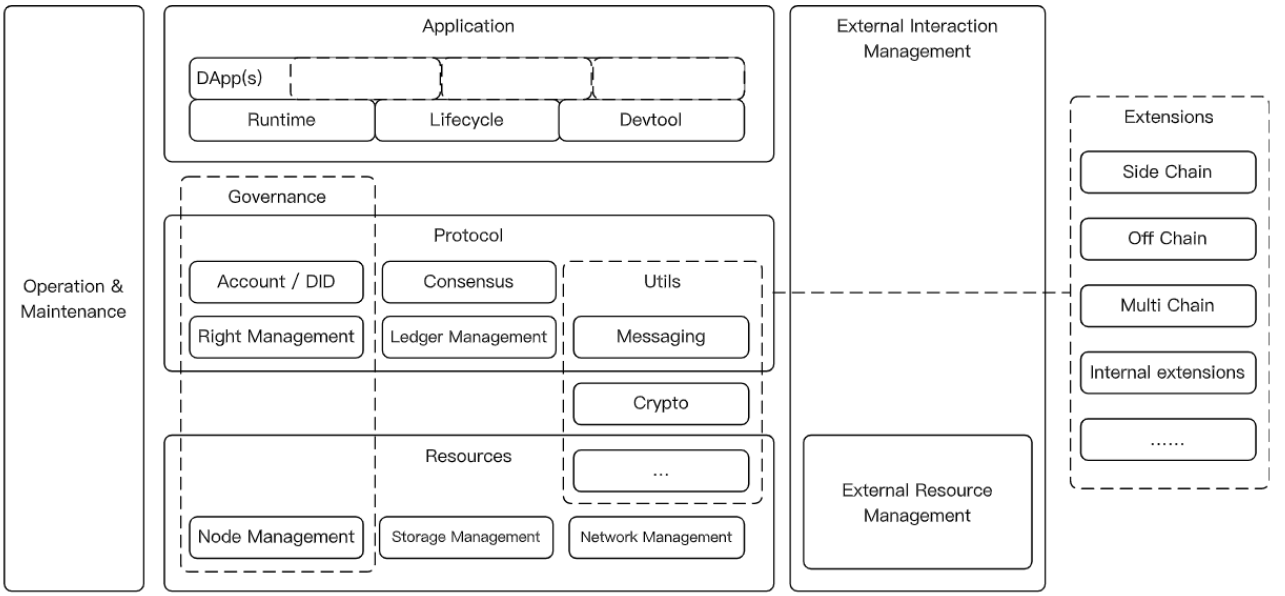

Figure 11-4: ITU-T FG DLT reference high level architecture ([37])

Decentralized Application Functions. Based on the runtime management, DApps are built to serve different business requirements in a distributed network environment. Furthermore, web applications with a combination of both off-chain services and on-chain services can be a solution for businesses.

Operation and Maintenance Functions. Operation and maintenance functions include various libraries such as log, monitoring, node/network management, and scaling libraries.

External Interaction Management Functions. Each DLT system has its network hypothesis, trust endorsement hypothesis and governance model. Thus, a DLT system with an open-network hypothesis can interact/interoperate with an external system.

Extension Functions. The extension component of a DLT platform aims to resolve different requirements of data interoperability. Extension functions include a series of protocols for data interoperations of external systems, such as multi-chain, side-chain, off-chain, or internal systems5GZORRO aims to offer a real-time decentralized market to facilitate dynamic spectrum, networking and computing resource trading capabilities. 5GZORRO leverages the tokenization of assets using smart contracts, DID and consensus DLT concepts (also inspired by ITU-T FG DLT), to establish trading operations with SLA requirements. Parties that do not trust each other can utilize 5GZORRO platform, to negotiate and set-up a new commercial relationship via a smart contract for 3rd-party resource leasing with associated SLA. Incorporation of external data sources supporting the smart contract is also envisioned and will be achieved through Oracles. A vital feature of the 5GZORRO Marketplace is a decentralized catalogue that includes the collection of 5GZORRO product offers ready to be traded between resource providers and resource consumers.

11.2.4 ETSI PDL

The ETSI Industry Specification Group on Permissioned Distributed Ledger (ISG PDL) analyses and provides the foundations for the operation of permissioned distributed ledgers, with the ultimate purpose of creating an open ecosystem of industrial solutions to be deployed by different sectors, fostering the application of these technologies, and therefore contributing to consolidate the trust and dependability on information technologies supported by global, open telecommunications networks. To understand the scope of the group, let us remark distributed ledgers can be considered as permissioned or permission-less, regarding the

requirements for a node to be approved to validate the transactions and record them on the ledger. While permission-less ledgers are the ones that have received most attention from the general public (with the paradigmatic example of Bitcoin), permissioned distributed ledgers are the ones best qualified to address most of the use cases of interest to the industry and governmental institutions. The main reasons for this are both technical (cost and delay of the recording of a transaction, cost of the consensus algorithm, fairness properties among participants…) and legal (support from external legal agreements, regulatory enforcement in critical sectors…).

The ISG PDL has started from already available experiences in the field of permissioned distributed ledgers, seeking for the definition of open and well-known operational mechanisms to validate participant nodes, support the automation of the lifecycles of the ledger and individual nodes, publish and execute operations regarding the recorded transactions through smart contracts, improve security of ledgers during both their design and operation and establish trusted links among different ledgers using these mechanisms.

The ISG PDL works in tight coordination with other groups in ETSI and elsewhere, including open-source initiatives and a clear connection with research activities, especially the collaborative research projects within the Horizon 2020 programme. PDL is committed to produce deliverables of three different natures:

- Informative, in the form of studies and recommendations for further work.
- Normative, essentially specifications and test procedures based on them.
- Demonstrative, focused on proof-of-concept reports and interoperability assessment events.

In its one year and a half of activity, the group completed a landscape document, intended to identify current activities and gaps, and a report that examines essential needs in terms of trust, security and effective conformity assessment, making recommendations on how PDL can be used by organizations, operations, deployment, hardware, and software to be trustworthy. The report also analyzes essential requirements for PDL technology to ensure regulatory compliance to preserve security and privacy in the conduits providing the data to be incorporated into the ledgers.

Work is close to complete as well on *application scenarios*, aimed to describe potential PDL operational scenarios, as much independent of particular use cases as possible, including provisioning models with special emphasis on 'as-a-service' paradigms, and PDL infrastructure governance aspects.

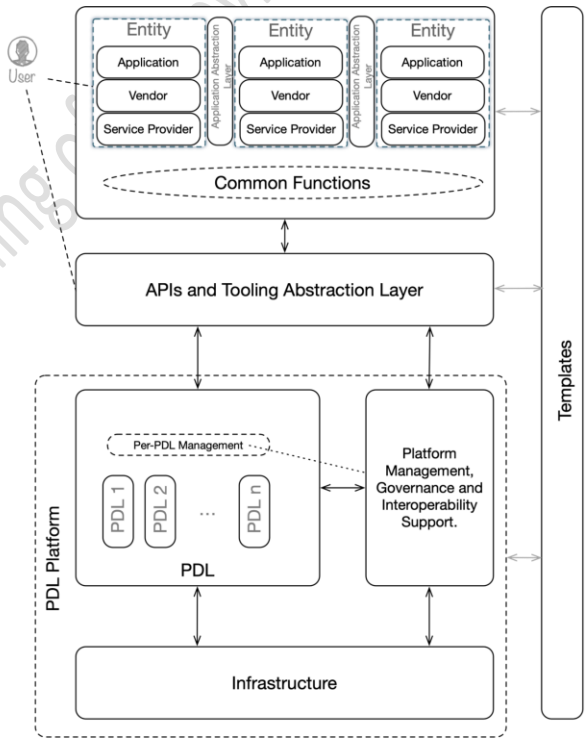

Figure 4-5: PDL Reference Framework for Application Scenarios

Furthermore, work progresses on *smart contracts* and *interoperability*. The first work item is committed to specify functional components, provide a reference architecture, describe the supported methods for planning, coding and testing smart contracts in this architecture, and a discussion of possible threats and limitations. The interoperability one is focused on describing the key elements of interoperability to exchange information between different ledgers and to mutually use the information that has been exchanged.

5GZORRO is in a good position to contribute in all active work-items. First, the current trust requirements are very much focused on industrial IoT environments, and the experience gained within the spectrum and

edge/cloud resources can be extremely valuable. A similar situation is applicable to the operational application scenarios, that are not currently tailored to the interaction with the kind of data infrastructures and dynamic trust assessment mechanisms considered in the project. Contributions in these two more mature work-items should be focused mainly on proof-of-concept execution, evaluating the assumptions already made in them, and driving the appropriate updates whenever required.

The contribution to the work-items on smart contracts and interoperability can be more direct, given the lower maturity of their results, bringing the direct implementation and validation experience accumulated in the project. Beyond this, other potential work-items can be influenced by the project since their inception. Among these, we can think of work focused on performance issues, the interaction with massive data infrastructures, or the application in dynamic trust assessment.

11.2.5 MEF LSO SONATA

The Metro Ethernet Forum (MEF) is standardizing Lifecycle Service Orchestration (LSO) Sonata APIs in an attempt to enable service automation across providers and network technology domains. LSO Sonata APIs relate to the interface reference point within the LSO Reference Architecture that supports automated business-to-business interactions between service providers. LSO Sonata APIs combine service-agnostic TM Forum Open APIs with MEF 3.0 service definitions. MEF 3.0 is a Transformational Global Services Framework for defining, delivering, and certifying assured communications services orchestrated across a global ecosystem of automated networks and includes dynamic Carrier Ethernet, Optical Transport, IP, SD-WAN, security, and other virtualized services that are orchestrated over programmable networks using LSO APIs [41].

The collection of available and planned LSO Sonata APIs deals with serviceability (address validation, site queries, product offering qualification), product inventory, quoting, ordering, trouble ticketing, contracts and billing [42]. The MEF LSO SONATA architecture is provided in the following Figure.

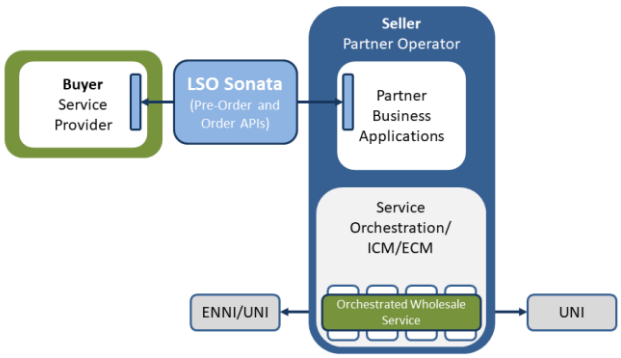

Figure 11-5: MEF LSO Sonata architecture

In the architecture of Figure 11-5: MEF LSO Sonata architecture a service provider buys resources/services from another operator to extend its service offerings to its customers. Specifically, in the Figure the seller is the provider of resources/services and the buyer is the consumer of resources/services.

The buyer orders resources/services through the use of LSO Sonata APIs, which allows a fast and automated way to a new service to its customers. Additionally, ENNI indicates denotes External Network-to-Network Interface, which is an interface representing the boundary between two operators. Likewise, UNI denotes User Network Interface, which indicates the interface to the customers of the service provider.

MEF is standardizing LSO Sonata APIs as part of a comprehensive effort to standardize multiple sets of LSO APIs enabling automation across different service providers and over multiple network technology domains. In the context of LSO, API describes the Management Interface Reference Point along with a data model, the protocol that defines operations on the data and the encoding format used to encode data according to the data model. LSO Sonata APIs relate to the interface reference point within the LSO Reference Architecture that supports business-to-business interactions between service providers. The full list of LSO Sonata APIs deals with business functionalities that are made available through a series of releases of the LSO Sonata SDK [43]. In particular, the MEF LSO Sonata SDK includes API definitions for:

1) *Address Validation*, allowing the buyer to retrieve address information from the seller - including exact formats - for addresses known to the seller.

2) *Site Query*, allowing the buyer to retrieve Service Site information including exact formats for Service Sites known to the seller.
3) *Product Offering Qualification*, allowing the buyer to determine whether it is feasible for the seller to deliver a particular Product with a given configuration to a particular geographic location.
4) *Product Quote*, which supports the inter-carrier resource/service quote process over the Sonata interface.
5) *Product Inventory*, where the buyer requests a list of resources/services from the seller based on certain criteria.
6) *Product Order*, which supports the inter-carrier resource/service ordering process over the Sonata interface.
7) *Ticketing*, allowing troubleshooting functionalities related to ordered resources/services using the Sonata interface (e.g. creating and managing tickets).
8) *Billing*, allowing to exchange usage and billing information over the Sonata interface.

From the above API definitions, only a few are stable and the rest are still under development. Among the available definitions is the Product Offering Qualification, which allows to monitor the state in which the product offering lies. Moreover, the API definition for Product Offering Qualification also allows to set the attributes of an order. Specifically, an offering is considered as in progress state, when it resides within the seller of services/resources, ready when approved by the buyer and has either been sent to or is ready to be sent to the buyer of services/resources. Further non-operational states are linked to insufficient information on the offer from the seller side or the failure to meet the buyer deadlines also from the seller side.

The Product Offering Qualification API consists of the following parts:

- Generic API framework
- Product-independent information (operations and data model)
- Product-specific information (MEF product specification data model)

These parts are illustrated in Figure 11-6.

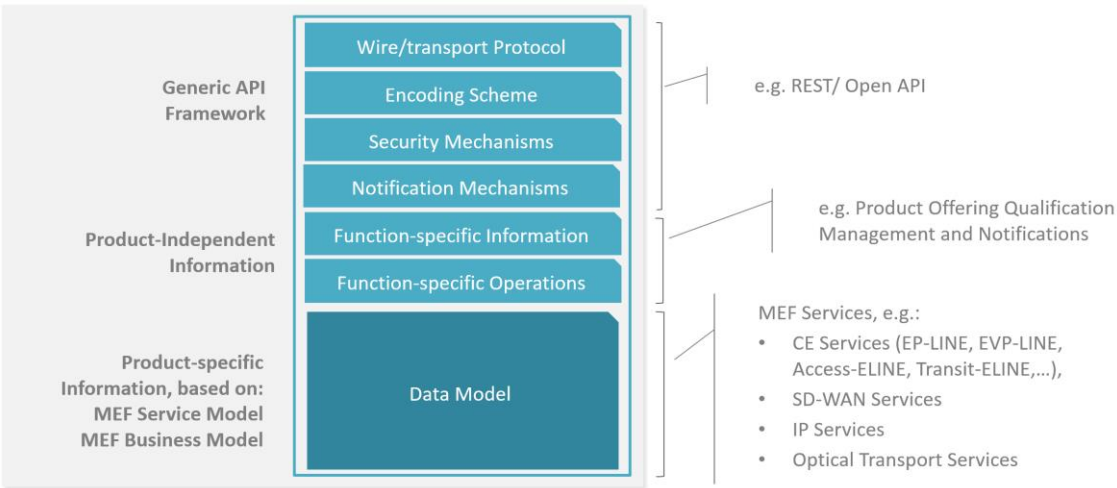

Figure 11-6: MEF LSO Sonata API definition for Product Offering Qualification

In the scope of 5GZORRO, MEF LSO Sonata can be used to provide an increased automation level in the interactions between buyers and sellers by using the catalogue of resources/services and providing API interfaces between for their interactions. In this scenario buyers are related to users of the 5GZORRO platform that are external to the 5GZORRO project and sellers the involved partners of 5GZORRO platform. Furthermore, the use of API interfaces also supports activities related to service and resource control, configuration, order, delivery and testing. An example towards this direction is the aforementioned Product Offering Qualification that can be used to provide API endpoints with offerings of services and resources from the seller towards the buyer.

Another aspect that can be considered within 5GZORRO is the exchange of resources/services between partners in the project. In this scenario 5GZORRO partners can be able to implement use-cases or scenarios using third-party resources/services that are missing from their infrastructure. MEF LSO Sonata APIs can facilitate their interactions in terms of resource/service ordering and delivery within the 5GZORRO platform.

Finally, MEF 3.0 LSO Sonata APIs can allow orchestrators that belong to different network operators to communicate and interact, in order to provide an end-to-end network as a service functionality. This concept is illustrated in Figure 11-7: MEF LSO Sonata interoperation with the Open Network Automation Platform, where the Open Network Automation Platform (ONAP) [44] that is maintained by Linux Foundation is used to orchestrate an NFVi platform to control the various virtual and/or physical network elements southbound. As a northbound interface, ONAP, using the LSO framework, is used to realize the end-to-end orchestrated, agile services that operators are looking to migrate towards.

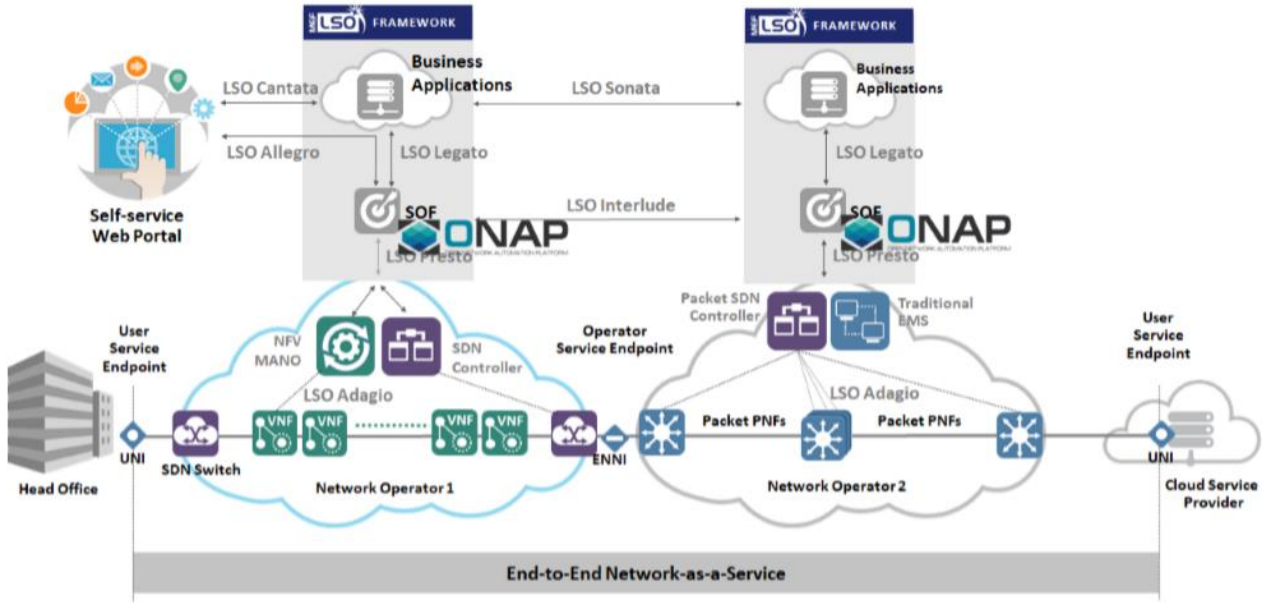

Figure 11-7: MEF LSO Sonata interoperation with the Open Network Automation Platform

11.2.6 CBAN

CBAN (Communications Business Automation Network) has born out of ITW Global Leaders' Forum (GLF) group, initially formed in 2018, whose goal was to study how DLT/Blockchain-based platforms could revolutionise and automate the inter-carrier settlement processes which was known to cost the industry billions[45]. Thus, leading CSPs and technology vendors, in the scope of CBAN, promote collaboration over competition to unlock "new revenue, savings, and services that were previously immaterial", as stated in the aforementioned official website.

While CBAN members understand the need to interconnect different a vast number of BSS (Business Support Systems) and OSS (Operations Support Systems) to achieve greater interoperability, it is also acknowledged that such a task may be practically impossible due to the individual technical complexity each may present with their own levels of requirements and inter-dependencies across different services. and the numerous silos created by ICT-SPs (ICT Service Providers). This is stated in the official whitepaper[46] and perceived as an opportunity which can only be attained by designing and developing a new architecture for an automated platform that must be interoperable with the existing aforementioned platforms. To CBAN, the adoption of such automated platforms may happen gradually over time with an ultimate goal of becoming the de-facto standard and architecture for automated and interoperable multi-network and multi-service future. The overarching goal is then to "allow ICT-SPs to trade network assets and capacity in real-time", under such a new operational and settlement framework.

CBAN's drafted reference architecture[47] has been studied and key building blocks have been identified and mapped to that of 5GZORRO's. Particularly, one MVP - CBAN's definition of a type of product or service a ICT-SP may offer as a use case - entitled Data on Demand (DoD)[48] has been carefully analysed, providing its resemblances to 5GZORRO's use cases. The DoD MVP considers a unified information modelling approach, unified processes and service definition to fully unlock the commercial potential of such service in an open and interoperable environment, so that ICT-SPs can benefit from new revenue streams from existing infrastructure owners. In turn, the DoD MVP extends the existing MEF standard LSO Reference Architecture (MEF-55)[49] to a framework which considers the different workflows of a wholesale supply chain of data services leveraging DLTs. Within such MVP, different workflows, which resembles some of those commonly present in different 5GZORRO use cases, have been identified:

- Service inquiry and quoting
- Ordering
- Provisioning
- SOAM
- Fault identification and repair
- Performance monitoring & utilization measurements
- Proposal for SLA reputation calculation
- Billing, Reconciliation & Settlement

Besides all of the above, this includes also CSPs satisfying partial orders and composing the remainder of a service's sub-elements through other partners, which is a core functionality of 5GZORRO's use case #3 (Pervasive vCDN Services).

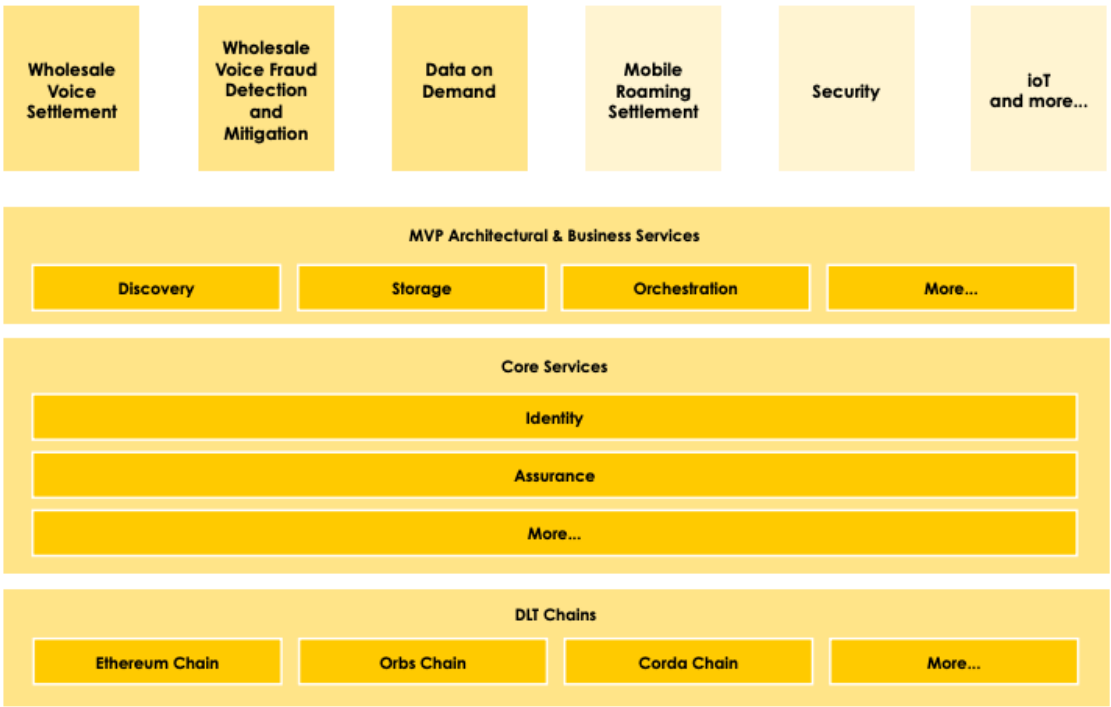

Figure 11-8: The CBAN Reference Architecture

The CBAN Reference Architecture found in Figure 11-8 above, taken from the official whitepaper[46], identifies the building blocks which provides the necessary services common to different MVPs (such as the previously identified DoD). Analysing the document in depth, different services (specifically CBAN's Core Services) found in the CBAN Reference Architecture can also be mapped to that of 5GZORRO's:

- Identity Management

- Governance
- Catalogue
- Product Catalogue (MVP services offered by a given CSP e.g. DoD)
    - Service – Composed Services that may be broken down into sub-elements
    - Elements – atomic elements of a service
    - Off-chain storage & Distributed data orchestration
- SLA Reputation
- DLT

For this reason, 5GZORRO's reference architecture is considering the findings coming already from CBAN and intends to further expand this work and contribute back as the project evolves.

11.3Area Security & Trust

11.3.1 Overview of Trust in SDO

Trust is a complex, subjective and abstract concept. In a general context, trust is understood as the assurance or honest hope that someone has about another person or some other entity. Thus, even in human relationships and interactions, it is complex to define trust and identify the elements that establish trust.

In the computer science and networking fields, trust is defined as a mechanism to evaluate, establish, maintain, and revoke the confidence relationships between devices of the same (*intra-domain*) or different networks (*inter-domain*) within one or multiple environments. Hence, prior to establishing a trust relationship between entities, it is necessary to carry out a set of phases that allow us to calculate the level or degree of trust among them. This process is known as trust computation.

Trust computation techniques [50] are usually classified into five design modules: trust composition, trust aggregation, trust formation, trust propagation, and trust update. *Trust composition module* represents the components, information, and features that are acquired from each participating entity in a trust relationship and later considered in the trust computation process, *trust aggregation module* refers to add trust evidence acquired through either self-observations (previous interactions) or feedbacks from other entities (reputation), *trust formation module* is about how to form the overall trust, based on either a single-trust or a multi-trust approach, *trust propagation module* refers to how to propagate the calculated trust degree or level (centralized and distributed), and finally, *trust update module* determines when trust should be re-calculated or updated among entities (event-driven scheme and time-driven scheme).

Despite the main steps carried out by trust calculation techniques are normally the same, there are proposals that optimize the design through the union of some modules. Sharma et al. [51] introduced a generic framework to manage trust in IoT which considers the same characteristics as the previous proposal but reduces the number of modules to four: (a) information gathering, (b) trust computation, (c) trust dissemination, and (d) update & maintenance.

Nowadays, trust is a characteristic extensively contemplated in multiple environments. In fact, more and more efforts are being made in the literature to generate a trust model that can be applied to 5G communications networks. In this sense, the 5G PPP project 5G ENSURE [52] took the first step towards a trust model where were provided a model capable of identifying primary and secondary threats for 5G networks in the context of operators sharing their network resources. Like 5G ENSURE, other proposals determine trust as a decision to accept (or not) risk arising from one or more threats, [53] addressed the lack of trust between SDN controller and network management application by means of a trust establishment framework that considers reputation, operational risk, information risk, and privacy level.

There are also approaches that consider privacy and identity in order to generate a trust modelling [54]. Privacy is a crucial characteristic in trust models due to sensitive information is utilised to generate a final trust vale, in fact, privacy-by-design is a novel approach increasingly used in recent years. In the same way, identity is a key factor that not all trust models contemplate but is beginning to be a common feature in trust models to ensure that entities participating in the trust network are authorized.

What is more, the emergence of decentralized identifiers [55], that allow in a simple and global way to perform the processes of identification, authentication, and authorization of entities [55][62] together with the use of leading technologies such as distributed ledger, is allowing the incorporation of identity in many enforcement environments, such as trust models. In consequence, multiple trust definitions can be recognised in the literature, for this reason the is no a standard pattern for defining trust in a particular scenario, this decision must be carried out based on the specific requirements of enforcement environments and entities involved.

International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)

ITU-T has been the last years working about the trust issue from different perspectives. Recommendation ITU-T Y.3052 [63] gives an overview of trust provisioning in information and communication technology infrastructures and services. According to this document, trust is a concept that can cover security and privacy. Security is considered to be the technological aspect, while privacy is considered to be the user aspect. By utilizing security and privacy mechanisms, trust can be realized in ICT infrastructures and services.

Recommendation ITU-T Y.3053 [64] introduces a framework of trustworthy networking with trust-centric network domains. It describes a trustworthy networking conceptual model that includes features of identification, trust evaluation, and trustworthy communication.

From the data perspective, ITU-T Y.3054 [65] provides a framework for trust-based media services. Recommendation ITU-T Y.3054 identifies inherent risks in existing media services and describes the necessity for trust-based media services. While trustworthiness and trust have been treated as subjective concepts, trust-based media services utilize objectified trust, which is determined by a set of trust indicators computed by collected data, such as ability, integrity, etc.

Internet Engineering Task Force (IETF)

RFC 8485 [66] proposes a standard for determining the amount of trust to be placed in a digital identity transaction. It defines the usage of a Vector of Trust (VoT) instead of a single scalar to measure the trust between entities.

The RFC 8485 specification defines four different components for each VoT: identity proofing, primary credential usage, primary credential management, and assertion presentation. This RFC expects that trust frameworks will provide context, semantics, and mapping to legal statutes and business rules for each value in each component. These vectors of trust are also compatible with OpenID Connect for identity related operations.

European Telecommunications Standards Institute (ETSI)

ETSI is actively working in network automation, ETSI ISG ZSM (zero-touch network and Service Management) [22] group is investigating in automated network and service management, and its application to future networks such as 5G.

Regarding trust, the requirement \#120 specifies that the "ZSM framework shall enable monitoring of the effects of automation functions to build trust in every stage of increased automation towards a fully automated solution". In this sense, we could say that this requirement seeks to enable a zero-touch trust framework included in the ZSM architecture.

Also, ETSI TR 103684 [67] technical report addresses existing trust service infrastructures that operate in different regions of the world and their possible mutual recognition/global acceptance. The report identifies

ways to facilitate cross recognition between EU trust services and trust services from other schemes. The trust services are based on ETSI standards and support the eIDAS Regulation (EU) No 910/2014.

ETSI TR 103684 concentrates on existing PKI-based trust services, the most prevalent across the world. The study analysed 37 global, sector and national public key infrastructure schemes and involved workshops held in 4 regions of the world in Dubai, Tokyo, Mexico City, and New York. This analysis is based on the comparison of the different trust schemes based on four areas: legal context, supervision and auditing systems, best practice, and trust representation.

International Organization for Standardization (ISO)

ISO/IEC TR 23186:2018 [68] [ISO_TR_23186] describes a framework of trust for the processing of multi-sourced data that includes data use obligations and controls, data provenance, chain of custody, security and immutable proof of compliance as elements of the framework.

Additionally, ISO/IEC AWI 27071 [69] [ISO_AWI_27071], a standard about security recommendations for establishing trusted connections between device and service, is currently on preparatory phase.

11.3.2 Applicability to 5GZORRO

In the multi-domain scenario where 5GZORRO platform is deployed, it is critical to enable secure and trusted relations between stakeholders. Similarly, it is also essential for each entity to ensure its internal assets and connections. For these reasons, one of the main objectives of 5GZORRO is to build a security and trust framework in order to enable automated trust establishment and security management.

Regarding trust, multi-stakeholder relationships are highly influenced by this factor, as a higher or lower trust level may determine if a transaction or relation between parties is triggered or not. In this sense, 5GZORRO security & trust framework leverages the usage of public and private information about the participant stakeholders to calculate and maintain a trust score for each participant stakeholder.

As trust is a subjective concept, and two stakeholders may give different trust scores to the same third stakeholder, each one of the participant stakeholders will calculate their own trust scores, utilising the data sources, rules, and algorithms that they decide. 5GZORRO security & trust framework provides the infrastructure and services to enable the calculation, such as a public data lake with monitoring information or a DLT with information regarding smart contracts between partners, but each stakeholder will utilise this trust infrastructure as it chooses, using its own criteria according to its priorities (some may give priority to security information, while others may be more concerned with performance or violation of previous contracts with other entities).

Once stakeholders' trust scores are calculated, these set of values (each entity calculates a value for each of the participants in the 5GZORRO platform) will be utilised as an additional parameter in the stakeholders relationships and actions, such as determining which service or resource to lease from the marketplace or smart contracts terms and conditions (e.g. pricing or QoS requirements).

Regarding security, the multi-stakeholder and multi-service heterogeneous environment presented by 5GZORRO implies several security requirements that should be fulfilled to guarantee the correct functioning of the platform, its services, and its assets. These requirements can be divided into:

- *Identification, Authentication, Authorization, Accounting (IAAA)*. In 5GZORRO, each stakeholder has a unique Identity used to authenticate, sign smart contracts, identify offered resources, etc. According to the decentralized nature of 5GZORRO architecture, an Identity management solution based on centralized IAAA services is not suitable as it would make the infrastructure dependant on a single entity. In this sense, 5GZORRO proposes the usage of a decentralized public key infrastructure (DPKI) to generate and manage the necessary cryptographic material required for identifying stakeholders, establishing secure communication channels, etc. Distributed identity management solutions based on Decentralized Identifiers (DIDs) and Verifiable Claims (VCs),

powered by DLTs, are combined with Authentication, Authorization, Accounting (AAA) solutions such as OAuth2.0.

- *Secure offloaded workloads across multiple domains.* One of the main services provided by 5GZORRO is the offloading of workloads based on 3rd party resources offered in a DLT-based marketplace. However, executing tasks in external infrastructures generates additional security risks that have to be solved to ensure data integrity and confidentiality. In this sense, the two main measurements proposed to secure tasks executed in 3rd party infrastructure are:
    1. Trusted Execution Environments (TEEs) for secure and isolated code execution. TEEs are isolated processing environments in which applications are securely executed irrespective of the rest of the system. TEEs provide both secure execution isolation and allow remote code and data integrity attestation. Then, TEEs are utilised by 3rd party entities to execute the tasks of their clients, ensuring that these tasks are not tampered or affected by other processes in the same system. Additionally, TEEs can be utilised by DLT Oracles to secure their actions and provide an additional guarantee to the data pushed into the DLT.
    2. Secure connection with the 3rd party provider assigned resources. To utilise the resources leased in the 3rd party infrastructure as its own, the client stakeholder must be able to connect to them as if they were in their own network. In this sense, a Virtual Private Network (VPN) solution is required to enable an encrypted connection between the infrastructures over the Internet, extending the client's private network.
- Vulnerability assessment. The detection and mitigation of security vulnerabilities is another critical security issue in a distributed multi-domain architecture like the one deployed by 5GZORRO. In this sense, 5GZORRO leverages the usage of a common Risk Assessment methodology addressing information security management. This methodology enables the organization security evaluation as well as the sharing of this information. Then, security and risk shared data can be utilised by other services, such as trust calculation.

11.4 Technology enablers

11.4.1 Distributed Ledgers & Smart Contracts

11.4.1.1 General DLT Architecture

Distributed Ledgers (also known as Shared Ledgers or Distributed Ledger Technology / DLT) are decentralized digital records systems that store copies of the same data (ledger state) on multiple nodes (devices) across a distributed system. Unlike with a traditional ledger – that would be managed & maintained by a single trusted entity – DLTs are formed of multiple nodes on a peer-to-peer (P2P) network that each replicate and store an identical copy of the ledger state and update themselves independently once consensus has been reached on a ledger update. Crucially, this means that a group of participants are responsible for the maintenance of the valid state of the ledger.

The high-level architecture from ITU-T DLT FG [37] briefly described in sec. 11.2.3 identifies the core set of elements that characterize and constrain a DLT architecture.

The resource and protocol layers provide the operating environment and components that support the normal operation of the DLT network. Node management functions allow a Node Operator to effectively manage their individual node, whilst Network & Storage functions realise the network model e.g. P2P, and distributed storage requirements of the DLT system. Discovery of nodes and establishing secure connections, data synchronization, transaction broadcasting & consensus messages ensure that neighbouring nodes can establish secure connections.

At the service & application layers we see Distributed Application (DApp) SDKs for developing applications that integrate with DLT and Smart contracts, programmable elements of the DLT that provide trusted deterministic transaction validation.

Whilst DLT implementations vary widely, we will discuss in the subsequent sections the key facets of DLT and how they give rise to trust.

11.4.1.2 DLT Transaction Model

Ultimately a DLT system is simply a decentralized mechanism to execute transactions and store the results (updated state) in a distributed system. There are two modes of approaching consensus, those being:

State mode – consensus upon states, whereby consensus is reached over a proposed ledger update (set of states) prior to a transaction being executed; an example being UTXO model (see below).

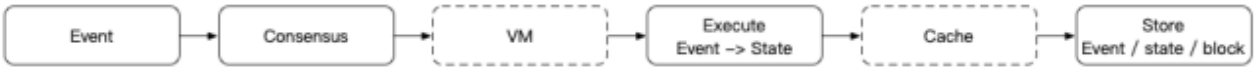

Figure 11-9: **State Mode Transaction Model - ITU-T DLT Reference Architecture [38]**

Event mode – consensus on events (transactions), whereby consensus is reached on the validity of a transaction execution result; this is the case with Account model based DLTs such as Ethereum.

Figure 11-10: Event Mode Transaction Model – ITU-T DLT Reference Architecture [38]

Below we detail two of the most prevalent transaction models used by today's DLT networks.

11.4.1.2.1 Unspent Transaction Output (UTXO) model

In the UTXO model, transactions are formed of input and output states. Input states are unspent ledger states that have been output from previous transactions. A node's wallet/vault keeps track of all of its unspent transaction output states (UTXOs), and in the case of for example Bitcoin, the sum of all these output states would represent the balance for that node/user; a private key capable of producing a valid signature for a given UTXO is required to claim ownership. Only unspent transactions can be used as inputs to subsequent transactions. Figure 11-11 depicts the relationship between transactions and how output states from previous transactions are consumed as inputs to subsequent transactions as well as creating additional output states.

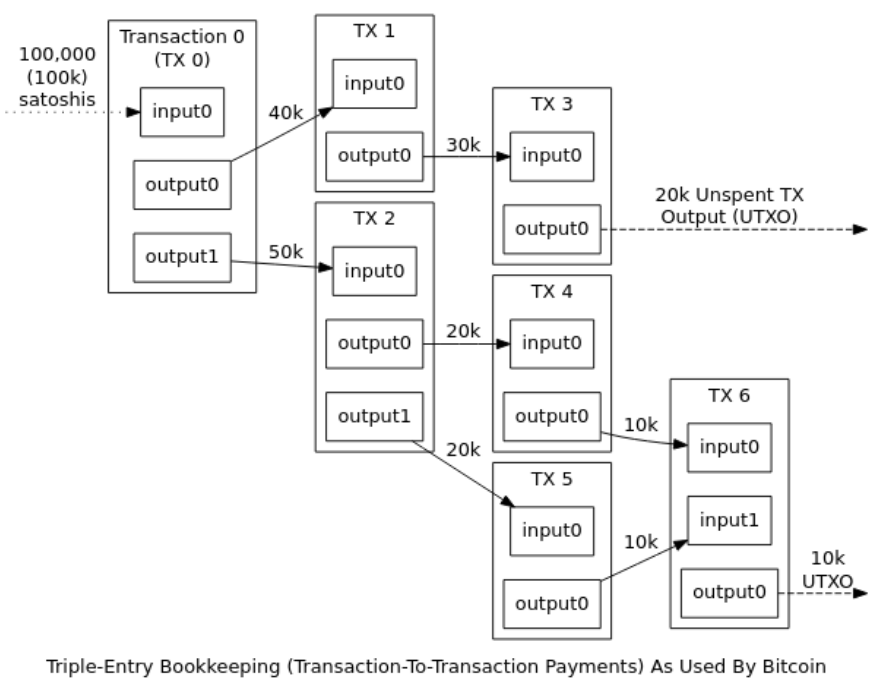

Figure 11-11: Relationship between transactions and output states in DLTs [39]

Where a Smart Contract enabled DLT utilises the UTXO model, smart contracts verify that a proposed transaction is valid i.e. the correct number of input and output states have been provided and any other business rules to verify the validity of the transaction.

Consensus around a UTXO transaction is generally to ensure that a double spend has not occurred I.e. that an input state has not already been consumed by another transaction, and that the transaction has been signed by the required parties.

The benefit of the UTXO model is that it is highly scalable because multiple UTXOs can be processed in parallel. The UTXO model also achieves a greater level of privacy as a user can use a different address per transaction if they wish. Additionally, a party will likely have many UTXOs, but only the UTXO(s) involved in each transaction are visible, therefore not divulging an account's entire balance. These benefits do come at the expense of a more complex programming model.

11.4.1.2.2 Account Transaction model

In contrast, the Account model defines a global state, stored by each node in the network. This global state stores a list of accounts, each having a balance and potentially code with internal storage that executes when a transaction is received. When making a transaction, a sending account sends the transaction to a given address and in the case of a transaction seen in Figure 11-12 if the sender's account has sufficient balance to satisfy the transaction then the specified amount is credited to the receiving account and debited from the sending account. If the receiving account has executable code, then this runs, which may prompt subsequent transactions with other accounts and updates to internal storage.

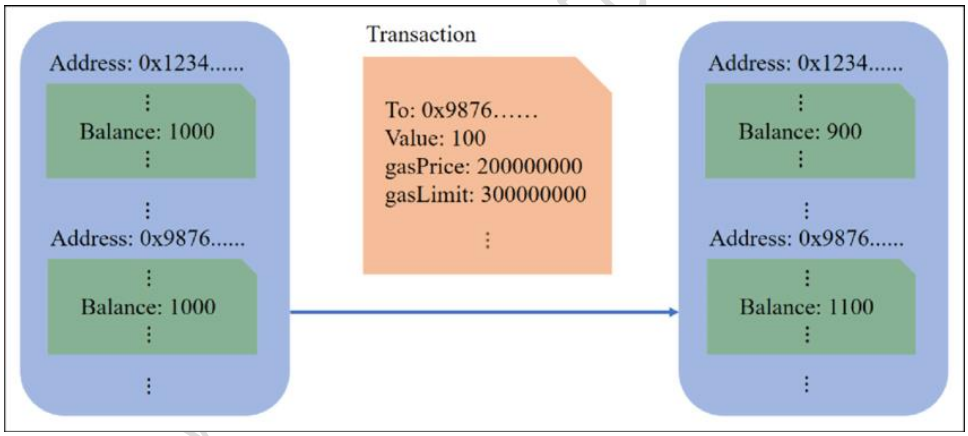

Figure 11-12: Ethereum-transaction-mode-based-on-account-model [40]

Accounts are stateful, which leads to a simpler programming model. There are also space savings when compared to the UTXO model due to a transaction only needing to make one reference and one signature that produces one output.

11.4.1.3 Smart Contracts

Smart Contracts are a DLT feature that facilitates transactions more complex than - for example - a simple transfer of monetary value from one entity to another. These are code/bytecode programs that contain business logic that are deployed on the DLT and executed as part of a transaction. Smart Contracts may encapsulate real-world legal contracts such as SLAs or simply enforce logic for processing & validating certain events agreed upon by parties in the network. Smart Contracts give rise to a greater level of trust between parties, reduce the need for trusted intermediaries and help protect against fraud losses and malicious behaviour.

As with all transactions on the DLT, each participant must come to agreement on the result of a Smart Contract's execution. Consequently, each participant will execute the Smart Contract code, which mandates that Smart Contracts must be deterministic since any level of entropy would mean that parties would

calculate different results and therefore not be able to come to agreement on the new ledger state. Should a Smart Contract require a fact or assertion during its execution that hasn't been provided as an input or already exist on the ledger, an Oracle can be used to facilitate this.

11.4.1.4 Oracles

Oracles bridge the gap between the DLT network and the outside world. Due to the mandated deterministic execution of Smart Contracts, any requirement for data or assertion required during the execution of a Smart Contract must be satisfied by an Oracle. An Oracle is simply a trusted service (or consortium of services that come to a quorum over a given fact) who provide off-chain data or attest to a fact on the ledger that requires knowledge from the outside world. A simple example being that a Smart contract may need an FX rate as part of its execution and if each execution of the Smart Contract requested this value, then likely there would be variances that would impact the afore-mentioned determinism of the contracts execution. It would request this fact from an oracle which would subsequently obtain the value from an FX service and store the 'fact' on the ledger for each Smart Contract execution to utilise. Each participant executing the Smart Contract would then be able to use a single value and come to consensus. Exactly how Oracles are implemented differs from one DLT to the next, for example Corda has built in support for Oracles, whereas Hyperledger Fabric does not.

11.4.1.5 Consensus

A consensus mechanism is required for network member nodes to agree on ledger updates, a principal element in building trust between transaction participants. This mechanism is required in order to prevent fraudulent transactions, attacks on the network and ensure entries in the ledger can be trusted by all parties who may or may not have a trusted relationship. All parties that process a transaction must come to the same conclusion about the effects of a transaction. In addition, where two transactions act upon the same data (ledger state), it is important that nodes process the transactions in the same order if they are to come to the same conclusion.

There are many consensus algorithms in use today, broadly being categorised into the following types:

- **Proof of Work (PoW)** - an algorithm whereby 'miners' compete to solve a mathematical puzzle and generate a block of transactions in exchange for native currency. It is reliable, but comparably slow and resource intensive versus other algorithms; used by Bitcoin & Ethereum
- **Proof of Stake (PoS)** - a collateral-based consensus algorithm whereby the validator has an economic stake and their vote as to the validity of a transaction is weighted based on that stake. It offers improved performance over PoW as removes the need for huge amounts of computational power; used by Ethereum Casper
- **Delegated Proof of Stake (DPoS)** - a PoS algorithm where all users vote democratically on who the final approvers of the transactions should be; used by EOS
- **Proof of Authority (PoA)** - modified version of PoS whereby identity is leveraged instead of currency. Essentially validators are staking their reputation and are chosen based on this. A much smaller set of validators are present and therefore makes the system more scalable & performant
- **Practical Byzantine Fault Tolerance (PBFT)** - derived from the classic Byzantine general's problem, consensus is reached when a minimum threshold of agreeing parties is reached, e.g. two thirds of nodes; Corda, Hyperledger Sawtooth, Ripple, Stellar
- **Directed Acyclic Graphs (DAGs)** - offer an alternative to Blockchains altogether, whereby a network of individual transactions linked to other transactions is formed. It is infinitely scalable and best suited to microtransactions with high transaction volumes; used by IOTA, HashGraoh

These consensus mechanisms are adopted across DLTs to meet the needs of the network with regards to performance, scalability and security, amongst other considerations. For enterprise applications security and performance are obviously key, making PBFT an ideal candidate.

11.4.1.6 Permission Models

The permission model of a DLT is a key decision in selection. The table below summarises DLT permission models and some of the DLTs that exemplify them:

Table 11-1: DLT permission models

		Read	Write	Commit	Examples
OPEN	**Public permission-less**	Open to anyone	Anyone	Anyone	Open Eco-systems e.g. Bitcoin, Ethereum
	Public permissioned	Open to anyone	Authorised participants	All or subset of authorised participants	Open ecosystems e.g. Ripple, Sovrin
CLOSED	**Consortium**	Restricted to an authorised set of participants	Authorised participants	All or subset of authorised participants	Multiple companies within or across sectors e.g. Hyperledger, Corda
	Private Permissioned	Fully Private or restricted to a limited set of authorised nodes	Network Operator only	Network Operator only	Internal ledger shared between parent company and its subsidiaries

11.4.1.7 DLT Requirements

As we have already discussed, there are some features common to all smart contract enabled DLTs, those being an Immutable Ledger, Smart Contracts, and Transaction & Consensus Models. For the most part enterprise use-cases dictate additional requirements that aren't satisfied out-of-the-box – currently - by public blockchains. Some of these considerations are:

- Performance
  - High transaction throughput
  - Low latency transaction confirmation
- Well-known participant identity
  - Know your customer (KYC) checks
  - Anti-money laundering (AML) checks
- Privacy & confidentiality of transactions and/or data

Consequently, DLTs offering capabilities to satisfy these industry use-cases have emerged, offering features such as:

- Private Transactions
- Private Data
- Identity Management & Role Based Access Control
- Pluggable Consensus Models
- Pluggable Storage
- Oracles

There is however an inevitable trade-off to be made as a result of introducing a level of privacy and the performance gains made by alternative consensus models. For example, the level of decentralization of the

overall network is usually reduced, with reduced transparency of ledger transactions and state; parties generally only have sight of portions of the ledger state and participation in the network and execution of consensus falls under a governance model which is generally managed by a smaller set of entities. Anonymity is a cornerstone of public DLTs, whereas under an enterprise DLT participant identification and legal culpability is generally mandated to support KYC & AML obligations.

The DLT market is fast-evolving, with new platforms emerging all the time, and similar numbers becoming obsolete for various reasons. As a result, it is important when considering a DLT that in addition to its feature-set, that it looks to have a diverse community of developers and market adoption. We have reviewed what we believe to be some of the leading Enterprise DLTs, assessing available feature-set, configurability, usability and adoption. We have focussed on open source projects that demonstrate good project health in terms of monthly commits and diversity of contributors. More detailed analysis of some of the market leaders Quorum, Hyperledger Fabric and R3 Corda has been carried out with a view to them being leading candidates for 5GZORRO.

11.4.1.7.1 Ethereum

Ethereum entered the market in July 2015 and has since grown to become one of the go-to smart contract enabled Blockchains. Due to its success it boasts a large developer network and great tooling. It is however a public permission-less DLT and as such does not satisfy many enterprise requirements out-of-the-box; most notably privacy, party identifiability and performance.

Subsequently the Enterprise Ethereum technical specifications have been developed in order to accelerate enterprise adoption of Ethereum. One promising implementation of this specification is Quorum.

11.4.1.7.2 Quorum

Quorum is an enterprise fork of Ethereum, originally developed by JP Morgan. It builds upon Ethereum, supplementing it with the following key enhancements:

- Transaction and contract privacy - public transactions are executed in the same manner as with the core Ethereum network with state being globally synchronized. Private transactions are achieved through a transaction manager, which exchanges encrypted payloads with other participant's Transaction Managers. Only those participants that have their public key identified in the transaction are permitted to view the transaction contents, all other participants will simply see a hash of the payload. Private transactions are therefore not globally synchronised and as such, consensus formed between the parties partaking in the transaction; identified by their public key.
- Pluggable consensus algorithms – RAFT or Istanbul BFT
- Network/Peer permissions management – unlike the public Ethereum network, a Quorum network comprises a group of organizations, with admin accounts able to propose new organizations and assign admin rights. Permissions are achieved through Smart Contracts and a modified P2P layer to only allow connections from permissioned nodes.
- Improved performance over the standard Ethereum network thanks to RAFT/IBFT consensus
- Fault tolerant thanks to IBFT

Whilst offering performance and fault tolerance improvements over Ethereum, Quorum is less performant than other leading Enterprise DLTs and does not offer fine grained control over permissions. Quorum utilises Solidity for Smart Contract development which, whilst being a popular development language thanks to the prevalence of Ethereum, it is still proprietary and therefore could potentially pose limitations when recruiting developers with the appropriate skill set.

11.4.1.7.3 Hyperledger Fabric

Hyperledger is a multi-project open source collaborative effort to advance cross-industry blockchain technologies, founded in 2015 by the Linux Foundation. In 2016, IBM donated what is now known as Hyperledger Fabric to the foundation, thus open sourcing it in the process. Hyperledger Fabric is an

enterprise-grade permissioned DLT developed with modularity and configurability at its heart, making it versatile for a broad range of industry use-cases.

There is an extensive community around the project which offers the following stand-out features:

- Identity Management, Role Based Access Control and Policies for governing the network
- Private Transactions achieved using Channels to allow participants to form sub-ledgers of transactions, only visible to those who belong to the channel
- Private Data, allowing when necessary for parties to retain data privacy between a selection of nodes within a channel through the use of side-channels to share the private data and hashing of the private data in the transaction that is visible to other parties in the channel
- Support for multiple mainstream programming languages for Smart Contract and client development
- Configurable Storage (LevelDB, CouchDb) to enable SQL-like ledger querying if needed
- Configurable Consensus, with Raft being the de-facto option, offering crash fault tolerant consensus
- Performant, by utilising a novel approach to ordering of transactions (introducing parallelism) and not needing to rely on conventional consensus algorithms found in public DLTs Fabric achieves significantly better performance both in terms of transaction throughput and latency (transaction confirmation)

There's no doubt that Hyperledger Fabric is a strong candidate, offering the features and performance enhancements required by an enterprise utilisation of DLT. It is not however Byzantine Fault Tolerant (BFT) out-of-the-box, which would offer protection against bad actors in the network.

11.4.1.7.4 R3 Corda

Corda was originally developed by R3, an enterprise blockchain software firm, specifically to satisfy the commercial requirement of recording and automating legal agreements between identifiable parties. In 2016, Corda was open sourced and has since cultivated a healthy developer community around it, with broad industry adoption, particularly in the financial sector.

Corda is a private permissioned DLT that offers the following stand-out features:

- Performant – thanks to utilising the UTXO transaction model and a novel approach to minimising the number of parties required to process a transaction, Corda achieves high transaction throughput; transactions and the associated state change to the ledger are shared on a need-to-know basis only
- Contract states have been developed to be legally enforceable by referencing human readable legal prose that align with the verification logic that is encapsulated by a Smart Contract
- Corda has 'Flows' which provide a means of programmatically defining multi-party, multi-step workflows for well-defined contract negotiations with check-pointing capabilities
- Corda is a containerised environment, and this enables business logic to interact with the DLT network with ease
- Identity management means that all parties within the network are have well-known identities, linked to legal entities
- Notary services (for transaction validation checks such as checking for double spends) introduce parallelism, which results in low latency and high throughput for ledger updates
- Supports Byzantine Fault Tolerant (BFT) Consensus out-of-the-box
- Strong tooling for both development and network visualisation
- Built in Oracle support

Corda offers all the features that you would expect from an enterprise-focussed DLT, particularly with regards to performance, legally enforceable contracts and best-of-breed tooling. Like Hyperledger Fabric, Corda satisfies the requirements of an enterprise use-case with the significant benefit of supporting BFT, making it a strong candidate.

11.4.1.7.5 Hyperledger Indy

Hyperledger Indy is another project under the Hyperledger Foundation and is a specialised DLT for powering decentralised identity. It provides tools, libraries and reusable components for providing digital identities rooted on DLT. Its key features are:

- DLT purpose-built for decentralised identity
- Byzantine fault tolerant
- Globally unique DIDs that are resolvable via a ledger and thus require no centralised resolution authority
- Interoperable Verifiable Credentials conforming to the W3C standard in development
- Zero Knowledge Proofs which prove a set of claims to be true, without revealing any additional information, including the identity of the 'prover'

Hyperledger Indy is generally coupled with other tooling, libraries and frameworks to achieve an identity trust layer, most notably Hyperledger Aries and Hyperledger Ursa. Aries enables trusted peer-to-peer interactions based on DIDs and verifiable credentials, where Ursa is a library that implements cryptographic work such as zero knowledge proofs to provide reusable pluggable cryptographic implementations.

It is believed that this combination of projects provides the necessary elements to realise a trusted identity layer for the 5GZORRO system.

11.4.1.8 Optimisations

11.4.1.8.1 Off-chain storage

In certain cases, data needs to be stored off-chain (I.e. not on the ledger). This may be because it is non-transactional data that cannot be stored efficiently on the ledger such as images, documents or verbose metadata that would be better suited to other storage mechanisms. In this case, each of the participants will likely need a copy of this data and as such the off-chain storage solution should be a distributed solution.

Another use case is when there is a requirement for keeping data private between say, two parties. Often this can be achieved by sharing this data via a side-channel and subsequently stored by just the permitted participants. In both cases, in place of this data existing on the ledger, a cryptographic hash is taken of the content and added to the transaction, such that it can be validated by the permitted parties before the transaction is committed to the ledger.

11.4.1.8.2 Off-chain compute

Another optimisation is to leverage off-chain compute when it is not pragmatic for a particular computation to be executed by every node. A trusted node (or quorum of nodes) will execute a particular computation based on the inputs to the transaction and each of the participating nodes will verify the result prior to it being committed to the ledger.

11.4.1.9 5GZORRO Focus

A marketplace that facilitates the free autonomous trading of resources & services underpinned by DLT will be the primary focus of 5GZORRO. DLT and Smart Contracts are the key enabler of autonomous trusted discovery, contract negotiation and provisioning of resources & services between potentially untrusting, competing entities without the need for a central authority. By treating resources and services as digital assets registered on the ledger, they can subsequently be incorporated into agreements that comprise of one or more assets.

Smart contracts will be derived from Ricardian Contracts meaning that any asset or agreement related action/event is verified & validated in-line with the business model agreed across all participants of the network.

Agreements will be derived from contract/SLA templates, which link directly to their Smart Contract counterparts. These templates comprising human readable prose and accompanying machine-readable parameters should be subject to a rigorous governance process, further adding weight to their legal enforceability and consistency across the 5GZORRO system. The full contract lifecycle of Discovery & Contract

Negotiation, Deployment, Monitoring, Billing & Enforcement and finally Termination will be underpinned by Smart Contracts, which will ensure that the agreed terms are enforced and enacted during the lifetime of the active contract.

A key focus aspect of the 5GZORRO system will be to automate SLA enforcement during the monitoring phase. Smart contracts will receive measurements at intervals from a service agreed within the terms of the contract. Should a breach be detected the smart contract will trigger subsequent actions based on what has been agreed; this may be termination of the contract for example.

As previously mentioned, the DLT market is still in its infancy, making it volatile with technologies arriving and disappearing. It is therefore also a key focus that we ensure the 5GZORRO system is architected in such a manner that protects against vendor lock-in and moreover to not dictate a particular DLT for 5GZORRO.

Ultimately, the utilization of DLT and Smart Contracts is the key enabler to realise trust and autonomy across the system. A tamper-proof immutable ledger of temporally ordered state changes governed by Smart Contracts will mean that participants can trust that their interactions are being processed in accordance with the agreed model. Additionally, this means that all actions are non-repudiable and attributable to a well-known legal identity which is imperative in this commercial context.

11.4.2 Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs)

With the arrival of 5G networks and the massive expansion of interconnected devices, identity management is an essential process when it comes to identify, authenticate, and authorize thousands of users, devices, resources, services, organizations, etcetera. In particular, decentralized identity management (DIdM) is a novel paradigm that is becoming an interesting research area due to its integration with novel distributed ledger technologies (DLTs) such as blockchains.

Related to decentralized identity management, Decentralized Identifiers (DIDs) are a novel type of identifiers proposed by W3C that allows associating any subjects such as stakeholders, resources, services, organizations, entities, and so on, with a digital identity.

Concretely, Decentralized Identifiers (DIDs) [55] are global identifiers which enable verifiable and decentralized digital identity, allowing to uniquely identify any subject, e.g. a person, organization, abstract entities, etc. To achieve this purpose, DIDs are associated with cryptographic material, such as public keys, and service endpoints, making each DID globally unique, resolvable with high availability, and cryptographically verifiable.

The usage of DIDs provides to an application of self-administered identity management, enabling further self-managed capabilities such as authentication, authorization, role management, and identity information exchange between two identity domains.

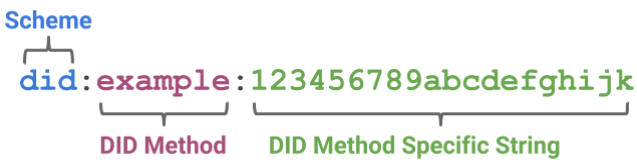

Figure 11-13: DID Format example [55].

Taking a closer view, a DID is a portable URL-based identifier generated as a string, similar to "*did:5g-zorro:123456789abcdefghi*", consisting of three parts: URL scheme identifier, DID method identifier, and DID method-specific identifier. Normally, DIDs associate a DID Subject, the DID's owner, with a DID Document, the important metadata related to a decentralized identifier that can be utilized to authenticate it or to verify its relationship with the DID. Among metadata registered in a DID Document, we can find service endpoints, public keys as well as other attributes or claims detailing *DID Subject* or *DID delegate* characteristics. These

attributes are made up of a generic format of DID, usually a JSON schema, that is declared in the *DID Core specification*. The official syntax of a decentralized identifier, also known as *DID scheme*, is declared in a *DID method* specification.

DID infrastructure can be thought of as a global and decentralized key-value database in which the database is all DID-compatible with DLTs, blockchains, and decentralized networks, allowing creating and managing their own identifiers on any number of independent, distributed trust roots. In this virtual database, the key is a DID, and the value is a DID document. The purpose of the DID document is to describe the public keys, authentication protocols, and service endpoints necessary to bootstrap cryptographically-verifiable interactions with the identified entity.

Another concept related to decentralized identity management is Verifiable Credentials, a Verifiable Credential (VC) [56] is a tamper-evident and privacy-preserving credential (set of claims) that can be demonstrated through a cryptographic process. Verifiable Credentials can represent the same information that physical credentials represent in real life such as driving licenses, passports, health insurance card, and so on. Therefore, Verifiable Credentials represent statements made by an issuer in a tamper-evident and privacy-preserving manner.

In this context, DIDs are utilized to associate an assertion made about a subject (claim) with the subject itself, so the credential can be transferred between domains or entities without needing to reissue it. Then, DIDs enable VCs to be shared rapidly to establish trust at a distance. Associated with VCs, the main data model concepts are Claim, Verifiable Credential and Verifiable Presentation:

- A claim is a statement about a subject, and the subject can receive one or more claims.
- A verifiable credential is a set of claims made by the same entity. It cryptographically demonstrates who issued it, this is, its authenticity, integrity, and non-repudiability.
- A verifiable presentation represents data packaged from one or more verifiable credentials.

A VC-enabled environment is composed of the following actor roles. Besides, their interactions and action flows are depicted in Figure 11-14.

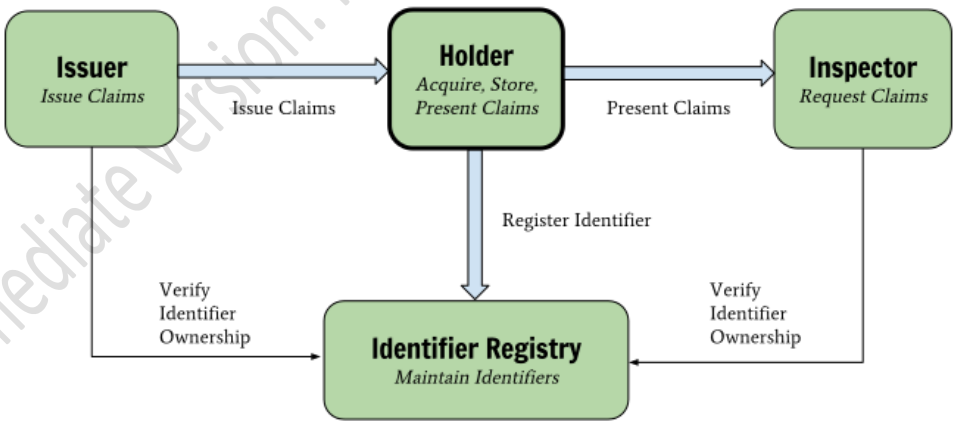

Figure 11-14: Roles and information flows in the basic Verifiable Claims architecture.

- Holder. It acquires verifiable claims from an Issuer and selectively provide them to Verifiers. The Holder is often, but not always, the Subject of the claims. Example holders include students, employees, and customers.
- Issuer. It asserts claims about one or more subjects, creating a verifiable credential from these claims, and transmitting the verifiable credential to a holder. Example issuers include corporations, non-profit organizations, trade associations, governments, and individuals.

- Verifier or Inspector. It receives one or more verifiable credentials for processing. Example verifiers include employers, security personnel, and websites.
- Identifier Registry. It mediates the creation and verification of identifiers, keys, and other relevant data, such as verifiable credential schemas, revocation registries, issuer public keys, and so on, which might be required to use verifiable credentials.

In this sense, the principal dissimilarity between DID Documents and VCs is that DID Documents are stored in the Blockchain while VCs are stored in a subject decentralized repository, for example. Thus, DID Documents must contain public data since could be accessed by everyone, whilst VCs are only consulted by entities that have the necessary permissions.

In sum, the main benefits provided by DIDs and VCs are identity control, privacy, security, discoverability, interoperability, portability and extensibility, among others. Both technologies enable to build a cross-domain Identity Management solution supported by DLT technologies.

11.4.2.1 Implementations

Hyperledger Indy

Hyperledger Indy [57] provides tools, libraries, and reusable components for providing digital identities rooted on blockchains or other distributed ledgers so that they are interoperable across administrative domains, applications, and any other silo.

Sovrin

The Sovrin Foundation [58] is a public utility for identity, built on top of Hyperledger Indy. It uses a Sovrin token as a built-in incentive for the privacy-preserving value exchange of digital credentials.

Cordentity

The Cordentity [59] app integrates Hyperledger Indy capabilities into the Corda platform. According to its documentation, Cordentity is a self-contained CorDapp that integrates Hyperledger Indy, for decentralized identity, with the R3 Corda Platform. This lab creates interoperability of two purpose-built ledger technologies, each with a focus on privacy. Corda is designed to enable private transact and Indy is a ledger built specifically for self-sovereign identity. The integrated platforms reinforce each other and allows building complicated scenarios.

Hyperledger Aries

Hyperledger Aries [60] is a client focused Infrastructure for blockchain-rooted, peer-to-peer interactions. Using source code base coming from Hyperledger Indy, it provides a shared, reusable, interoperable tool kit designed for initiatives and solutions focused on creating, transmitting and storing verifiable digital credentials. In early stage (Incubation).

11.4.2.2 5GZORRO Focus

In 5GZORRO scope, DIDs and VCs, together with DLTs, enable a multi-stakeholder decentralized identity management solution. Concretely, these technologies comply with the idea of generating an access token in the Trust & Security Framework to regulate access control to information stored by stakeholders, enabling cross-domain AAA (*Authentication, Authorization and Accounting*).

Besides, DIDs can be employed to identify the resources offered in the 5GZORRO marketplace, linking the available offers with the resource provider and ensuring its non-repudiation. In the same line, DIDs can also identify VIM or NFVI components running in 3rd party infrastructures. Finally, the cryptographical keys utilized for DID and VC verification can be also employed to generate secure connections using asymmetric cryptography communication protocols.

11.4.3 Data Lakes

Data lakes and data warehouses are both widely used for storing big data. A data lake is a vast pool of raw data, the purpose for which might not yet be defined. A data warehouse is a repository for structured, filtered data that has already been processed for a specific purpose. As opposed to data warehouse technology, data lakes do not require the entering data to be uniform or structured in any predefined way. Instead, ingested data is structured and unified inside the data lake's internal data processing engine that prepares it for consumption by data users, e.g. analytics and algorithms that implement the required business logic. Usually, the business case that the data lake serves dictates the requirements, such as the richness and the scale of the data and the data sources, the structure of data processing pipeline, and the way prepared data is fed to consuming components. In addition to storing raw data, a data lake typically provides services to process and analyse its data. A typical pipeline for data processing in a data lake may include the following stages [70]:

- Data ingestion
- Data curation and cleaning
- Feature generation (including metadata enhancement)
- Statistical or learning model
- Root cause and feature isolation
- Dashboard, notifications and other actions

Many commercial Cloud Providers provide Data Lake services.

- **Google - Google Cloud Storage** [71]
- **Oracle** - The data lake is a combination of object storage plus the Apache Spark™ execution engine and related tools contained in Oracle Big Data Cloud [72].
- **Microsoft - Azure Data Lake Storage Gen2** is a highly scalable and cost-effective data lake solution for big data analytics [73].
- **Amazon - AWS Lake Formation** [74] is a service that makes it easy to set up a secure data lake in days. A data lake is a centralized, curated, and secured repository that stores all your data, both in its original form and prepared for analysis.
- **Cloudera** [75] - provides a software platform for data engineering, data warehousing, machine learning and analytics that runs in the cloud or on premises.
- **Zaloni** [76] - provides data management software and solutions for big data scale-out architectures, such as Apache Hadoop, Amazon S3, Microsoft Azure, and Google Cloud.
- **Teradata** [77] - is an enterprise software company that develops and sells database analytics software subscriptions. The company provides three main services: business analytics, cloud products, and consulting.
- **Impetus** [78] - is a software products and services company focused on creating powerful and intelligent enterprises through deep data awareness, data integration and advanced data analytics.
- **Redhat Open Data Hub**[79] - is an open source community project that implements end-2-end workflows from data ingestion, to transformation to model training and serving for AI and ML with containers on Kubernetes on OpenShift.

11.4.3.1 Open Data Hub

As an example of a data lake, we present in a little detail the Open Data Hub (ODH). The information here is taken from [79].

The basic pipeline is depicted in the following Figure 11-15, followed by a brief description of the main components.

Ceph [80] is an open-source software storage platform, implements object storage on a single distributed computer cluster. Ceph delivers object, block, and file storage in one unified system.

Apache Spark [81] is a unified analytics engine for large-scale data processing. Applications send tasks to executors using the SparkContext and these executors run the tasks on the cluster nodes to which they are

assigned. Distributed parallel execution as provided by Spark clusters are typical and essential for the success of AI/ML workloads.

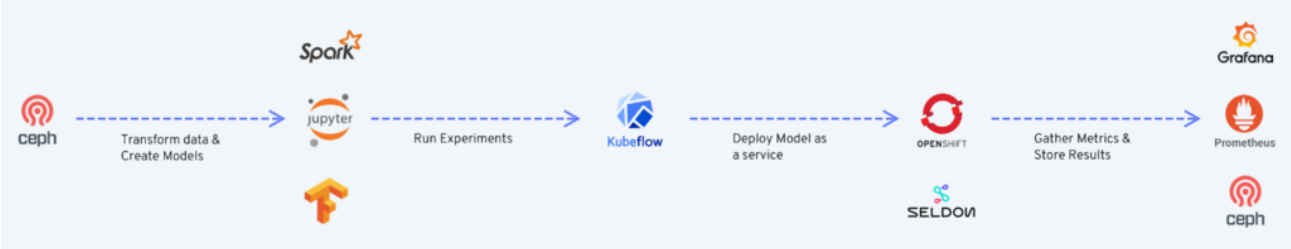

Figure 11-15: Basic pipeline for Open Data Hub

JupyterHub [82] is an open source multi-user notebook platform that ODH provides with multiple notebook image streams that incorporate embedded features such as Spark libraries and connectors. JupyterHub provides many features such as multi-user experience for data scientists allowing them to run notebooks in their own workspaces. Authentication can also be customized as a pluggable component to support authentication protocols such as OAuth. Data scientists can use familiar tools such as Jupyter notebooks for developing complex algorithms and models. Frameworks such as numpy, scikit-learn, Tensorflow and more are available for use.

Prometheus [83] is an open source monitoring and alerting tool that is widely adopted across many enterprises. Prometheus can be configured to monitor targets by scraping or pulling metrics from the target's HTTP endpoint and storing the metric name and a set of key-value pairs in a time series database. For graphing or querying this data, Prometheus provides a web portal with rudimentary options to list and graph the data. It also provides an endpoint for more powerful visualization tools such as Grafana to query the data and create graphs. An Alert Manager is also available to create alert rules to produce alerts on specific metric conditions.

Grafana [84] is an open source tool for data visualization and monitoring. Data sources such as Prometheus can be added to Grafana for metrics collection. Users create Dashboards that include comprehensive graphs or plots of specific metrics. It includes powerful visualization capabilities for graphs, tables, and heatmaps. Ready-made dashboards for different data types and sources are also available giving Grafana users a head start. It also has support for a wide variety of plugins so that users can incorporate community-powered visualisation tools for things such as scatter plots or pie charts.

Apache Kafka [85] is a distributed streaming platform for publishing and subscribing records as well as storing and processing streams of records.

Seldon [86] is an open source framework that makes it easier to deploy AI/ML models on Kubernetes and OpenShift. The model can be created and trained using many tools such as Apache Spark, scikit-learn and TensorFlow. Seldon also provides metrics for Prometheus scraping. Metrics can be custom model metrics or Seldon core system metrics.

In 5GZORRO, we build our data lake based on open-source components, similar to Open Data Hub. We need storage, messaging, monitoring, analytics, etc, and it is most reasonable to build upon the existing tools. Some particular APIs and services may have to be introduced to support zero-touch, but the basic data lake services will be taken over directly from existing solutions. There are several pipelines that are specific to 5GZORRO. One example of a 5GZORRO-specific pipeline is:

1. Provide relevant monitoring data;
2. Aggregate monitoring data;
3. Perform analytics to predict violation of SLA;
4. Perform action upon prediction of SLA violation.

Another 5GZORRO-specific pipeline surrounds resource discovery and sharing of resources between Operators.

11.4.4 Artificial Intelligence solutions for Network Management

The management of mobile networks is primarily concerned with the control, administration, and orchestration of network components functionalities to reach the desired system state, according to a set of high-level business goals and objectives. Since the 1990's network management has come a long way from admins manually interpreting the data and performing various network configurations using a set of standard interfaces to using software agents and active networks in bringing programmability within network devices to autonomic network management for self-managing or self-organizing or self-governing networks based on predefined conditions defined by an admin to cognitive network management for predicting network states beforehand using Artificial Intelligence/Machine Learning techniques. Machine Learning algorithms can be classified into three main categories:

Supervised Learning [87]. In the supervised learning, the output is known beforehand, i.e., there is a mapping between input and the output. Therefore, to design a model, the machine is fed with many training input data samples. The training data assists in reaching a level of accuracy for the designed data model. The created model is now ready to be fed with new input data for predicting outcomes. Some examples of supervised learning algorithms are Decision Trees, K-Nearest Neighbour, Linear Regression, Support Vector Machine and Neural Networks.

Unsupervised Learning [88]. In the unsupervised learning, there is no mapping between input and the output, i.e., the target values are unlabelled. The model has to detect the hidden patterns from the data input to it and learn the mapping by itself using data mining techniques. Some examples of unsupervised learning algorithms are K-means clustering, KNN clustering, Apriori algorithm, FP-growth algorithms and Neural Networks.

Reinforcement Learning [89]. In reinforcement learning, the algorithm learns using a feedback mechanism based on past experiences and each step in the algorithm is taken to reach a desired goal. In each step the algorithm receives the feedback from the previous step, based on the learning experience and predicts the next best step. Basic reinforcement learning is also referred to as Markov Decision Process. Some examples of Reinforcement Learning algorithms are Q-Learning, Deep Adversarial Networks and Temporal Difference.

In recent years, along with the centralized machine learning techniques, the decentralized machine learning techniques such as Federated Learning [90] have come to limelight. Federated Learning aims at training a machine learning or deep learning algorithm, across multiple local datasets, contained in decentralized edge devices or servers holding local data samples, without exchanging their data — thus addressing critical issues such as data privacy, data security, and data access rights to heterogeneous data. This approach of Federated Learning is in contrast to traditional centralized learning techniques where all data samples are forwarded to a centralized server and also to classical distributed machine learning techniques, which assume that the local data samples are identically distributed and have the same size.

Below are some of the existing commercial and open-source platforms that support automation of network management and orchestration in the context of 5G networks and beyond:

Open Network Automation Platform (ONAP) [44]. ONAP is a comprehensive platform for orchestration, management, and automation of network and edge computing services for network operators, cloud providers, and enterprises. Real-time, policy-driven orchestration and automation of physical and virtual network functions enables rapid automation of new services and complete lifecycle management critical for 5G and next-generation networks.

Nokia's Network Operations Master [91]. Nokia Network Operations Master is a new software based on cloud-native architecture for managing 4G and 5G networks with extreme automation. It boosts the productivity and efficiency of network operations by maximizing resource utilization with AI-powered end-to-end monitoring, control and orchestration.

Ericsson Automated Network Operations [92]. Ericsson Automated Network Operations has a portfolio of autonomous network offerings that enables better network performance and scaling across physical and virtual resources, as well as simplifying the way networks are run in terms of time and efforts required. The analytics applications allow to leverage untapped data into actionable insights and real-time decisions, ensuring an efficient customer experience leading to closed-loop, zero-touch operations and orchestration.

Below are some of the machine learning tools/platforms that simplifies designing AI workflows for end-users:

Acumos AI [93]: Acumos AI is a platform and open source framework that makes it easy to build, share, and deploy AI apps. Acumos standardizes the infrastructure stack and components required to run an out-of-the-box general AI environment. This frees data scientists and model trainers to focus on their core competencies and accelerates innovation. It packages tool kits such as TensorFlow [94] and SciKit Learn [95] and models with a common API that allows them to seamlessly connect. It leverages modern microservices and containers to package and export production-ready AI applications as Docker files and Includes a federated AI Model Marketplace – a catalog of AI models contributed by the community that can be securely shared.

Microsoft Azure Machine Learning [96]: Azure Machine Learning empowers developers and data scientists with a wide range of productive experiences for building, training, and deploying machine learning models faster. It accelerates time to market and foster team collaboration with industry-leading MLOps—DevOps for machine learning. It innovates on a secure, trusted platform, designed for responsible ML. It offers best-in-class support for open-source frameworks and languages including MLflow, Kubeflow, ONNX, PyTorch, TensorFlow, Python, and R.

Google AI Hub [97]: Google Cloud's AI Hub provides enterprise-grade sharing capabilities, including end-to-end AI pipelines and out-of-the-box algorithms, that let your organization privately host AI content to foster reuse and collaboration among internal developers and users. Enterprise users can find AI components built by other teams within an organization and access AI content published by Google AI, Google Cloud AI, and Google Cloud partners. It allows to easily deploy unique Google Cloud AI and Google AI technologies for experimentation and production on Google Cloud and hybrid infrastructures.

Amazon SageMaker [98]: Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy ML models at scale. It removes the complexity from each step of the ML workflow so you can more easily deploy your ML use cases, anything from predictive maintenance to computer vision to predicting customer behaviours. It allows users to choose from TensorFlow, PyTorch, Apache MXNet, and other popular frameworks to experiment with and customize machine learning algorithms.

In the context of the 5GZORRO project, below we mention the most relevant network management and orchestration automation operations that we will be focusing on and striving towards achieving before the end of the project.

Horizontal and Vertical VNF Auto-scaling: Resource scaling could be either horizontal or vertical. In horizontal scaling (i.e., scaling in/out), the smallest resource unit is the VNF (e.g., running on a container or a Virtual Machine (VM)), and new VNFs are added or released as needed. In contrast, vertical scaling (i.e., scaling up/down) changes the resources assigned to an already running container or VM, for example, by increasing or decreasing the allocated CPU. In VNF autoscaling, there is a trade-off between cost and Quality of Service (QoS). More VNF instances or CPU resources need to be allocated to guarantee QoS, but allocating more resources increases the cost. Therefore, the auto-scaling mechanism must be aware of the economic costs of its decisions to reduce the total expenditure but maintaining an acceptable QoS as agreed in the Service Level Agreement (SLA) between the end-user and the application provider (e.g., a round-trip-time). We will design different machine learning and deep learning algorithms using Feed Forward Neural Networks (FFNN), Long Short Term Memory (LSTM) networks, and Convolutional Neural Networks (CNN) for predictive

VNF auto-scaling, in interims of single and multi-step predictions. Furthermore, we will use the Federated Learning approach to design the algorithms to preserve the privacy of user data considering the multi-domain nature of 5G networks and compare their performance against the centralized approach.

Network Slice resource Auto-scaling: Network Slicing offers new revenue opportunities to Communication Service Providers (CSPs) to simultaneously lease virtual networks to multiple verticals. Nonetheless, CSPs also need to ensure the fulfilment of Service Level Agreements (SLAs) for each of those network slices. However, resource requirements (e.g., CPU, memory, resource blocks) for each network slice may vary and fluctuate over time, depending on their Quality of Service (QoS) requirements. Therefore, utilizing the wealth of data collected by the 5G networks, Artificial Intelligence-based proactive auto-scaling of network slice resources allows CSPs to meet customer SLAs as well as to use scarce resources (e.g., CPUs in MEC nodes) optimally.

End-to-End SLA violation prediction: Anomaly detection techniques can be used to predict anomalies in the network (e.g., unexpected traffic load peaks, security breaches) before it happens and thereby informing CSPs and allowing their MANO platforms to satisfy future network requirements efficiently.

Smart Resource Discovery: API's (also incorporating intent based technologies) that are exposed by the 5GZORRO platform will be leveraged by AI-driven business agents to perform smart resource discovery and smart resource selection. AI-based agents will be mostly based on data clustering techniques where resources of certain category (e.g., based on price) will be grouped together and published to the resource consumer.

11.4.5 Trusted Execution Environments

Trusted Execution Environment is a technology used to provide a tamper-resistant processing environment that runs on a separation kernel. Such a kernel enables systems with different levels of security to coexist on the same platform. The TEE can resist both software attacks and physical attacks performed on the main memory of the system. Furthermore, attacks performed by exploiting backdoor security flaws or system vulnerabilities are also avoided using a TEE.

With respect to its functionality, the TEE divides the system into trusted and untrusted partitions that are isolated. These partitions use a secured interface for inter-partition communication to ensure that no lateral movement can occur if the untrusted partition is compromised. Currently the secure interfaces is implemented using three main mechanisms: 1) GlobalPlatform TEE Client API [99], 2) secure RPC (Remote ProcedureCall) of Trusted Language Runtime [100]; and 3) real-time RPC of SafeG [101].

Three categories of TEE implementations are usually found: a) at the hardware level, b) at the software level and c) integrated hardware-software solution. Hardware-based TEEs are using enforced isolation that is built into the CPU. A characteristic example is the Arm TrustZone technology[1]. Extensions to this technology allow to secure the memory area. Hardware-based TEE allows trusted applications to have complete access to the main processor, resources (e.g. peripherals as sensors, actuators) and memory, while hardware isolation protects them from untrusted applications running on the main operating system.

On the other hand, software-based TEEs are providing a normal and a secure partition at the device firmware level, where both the kernel and the operating system are encrypted with strong cryptographic mechanisms to protect the data and applications of each individual device. Access to the keys for encrypting/decrypting the data is provided only to trusted entities through proper authentication or authorization schemes. A characteristic example of software-based TEE is the Open Portable Trusted Execution Environment (OP-TEE)[2]. The main difference with hardware-based TEEs is that they do not provide protection schemes for hardware resources, hence applications may only rely to the trusted kernel and not the processor and resources.

[1] https://developer.arm.com/ip-products/security-ip/trustzone
[2] https://github.com/OP-TEE

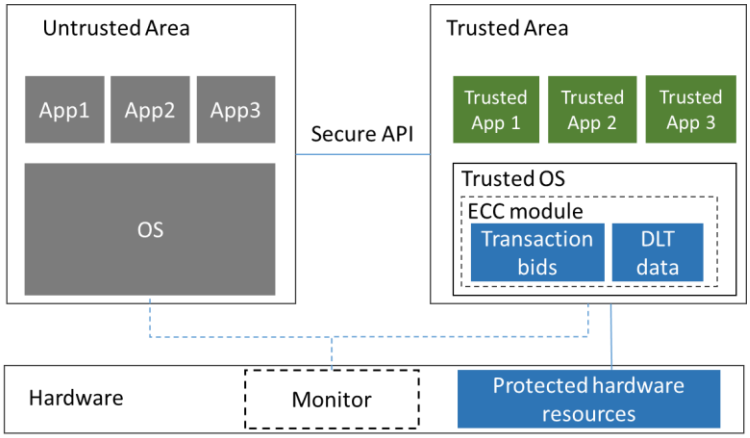

Figure 11-16: Trusted Execution Environment overview

An overview of the TEE is illustrated in Figure 11-16. The figure depicts an untrusted as well as a trusted area, which communicate using the secure API. The Trusted Area uses kernel encryption and Elliptic Curve Cryptography (ECC) to protect the data inside the operating system. Furthermore, the trusted applications are meant to handle confidential information such as credit card PINs, private keys, customer data, DRM protected media, etc. and provide services to the normal world OS to make use of the confidential information without compromising it. Finally, the dedicated hardware is used to protect the associate hardware resources as well as to provide an optional monitor for switching between the secure and normal environment.

Available TEE platforms. Several open-source and proprietary TEE platforms exist that are documented in literature and are available as market or community solutions. These implementations are separated into the categories of the previous section. Along with each platform, a brief description of its offerings is also provided.

1. **Hardware-based TEEs**
    a. *Arm's TrustZone* (proprietary): technology offers an efficient, system-wide approach to security with hardware-enforced isolation built into the CPU. Genode TEE is based on this technology and extends it through a hypervisor environment that allows to switch between secure and normal world.
    b. *Intel SGX* (proprietary): is a set of security-related instruction codes that are built into some modern Intel CPUs that could be used to implement a TEE.
    c. *Intel Trusted Execution Technology* (TXT) (proprietary): provides a root of trust and verifies the integrity of a platform by relying on a TEE. During boot, it performs measurements on the platform components (boot loader, firmware, hypervisor, operating system) and verifies them against pre-calculated white list values.
2. **Software-based TEEs**
    a. *OP-TEE* (open-source): OP-TEE comprises of a secure world OS, normal world client, test suite and Linux driver.
    b. *TRUSTED LITTLE KERNEL* (open-source): NVIDIA defines a software-partitioned, environment that provides trusted operations, a Monitor for switching between the secure and normal environment and secure storage.
3. **Integrated TEEs**
    a. *Trustonic* (proprietary): An offering that allows devices to be embedded with a Trusted Identity. This is achieved by combining a secure OS (kernel and memory) along with a hardware-secured environment (ARM TrustZone chip).
    b. *Solacia SecuriTEE* (proprietary): it is a hardware environment that provides security service for processor, peripherals, and storage device through that runs on top of the ARM

TrustZone chip. Additionally, the hardware environment is coupled with a secure kernel, which contains a trusted core environment, trusted function and an integrated API for communication with the untrusted partition. The API adheres to the GlobalPlatform TEE mechanism (described in the previous section).

5GZORRO focus. A primary focus aspect of TEE's in 5GZORRO is for protecting the data that are stored locally in each DLT node. Such data are usually replicated in the DLT nodes and if one amongst them is compromised, malicious users can gain access on its local data and applications. Additionally, they can also leverage the compromised nodes as an entry point for accessing further nodes in the DLT system using lateral movement techniques. The use of a TEE allows to maintain the transparency of the DLT, whist ensuring the privacy of data and applications. This can be ensured by an ECC module that encrypts the data associated with each transaction and which stored locally in each node (illustrated in Figure 11-16). Apart from the data, keeping the bids secret is also of primary importance, so that neither another DLT node nor any other party can learn anything about them. Hence, encryption is also added to the transaction bids and the key to decrypt them resides only inside the trusted environment. In this way whenever each DLT node commits a bid in the blockchain, it is always encrypted and only trusted recipients that have the key can decrypt the bid to visualize the underlying transaction. Trust can be established using dedicated authentication or authorization methods. Encryption of the transaction bids ensures both privacy and trustworthiness of data being exchanged between different DLT nodes.

Another focus aspect in 5GZORRO is the integration of a TEE in the VIM or NFVI components. To the best of our knowledge, a possible approach for this aspect is described in [102]. Specifically, this work describes that VIM trusted execution is achieved for controller nodes by allowing isolation both at the hardware and the software level. Isolation allows to protect the sensitive data of applications and services that are running on them. Specifically, a trusted VIM usually includes the following services: 1) secure boot process of both the controller and the compute nodes, 2) authentication process initiated by the controller node towards the compute nodes for avoiding man-in-the-middle attacks and 3) kernel integrity check for both the controller node and the compute nodes to avoid the corruption of the kernel memory. An example of an integration of a trusted OP-TEE environment in the OpenStack VIM platform is illustrated in Figure 11-17.

This example illustrates the communication between the OpenStack controller (situated in the untrusted area or normal world part as depicted in Figure 11-17) and the compute nodes that are located in the trusted area or secure world part.

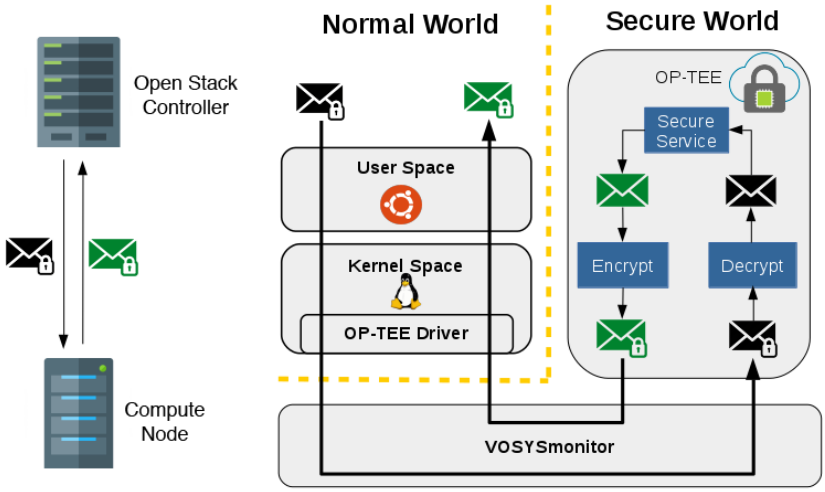

Figure 11-17: Trusted VIM with OpenStack (source [102])

The communication is encrypted as well as uses authentication mechanisms, where the OpenStack controller knows the public key of all the compute nodes, while all the compute nodes know the public key of the

OpenStack Controller. Both controller and compute nodes also have a private key that is stored locally and is used to encrypt the data. Furthermore, the public key of OpenStack Controller can be used to decrypt the data with the TEE of the controlled nodes and the presence of the trusted environment ensures that this data is secure against possible cyber-attacks. However, such a setup faces two main challenges that have to be considered towards its adoption in the 5GZORRO project:

1) The TEE is implemented only for the compute nodes and associated services and not the OpenStack controller, as this would require an extension to OpenStack platform. This leaves the controller exposed to cyber-attacks through existing vulnerabilities as well as open ports or exposed interfaces. As an outcome of an attack, the keys of the compute nodes can be stolen and then used to decrypt their data as well as transmit malicious data to cause abnormal operation on the compute nodes. Thus, complementary security mechanisms must be provisioned to shield the controller against cyber-attacks, such as endpoint security solutions or intrusion detection systems.
2) The adopted TEE is based on the ARM TrustZone technology to allow integrated hardware/software isolation. A simpler implementation scenario would be to avoid using dedicated hardware and thus rely only in a software approach. Even in this scenario though, any existing compute nodes within 5GZORRO's infrastructure have to be adapted entirely to support a software-based TEE, such as OP-TEE in the example of Figure 11-17.

11.4.6 Cloud native technologies for 5G

Cloud native technologies enable and facilitate the realization of cloud native applications, which are characterized by a microservice architecture, modularity, high availability, scalability and automation in deployment. Some of these enabling technologies are:

- Container technology
- Orchestration engine
- Service mesh

A **container technology** allows to package software in a lightweight entity, called container image, and to run it anywhere in an isolated environment from other processes. This technology can be considered as a lightweight virtualization, different from the traditional virtualization which uses virtual machines, because the containers use an OS-level virtualization without a hypervisor. In an OS-level virtualization, as depicted in Figure 11-18, the host OS kernel allows the coexistence of multiple isolated user-space instances, called containers, as if they were multiple isolated guest operating systems with their applications on top. In the other case, the hardware-level virtualization, the hypervisor, or the virtual machine manager (VMM) allows the virtualization of different guest OSs with their kernel.

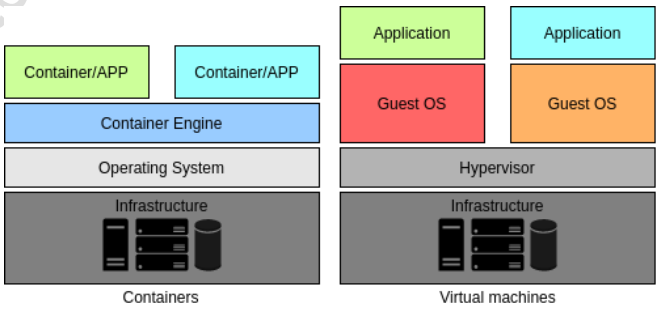

Figure 11-18: Comparison between containers and virtual machines

A container technology has a lot of advantages with respect to virtual machines; first of all, it reduces the size of images, indeed a virtual machine image may be 10GB while a container image can be an order of magnitude smaller, maybe 10 MB. This brings a better and faster mobility, since a container can be easily uploaded to a public repository and downloaded by the customer. Moreover, a container has less overhead, therefore a containerized application has a faster start up, allowing to scale up and down the resources on demand in near real time. The main disadvantage is that the OS-level virtualization is not as flexible as the

other virtualization, it can only host a guest host that is the same as the host OS. There are several container technologies such as Docker [103] and RKT [104].

Docker is the most popular container technology. The Docker architecture is a client-server architecture, described in Figure 11-19. The Docker daemon listens for Docker API requests and manages images, containers, networks, and volumes. The Docker client is a way to interact with Docker using Docker API. A Docker registry stores Docker images. The Docker registry can be public, such as Docker Hub, or private.

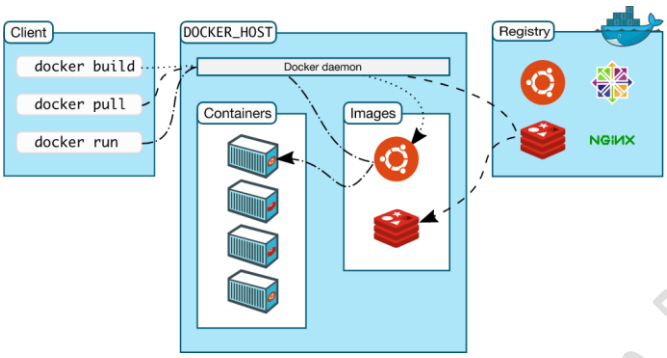

Figure 11-19: Docker architecture (source [103])

As the complexity of the application grows, all these containers should be managed through an orchestration engine, such as Docker Swarm or Kubernetes. An **orchestration engine** allows to facilitate the configuration and the automate the management of the lifecycle of the application, the scaling of the containers and the allocation of resources. Basically, it is composed of a set of worker nodes, that run containerized applications, and a control plane, which consists of a set of master (or manager) nodes responsible for maintaining entire state of the applications and making global decisions, such as scheduling a scale up/down of applications.

The most popular open source orchestration engine is Kubernetes [105]. It was first developed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF). In Kubernetes, the smallest deployable unit of computing is the Pod which is a group of one or more containers with shared storage and network resources. The architecture of Kubernetes, depicted in Figure 11-20, follows the general one described above.

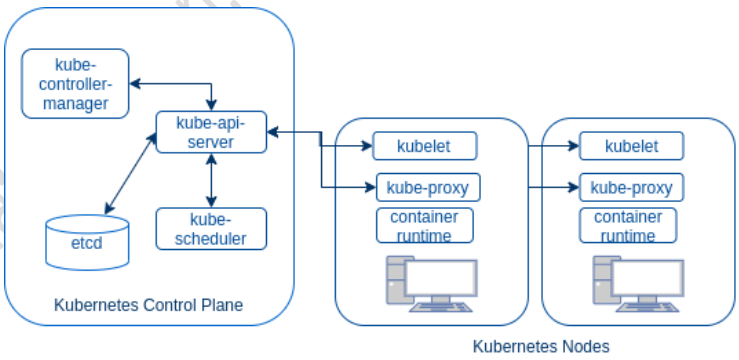

Figure 11-20: Kubernetes architecture

The Kubernetes Control Plane is composed of:

- *kube-api-server* is the component of the Kubernetes control plane that exposes the Kubernetes API. The API server is the front-end for the Kubernetes control plane.
- *etcd* is a lightweight, distributed key-value store used as Kubernetes' backing store for all cluster data.
- *kube-scheduler* is the component responsible for assigning workloads to nodes.
- *kube-controller-manager* is the component that runs controller processes.

The components of each Kubernetes node are:

- *kubelet* is a service responsible for relaying information to and from the control plane services, for example receiving commands and work or authenticating to the cluster.
- *kube-proxy* is a network proxy that maintains network rules on nodes. These network rules allow Pods to communicate with other Pods or with the outside of the cluster.
- *container runtime* is the software that is responsible for starting, and managing containers (Docker, rkt or others).

A **service mesh** is an infrastructure layer designed to simplify and secure communication among components (containers) in a cloud application. It separates the business logic of the components from the observability, network and security policies. Even though service meshes are mainly used in microservices applications, they can also be integrated with VMs and physical servers. The mesh is implemented adding a proxy instance, called sidecar, inside each service in the application. In this way, a service communicates with other services through the proxy sidecars, which encapsulate all the network and security rules. All together, these sidecars in the applications form the so-called "service mesh". Generally, a service mesh is logically split into Data plane and Control plane. The Data plane is the mesh made of all the sidecars, that is responsible for service discovery, health checking, routing, load balancing and authentication/authorization. The Control plane is responsible for managing and configuring the Data plane. There are two different types of service meshes depending on how sidecars are designed and the layer of the network stack offered: Istio [106] and Network Service Mesh (NSM) [107].

Istio is an open source service mesh framework that operates between L7 and L4. Actually, Istio is the control plane and Envoy is the default proxy of Istio. The Istio architecture is depicted in Figure 11-21. As previously mentioned, the data plane is made of services with a proxy as sidecar, the control plane is composed of:

- *Pilot* is the component responsible for managing and configuring the data plane, or Envoy proxies. It converts high level routing rules that control traffic behavior into Envoy-specific configurations and propagates them to the sidecars at runtime. It is also responsible for service discovery.
- *Mixer* enforces access control and usage policies across the service mesh, and collects telemetry data from the Envoy proxy and other services.
- *Citadel* is responsible for assigning certificates to each service.

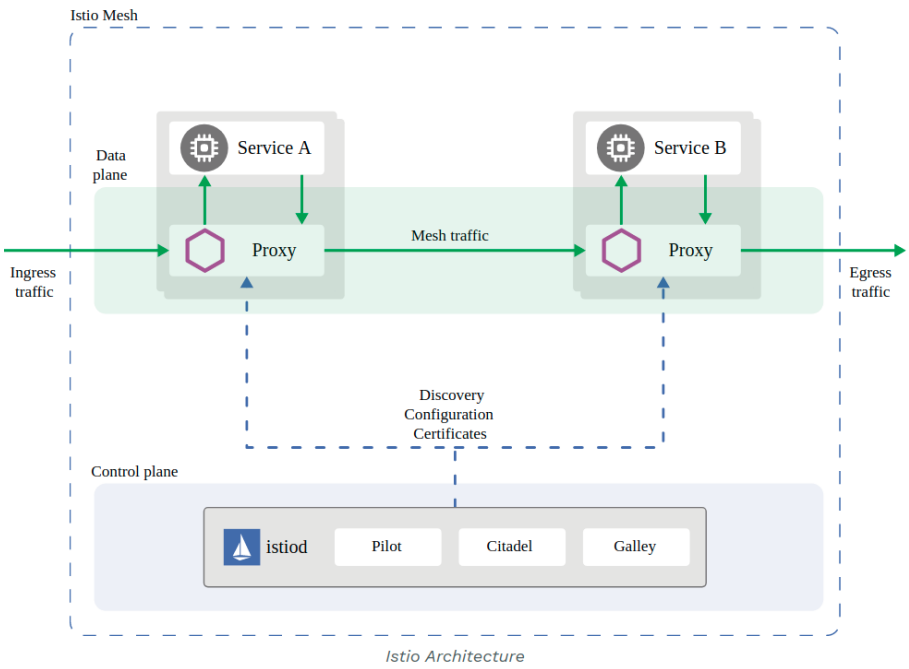

Figure 11-21: Istio architecture (source [106])

Network Service Mesh (NSM) is the other type of service mesh that provides L3/L2 connectivity. The architecture of NSM is depicted in Figure 11-22. The main components are:

- *Network Service Endpoint (NSE)* is the implementation of Network Services, which can be a container, pod or a virtual machine.
- *Network Service Client (NSC)* is a consumer of a Network Service.
- *Network Service Registry (NSR)* is the registry of NSM, it stores information of NSE.
- *Network Service Mesh Dataplane (NSMD)* is the data plane component providing end-to-end connections, mechanisms and forwarding elements to a network service.
- *Network Service Manager (NSMgr)* is the control plane of NSM, it is capable of handling service discovery, service routing and service connection management to create a L2/L3 connection. A NSMgr is deployed on each node and they communicate with each other to form a distributed control plane. A NSMgr is mainly responsible for two things:
    - It accepts the Network Service requests from the NSC and matches the request with appropriate NSE, then creates the connection between the NSC and NSE using the data plane.
    - It registers the NSE on its node to the NSR.

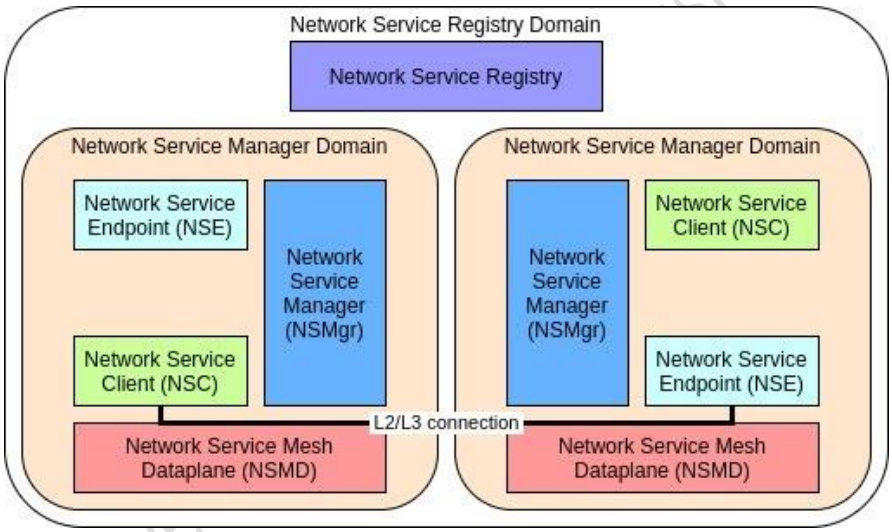

Figure 11-22: Network Service Mesh architecture

Following these cloud-native principles, the idea in 5GZORRO is to create an environment where Virtualized Network Functions (VNFs), which are the software implementation of network functions usually realized in hardware and provided as a virtual machine images, are split and developed as microservices, using containers, and so inheriting all the benefit of the cloud-native technologies such as scalability, resiliency and observability. Those Cloud-Native Network Functions (CNFs) should be easily deployed using a container orchestrator such as Kubernetes and they should coexist and communicate to functions in other domains using different levels of service mesh.

<END OF DOCUMENT>