



H2020 5Growth Project
Grant No. 856709

D3.5: Final version of software implementation for the platform

Abstract

The goal of this deliverable is to provide the documentation of the final release of software developed within the 5Growth project for the integration of the 5Growth platform with ICT-17 platforms and with the infrastructure deployed for the project pilots (available in the project GIT repository). This code has been developed based on the initial analysis and design reported in D3.2. Further design decisions were described in D3.3. And the final implementation overview and deviations from the first implementation, described in D3.3, are summarized in this document.

Document properties

Document number	D3.5
Document title	Final version of software implementation for the platform
Document responsible	Oleksii Kolodiazhnyi (Mirantis)
Document editor	Oleksii Kolodiazhnyi (Mirantis)
Editorial team	Andrea Boddi (TEI), Diego San Cristobal (ERC), Fernando Beltrán (ERC), Fabio Ubaldi (TEI), Giada Landi (NXW), Juan Brenes (NXW), Ramón Perez (TELCA), Matteo Pergolesi (TELCA), José Bonnet (ALB), Carlos Marques (ALB), Miguel Mesquita (ALB), Jorge Baranda (CTTC), Josep Mangues (CTTC), Engin Zeydan (CTTC), Paolo Dini (CTTC), Ricardo Martínez (CTTC), Luca Vettori (CTTC), Carlos Guimaraes (UC3M), Jaime García (UC3M), Carlos J. Bernardos (UC3M), Daniel Corujo (IT), João Fonseca (IT), João Alegria (IT), Vitor Cunha (IT), David Santos (IT), Jose Quevedo (IT), Mario Antunes (IT), Pedro Gonçalves (IT), Andres Garcia Saavedra (NEC), Xi Li (NEC), Simone Panicucci (COMAU)
Target dissemination level	PU
Status of the document	Final
Version	1.0
Delivery date	June 30, 2021
Actual delivery date	June 30, 2021

Production properties

Reviewers	Diego San Cristobal (ERC), Carlos Guimaraes (UC3M), Andres Garcia Saavedra (NEC), Diego Lopez (Telefonica), Carlos J. Bernardos (UC3M)
------------------	--

Disclaimer

This document has been produced in the context of the 5Growth Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement N° H2020-856709.

All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors' view.

Contents

List of Figures.....	5
List of Tables.....	6
List of Acronyms	7
Executive Summary.....	9
1. Introduction.....	10
1.1. Initial context.....	10
1.2. Structure of the document.....	10
2. 5Growth integration with 5G-EVE	11
2.1. Software artifacts	12
2.1.1. 5Gr-VS driver towards 5G-EVE	13
2.1.1.1. General description	13
2.1.1.2. Services and workflows	15
2.1.2. 5G-EVE IWL driver	18
2.1.2.1. IWL catalogue	18
2.1.2.2. IWL Lifecycle Manager.....	19
2.2. Blueprints for 5Growth pilot vertical services deployed in 5G EVE.....	19
2.2.1. Blueprints and descriptors for INNOVALIA pilot.....	19
2.2.1.1. INNOVALIA UC1: Connected Worker Remote Operation of Quality of Equipment...20	
2.2.1.2. INNOVALIA UC2: Connected Worker - Augmented ZDM Decision Support System (DSS).....	21
2.2.2. COMAU Pilot	21
3. 5Growth integration with 5G-VINNI.....	24
3.1. Software artifacts	25
3.1.1. 5Gr-VS modifications for interaction with 5G-VINNI.....	25
3.1.1.1. General description	25
3.1.1.2. Services and workflows	26
3.1.1.3. Implementation.....	28
3.1.2. 5Gr-VS driver towards 5G-VINNI.....	28
3.1.2.1. New features description	28
3.1.2.2. New services and workflows	28
3.1.2.3. Implementation.....	29

3.1.3. SONATA adaptor	30
3.1.3.1. New features description	30
3.1.3.2. New services and workflows	30
3.1.3.3. Implementation.....	32
3.2. Blueprints for 5Growth pilot vertical services deployed in 5G VINNI.....	34
3.2.1. Interdomain use-case.....	35
4. Conclusions	37
References.....	38

List of Figures

Figure 1: 5Growth integration with 5G-EVE	11
Figure 2: Example of VSB for 5GR-VS Catalogue	14
Figure 3: Workflow for the design of a VS deployed in the 5G-EVE platform	15
Figure 4: Workflow for the instantiation of vertical services in the 5G-EVE platform	17
Figure 5: 5G-EVE IWL Driver Components Location	18
Figure 6: Overview of the NSD and VNFD for INNOVALIA use-cases: (A) UC1; (B): UC2.....	21
Figure 7: Relationship between UC3 blueprint and descriptors with the Pilot infrastructure.....	22
Figure 8: 5Growth integration with 5G-VINNI.....	24
Figure 9: NST configuration rule onboarding workflow	26
Figure 10: 5Gr-VS enhanced instantiation workflow for Day-1 network slice configuration	27
Figure 11: 5Gr-VS VSMF Implementation updates.....	28
Figure 12: New Operations supported by the 5Gr-VS driver for SONATA.....	29
Figure 13: Flows for configuration OF A SONATA service inside 5GR-VS	29
Figure 14: 5Gr-VS Software architecture for Sonata integration.....	30
Figure 15: SONATA Adapter High-Level Architecture	30
Figure 16: High-Level Sequence Diagram for Configuration.....	31
Figure 17: VSB for 5Gr-VS Catalogue with SONATA NstIds.....	35

List of Tables

Table 1: List of software components for interaction between 5G EVE and 5Growth 13

Table 2: List of software components for interaction between 5g-VINNI and 5Growth..... 25

Table 3: NBI API..... 32

Table 4: Message Body Definition..... 32

Table 5: SONATA API supported by SBI..... 33

List of Acronyms

5Gr-RL – 5Growth Resource Layer
5Gr-SO – 5Growth Service Orchestrator
5Gr-VS – 5Growth Vertical Slicer
AGV – Autonomous Guided Vehicle
API - application programming interface
BBU – Baseband Unit
CSMF – Communication Service Management Function
CtxB – Context Blueprint
CN – Core Network
DSS – Decision Support System
ExpB – Experiment Blueprint
ExpD – Experiment Descriptor
HSS – Home Subscriber Server
IWL – Interworking Layer
MSO-LO – Multi-Site NSO to Local Orchestrator
NBI – North-Bound Interface
NF – Network Function
NFVO – Network Functions Virtualization Orchestrator
NSD – Network Service Descriptor
NST – Network Slice Template
PNF – Physical Network Function
QoS – Quality of Service
RAN – Radio Access Network
RAT – Radio Access Technology
RBAC – Role-Based Access Control
REST - Representational state transfer
RS – Remote Support
RU – Remote Unit
SBI – South-Bound Interface

TCB – Test Case Blueprint

TN – Transport Network

UPF – User Place Function

vEPC – virtual Evolved Packet Core

VS LCM – Vertical Service Lifecycle Manager

VSF – Vertical Service Blueprint

VSD – Vertical Service Descriptor

VSMF – Vertical Service Management Function

VNFD – VNF Descriptor

ZDM – Zero-Defect Manufacturing

Executive Summary

The 5Growth project aims to empower vertical industries such as Industry 4.0, Transportation, and Energy with Artificial Intelligence-driven automated 5G end-to-end solutions that will allow them to simultaneously achieve their respective key performance targets. To this end, the 5Growth project targets the validation of vertical use cases based on innovative technologies such as 5G infrastructure, virtualization, and dynamic service orchestration. A number of vertical pilots are planned to run on top of the 5Growth infrastructure to validate their use-cases, namely INNOVALIA, COMAU, EFACEC_S, and EFACEC_E pilots. The 5Growth software stack contributes to that purpose by providing functionalities to orchestrate vertical services. At the same time, an experiment validation framework is required as well, which is a feature provided by ICT-17 platforms. In order to provide more capabilities and features for the verticals, the 5Growth stack must work jointly with ICT-17 platforms to benefit from their experiment validation facilities.

The 5Growth contribution teams came out with the final software solutions, as an evolutionary outcome from previous deliverables. The initial plans regarding the implementation of the integration and software solutions, designed around satisfying vertical requirements and use-cases for the pilots, were developed and proposed in document D3.2 [1]. The final software implementation, described in this document, is based on the first software release covered in D3.3 [2]. The final version of the software release includes additional patches and fixes that were performed after verification.

1. Introduction

This deliverable D3.5 ("Final version of software implementation") provides the documentation of the final release of integration of 5Growth with ICT-17 platforms software components. The deliverable includes a high-level view of the software components, and a detailed description of the changes that were made since the first software version snapshot described in the D3.3 [2].

1.1. Initial context

This software is based on the analysis, architecture, and interface definitions introduced in D3.2 [1], and the first implementation described in D3.3 [2]. In these previous deliverables, some different alternatives were being considered and a summary of the final design decision was described. The current document will expose the updates made upon this final design.

1.2. Structure of the document

The main content of this document is presented into two sections:

- Section 2 describes a high-level summary of the improvements and updates performed for the software components implementing the integration between the 5Growth and 5G-EVE platforms.
- Section 3 describes a high-level summary of the improvements and updates performed for the software components implementing the integration between the 5Growth and 5G-VINNI platforms.

2. 5Growth integration with 5G-EVE

The integration of the 5Growth architecture presented in D2.3 [3] with the 5G EVE platform is based on the concept of delegating the provisioning of the vertical services to the 5G EVE system, exploiting the REST API offered by the 5G EVE Portal for the management of 5G service experiments. The developed solution, already presented in D3.3 [2], is based on option 2, as described in D3.2 Section 3.3. As mentioned in D3.3, this choice is motivated by the need to exploit the inter-site capabilities introduced by the 5G EVE platform in its latest release. In particular, the 5G EVE platform offers the possibility to instantiate, monitor, and evaluate the performance of vertical services that can be deployed across multiple sites, handling the logic of the inter-site service composition. Following this approach, we can rely on the 5G EVE features to deploy vertical services that span across 5Growth verticals' premises (e.g., the COMAU premises for COMAU use cases) and 5G EVE sites (e.g., the 5G EVE Italian site for COMAU use cases).

This integration approach is fully aligned with the default model defined in the 5G EVE framework to interact with external platforms from ICT-19 projects. From the perspective of the 5Growth architecture, the workflow can be summarized as follows. The Communication Service Management Function (CSMF) at 5Growth Vertical Slicer (5Gr-VS) interacts with the 5G EVE Portal through a programmable REST API to request the deployment and instantiation of the whole vertical service by the 5G EVE platform (i.e., vertical service request). When this request is received on the 5G EVE side, the 5G EVE Portal forwards the request to the 5G EVE Interworking Layer (IWL), which coordinates the creation of the network services across the different target sites, including the 5Growth facilities managed by the 5Growth Service Orchestrator (5Gr-SO). The IWL interacts with the 5Gr-SO placed in the vertical premises through a dedicated 5Growth Adapter, already introduced in D3.3, which takes care of the translation and adaption between 5G EVE and 5Growth interfaces and information models. This interaction is visually depicted in Figure 1.

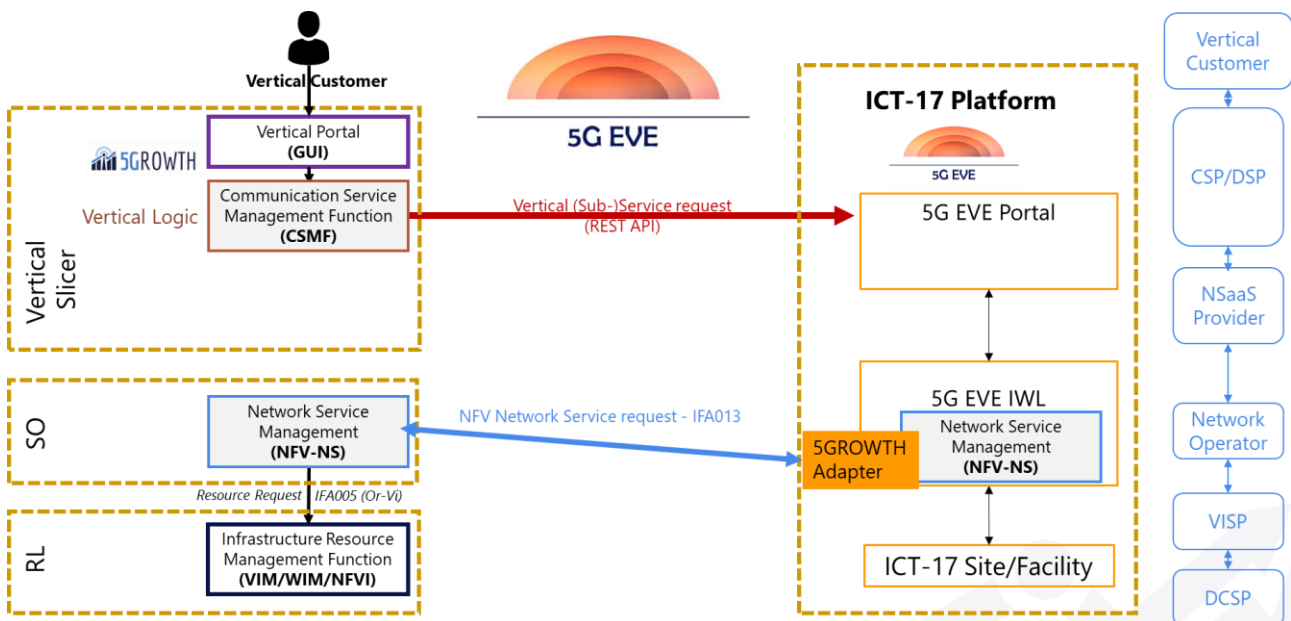


FIGURE 1: 5GROWTH INTEGRATION WITH 5G-EVE

The software components required to enable the interaction between 5Growth and 5G EVE are the following:

- 5Gr-VS driver towards 5G-EVE in the 5Growth platform (see D3.3, section 2.1);
- 5G-EVE IWL catalogue driver in the 5G-EVE platform (see D3.3, section 2.2.1);
- 5G-EVE IWL Lifecycle Manager driver in the 5G-EVE platform (see D3.3, section 2.2.2).

All these components have already been released in D3.3, and they have mostly remained unchanged in terms of functionalities and interfaces, except for the usual bug fixes and refinements performed during the testing and integration activities with the pilot real environments. The reader can refer to D3.3 for the documentation.

The only component that has been quite heavily modified is the 5Gr-VS driver, in order to fully exploit the most recent updates of the 5G EVE platform in the support of the 5G RAN configuration. This feature, introduced with the latest release of the 5G EVE Portal, allows specifying the service requirements in terms of mobile connectivity so that the 5G EVE platform can accordingly allocate and configure the RAN resources on the 5G EVE sites where this capability is supported. The integration of this function in the context of 5Growth pilots allows to better manage the QoS of the vertical services in the radio segment, allowing evaluating the service behaviour with different conditions of the mobile connectivity. The main software updates are related to the 5Gr-VS driver changes for adapting the client and the service requests to the enhanced REST API exposed by the 5G EVE Portal to support this feature. These updates are documented in section 2.1.1.

This section provides an overview of the software artifacts that have been implemented in the 5Growth project to enable its interaction with the 5G EVE platform (section 2.1) Moreover, it presents the blueprints and descriptors for the vertical services designed and developed for the Innovalia (section 2.2.1) and COMAU pilots (section 2.2.2), which use the 5G EVE Spanish and Italian sites, respectively, to deploy (part of) their services for early evaluation or to complement the infrastructure installed in their local premises.

2.1. Software artifacts

This section provides the list of the software components developed in the 5Growth project to enable the interaction between the 5Growth and 5G EVE platforms. It should be noted that some components, in particular the 5G EVE IWL drivers, have been developed in the 5Growth project and fully integrated with the 5G EVE software, thus providing a concrete contribution also in terms of 5G EVE platform enhancements. As mentioned in Section 2, some components have remained unchanged from the previous software release in D3.3 (except for minor bug fixes and pilot-specific customizations). In the following we focus on the software updates introduced since the last version; the reader can refer to D3.3 for the description of all the software modules.

The list of software components for the interaction with 5G EVE is collected in Table 1, reporting the link to their repository and the reference to the D3.3/D3.5 sections where they are documented.

TABLE 1: LIST OF SOFTWARE COMPONENTS FOR INTERACTION BETWEEN 5G EVE AND 5GROWTH

Name	Documentation	Software repository
5Gr-VS driver towards 5G-EVE	First release: D3.3, section 2.1 Final release: D3.5, section 2.1.1	https://github.com/5growth/5gr-vs
5G-EVE IWL catalogue driver	First release: D3.3, section 2.2.1 Final release: D3.5, section 2.1.2.1	https://github.com/nextworks-it/5g-catalogue Note: in "5Growth" branch
5G-EVE IWL Lifecycle Manager driver	First release: D3.3, section 2.2.2 Final release: D3.5, section 2.1.2.2	https://github.com/5growth/mso-lo
Code developed for COMAU integration	First release: D3.3, section 2.5 Final release: D3.5, section 2.2.2	https://5growth.eu/redmine/projects/5growth/repository/pilots/COMAU

2.1.1. 5Gr-VS driver towards 5G-EVE

2.1.1.1. General description

As described in D3.3 [2] the integration between the 5Gr-VS and the 5G EVE platform is based on the decomposition of the end-to-end Vertical Service (VS) in vertical sub-services, and the delegation of the provisioning and management of one or more sub-services to the 5G EVE platform (by means of a CSMF level interface). This allows a vertical to request a service using the North-bound Interface (NBI) of the 5Gr-VS, which will automatically map one or more sub-service to 5G EVE experiments and manage the lifecycle of the 5G EVE experiments based on the lifecycle of the end-to-end service. These automated vertical service customization, instantiation, and termination rely on a vertical design phase, where the Vertical Service Blueprints (VSBs) and Experiment Blueprints (ExpBs) are onboarded on the 5Gr-VS and 5G EVE Portal respectively.

In D3.3 [2], we defined the format of a composite VSB to be onboarded on the 5Gr-VS catalogue, providing an example where one of the service components was a vertical sub-service to be requested to the 5G EVE platform. The 5G EVE Platform API has been extended since then to support inter-domain experiments and to enable the configuration of RAN characteristics and requirements to be associated with the experiment, as described in [4]. This update extends the parameters to be specified during the 5G EVE experiment customization and instantiation. As described in D3.3, the experiment customization values are filled in using the configurable parameters of the VSB onboarded in 5Growth. In Figure 2, we show the changes of the 5Growth VSB example from D3.3, adding in bold the new configurable parameters used to identify the RAN requirements to be sent to the 5G EVE platform.

```
{
  "vsBlueprint": {
```

```

    "interSite": true,
    "version": "1.0",
    "name": "vs service name",
    "description": "end_to_end service",
    "parameters": [
      {
        "parameterId": "5GEVE_VSS.<experiment_param_id>",
        "parameterName": "experiment parameter name",
        "parameterType": "number",
        "parameterDescription": "description",
        "applicabilityField": "applicabilityField"
      }
    ],
    ...
  ],
  "configurableParameters": [
    "5GEVE_VSS.expd.kpi.<expb_kpiId>.lowerbound",
    "5GEVE_VSS.expd.kpi.<expb_kpiId>.upperbound",
    "5GEVE_VSS.exp.targetSites",
    "5GEVE_VSS.exp.experimentStopTime",
    "5GEVE_VSS.exp.useCase",
    "5GEVE_VSS.tc.tcBlueprintIds",
    "5GEVE_VSS.tc.<tcBlueprintId>.params.<param_id>",
    "5GEVE_VSS.endpoint.<endpoint_id>.params.<param_type>"
  ],
  "atomicComponents": [
    {
      "type": "SERVICE",
      "componentId": "5GEVE_VSS",
      "compatibleSite": "5GEVE",
      "associatedVsbId": "<Experiment blueprint id>"
    }
  ]
}

```

FIGURE 2: EXAMPLE OF VSB FOR 5GR-VS CATALOGUE

The *endpoint_id* of the new configurable parameter element corresponds to the endpoint associated with the RAN in the 5G EVE service blueprint. The *param_type* can be *RAT* (Radio Access Technology), *LATENCY*, *UplinkThroughput*, or *DownlinkThroughput*. The *RAT* parameter can be used for any type of service (e.g. 4G, 5G non-standalone and standalone, etc), *LATENCY* is only used for URLLC services,

while *UplinkThroughput/DownlinkThroughput* is used for EMBB services. We refer to [4] for further details regarding the 5G EVE Portal API. The corresponding values will then be provided at instantiation time when requesting the provisioning of the vertical service from the 5Gr-VS GUI.

2.1.1.2. Services and workflows

Figure 3 and Figure 4 describe the vertical service design workflows, as defined in D3.3, with the corresponding changes to integrate the 5G EVE Platform updates. Figure 3 describes the workflow for the vertical service design phase, highlighting in red the two interactions affected by the 5G EVE Platform updates. As described in D3.3, the vertical first onboards the service and experiment blueprints through the 5G EVE Portal. In this case, the endpoints of the VSB enable the definition of the service slice type (EMBB and URLLC), and an identifier of the geographical area for the endpoints of the service associated with the RAN. In the 5Growth side, the vertical onboards the end-to-end VSB defining accordingly the RAN customization parameters to be configured during the service instantiation.

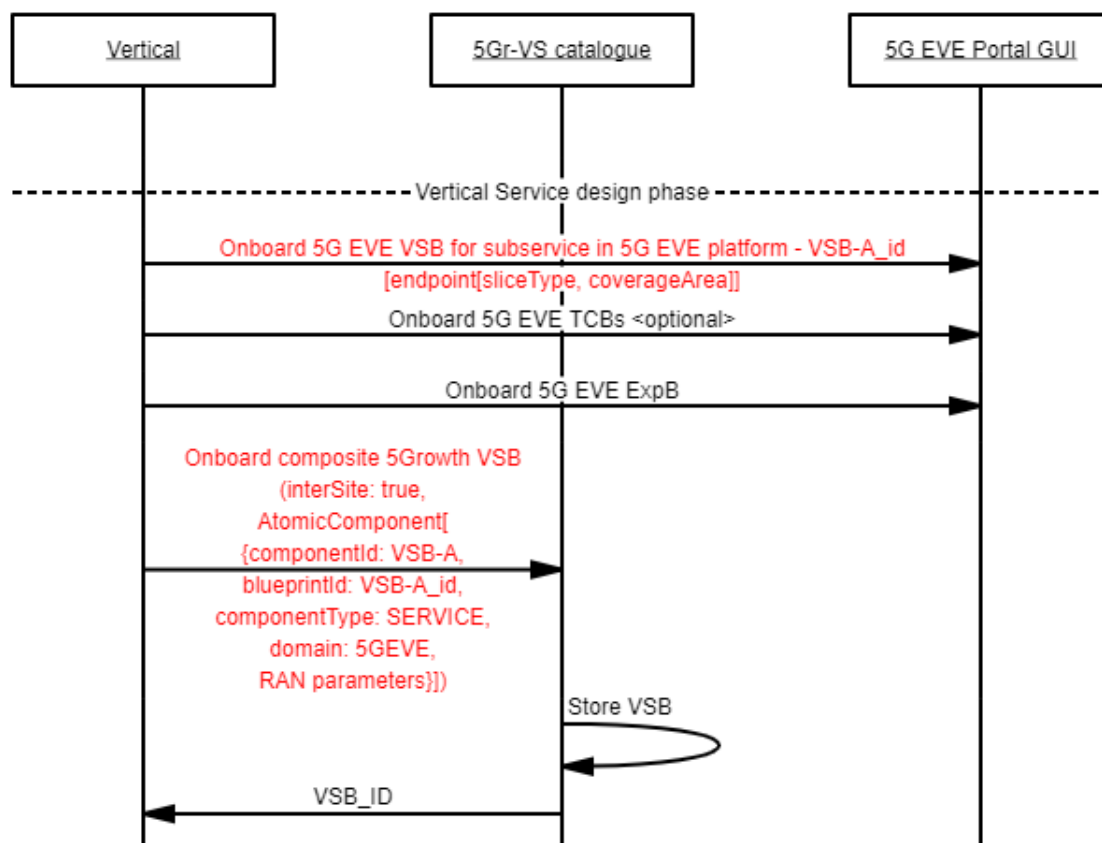


FIGURE 3: WORKFLOW FOR THE DESIGN OF A VS DEPLOYED IN THE 5G-EVE PLATFORM

The instantiation workflow, detailed in Figure 4, starts when a vertical service instantiation request arrives at the North-bound Interface of the 5Gr-VS. As mentioned previously, the instantiation request contains the encoded values for the VSB configuration parameters, which includes the RAN-

related parameters in this final version. The 5Gr-VS Vertical Service Lifecycle Manager (VS LCM) identifies the vertical sub-service to be delegated to the 5G EVE Platform and relays the request to the 5G EVE Vertical Service Management Function (VSMF) Driver, by means of the `VsmfInteractionHandler`. This latter driver uses the encoded parameters to create and onboard the 5G EVE Experiment Descriptor (ExpD) corresponding to the 5Growth vertical service to be provisioned, and it generates the subsequent experiment lifecycle management requests (i.e., to schedule, deploy and execute the experiment on the 5G EVE platform). Once the experiment is in running execution state, the vertical sub-service is considered to be instantiated and a notification is issued to notify the VS LCM about the vertical sub-service status change. We refer to D3.3 for further details about the different phases and translations performed by the 5G EVE VSMF Driver. The RAN parameters introduced in this final release are encoded in the 5G EVE Vertical Service Descriptor (5G EVE VSD), which is part of the ExpD. In particular, the RAN parameters are encoded using a map, with the endpoint id as the key and a `SliceProfile` object containing the RAN parameters as the value. In Figure 4, we highlight in red the updated interactions for this final release.

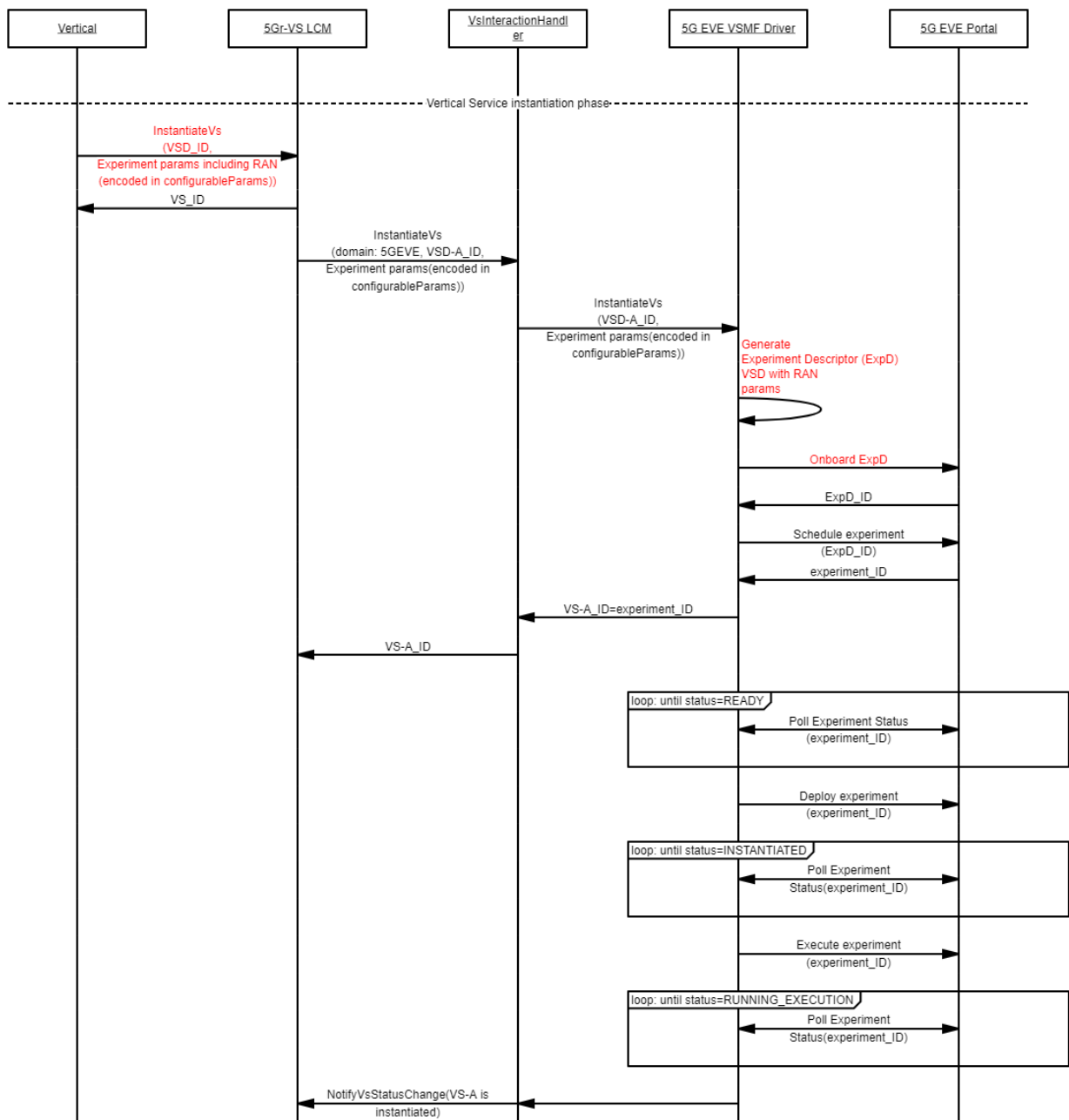


FIGURE 4: WORKFLOW FOR THE INSTANTIATION OF VERTICAL SERVICES IN THE 5G-EVE PLATFORM

In terms of implementation, the final release follows the functional and software architecture of the initial release defined in D3.3. The code of the EveVsmfDriver class and the REST client to interact with the 5G EVE Portal catalogue (5GEVE_PORTAL_CATALOGUE_REST_CLIENT) were updated to add the enhanced translation mechanisms and support the new 5G EVE API mentioned in the previous section.

Moreover, the logging mechanisms at these software components have been enhanced to support the 5Gr-VS monitoring functionalities related to the automated computation of the service instantiation time. These updates have been introduced to enable the seamless interaction with the global

5Growth logging system used for the elaboration of the 5Growth platform KPIs, in support of monitoring the various per-domain contributions to the end-to-end provisioning time for the vertical services in the pilots using 5G EVE as target 5G infrastructure.

2.1.2. 5G-EVE IWL driver

As presented in D3.3 [2], The 5Growth Service Orchestrator (5Gr-SO) driver in the 5G-EVE Interworking Layer (IWL) is located at the South Bound Interface (SBI) of the 5G-EVE Adaptation Layer in the IWL.

This component is in charge of abstracting the on-boarding and lifecycle management capabilities from multiple NFV Orchestrators (NFVOs). The adaptation layer driver exposes to 5G-EVE upper layers an ETSI SOL005 [5] REST interface, and the 5Gr-SO exposes an ETSI IFA013 [6] REST interface so that this module performs the corresponding translation between both interfaces.

The integration point between both domains is the 5G-EVE Multi-Site NSO to Local Orchestrator (MSO-LO) REST API application, which will be the component in charge of translating ETSI IFA013 and ETSI SOL005 data models.

Figure 5, extracted from D3.3 [2], depicts where the 5G-EVE IWL driver component is located in the architecture.

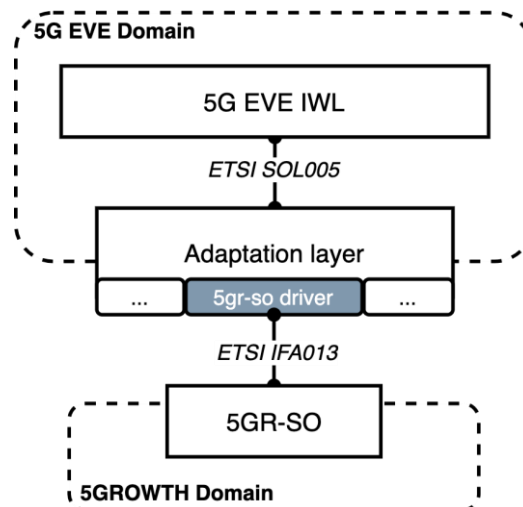


FIGURE 5: 5G-EVE IWL DRIVER COMPONENTS LOCATION

The driver implementation has been organized into two modules: the first includes catalogue related operations, while the second includes lifecycle management operations. The main characteristics of these two modules and the latest advancements in development are presented in the next subsections.

2.1.2.1. IWL catalogue

This component has been improved with bug fixes, and there are no significant changes from the software reported in D3.3 [2].

The 5G-EVE IWL catalogue is responsible for the management of Network Service Descriptors (NSD) and VNF packages as well as their synchronization with the catalogues of the specific NFVOs deployed in each 5G-EVE site facility. This module exposes a North-Bound Interface compliant with the ETSI NFV SOL 005 specification and adopts the ETSI NFV SOL 001 specification for the NSD and VNF format.

Please, refer to that deliverable to find all the information related to the IWL Catalogue.

2.1.2.2. IWL Lifecycle Manager

This component has been improved with the bug fixes, and there are no significant changes from the software reported in D3.3 [2].

This module of the 5G-EVE IWL driver includes all the required management operations to control the lifecycle of a Network Service. The features supported are developed following the ETSI IFA013 interface definition, in order to cover all the requirements exposed by the 5Gr-SO that indeed implements it. The module also extends the standard definition in order to support the retrieval of the lifecycle operation list. A detailed list of the supported lifecycle operations is provided in D3.3 [2].

Please refer to that deliverable to find all the information related to the IWL Lifecycle Manager.

2.2. Blueprints for 5Growth pilot vertical services deployed in 5G EVE

In addition to the different blueprints and descriptors provided to the 5Growth platform, other descriptors were required to be defined and on-boarded on the 5G EVE platform. These allow an experiment to be designed in 5G EVE platform, and are defined as follows [7]:

- **Vertical Service Blueprints (VSBs)**: high-level information model for a template of a network service.
- **Context Blueprints (CtxBs)**: template for network elements used to define an operational context for the network service.
- **Test Case Blueprints (TCBs)**: customizable templates that describe the actions to be performed during the execution of the experiment.
- **Experiment Blueprint (ExpB)**: template for an experiment to be executed on the 5G EVE platform.
- **Experiment Descriptor (ExpD)**: defines a specific experiment for the 5G EVE platform.

Details for the INNOVALIA and COMAU pilots are provided in the following subsections.

2.2.1. Blueprints and descriptors for INNOVALIA pilot

For the deployment of the two targeted Use Cases in the INNOVALIA Pilot, referred to as INNOVALIA UC1 and UC2, as already explained across several deliverables and summarized in D3.3 Section 1.1, there are different blueprints and descriptors that must be onboarded on 5G EVE and 5Growth platforms.

The blueprints and descriptors required by the 5G EVE side are:

- A Vertical Service Blueprint (VSB).
- An Experiment Blueprint (ExpB).
- A Test case Blueprint (TcB).
- A Network Service Descriptor (NSD).

On the 5Growth side, the following are required:

- A VSB.
- A Vertical Service Descriptor (VSD), which is a template providing values to the parameters specified in the VSB.
- A Virtual Network Function Descriptor (VNFD).
- A NSD.

Find all the files listed above at the GitHub path within the Innovalia_Vertical_Pilot/5Growth_descriptors folder at [8].

The following provides details about the NSDs and VNFDs needed for UC1 and UC2.

2.2.1.1. INNOVALIA UC1: Connected Worker Remote Operation of Quality of Equipment

Figure 6(a) depicts the deployed VNF at the 5Growth Nfvi-Pop required for the INNOVALIA UC1. Such VNF integrates the functions of both Data Assembler and Robot Link. Both functions are interconnected via the UPF PNF to support the communication with the RAN element. The descriptors used for deploying the Data Assembler and Robot Link are two: i) the VNFD describing the VNF (and the associated requirements) to be instantiated at the 5Growth Nfvi-Pop; ii) and the NSD determining among other information the peer connectivity of the deployed VNF. The description of the information elements for both VNFD and NSD are encoded via JSON data being available at [8]. In the following, the most relevant content for the UC1 VNFD and NSD are:

VNFD

- VNFs: Data Assembler and Robot Link
- Compute requirements: CPU (4), Virtual Memory (16 GB), Storage (100 GB)
- Supported operation: Instantiation and Termination of the VNF

NSD

- A virtual link connecting the VNF with the UPF through the so-called n6_cmm_vl
- The deployment flavor ("df_Inno_CMM") supports only a single instance of the VNF

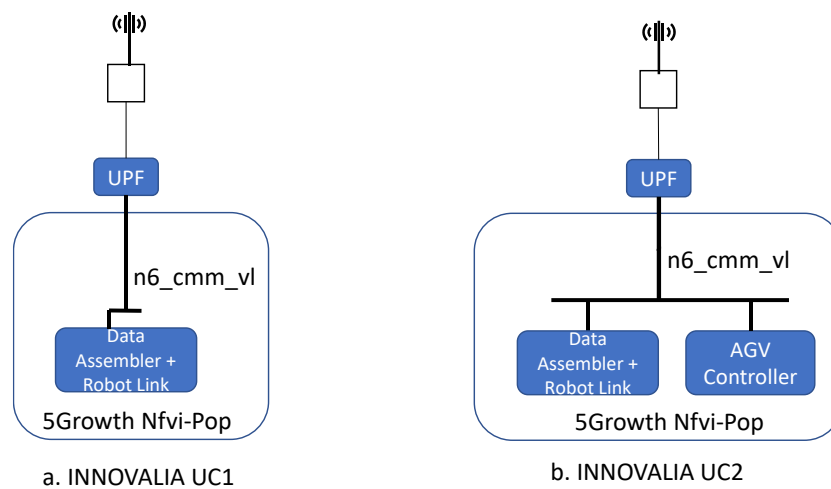


FIGURE 6: OVERVIEW OF THE NSD AND VNFD FOR INNOVALIA USE-CASES: (A) UC1; (B): UC2

2.2.1.2. INNOVALIA UC2: Connected Worker - Augmented ZDM Decision Support System (DSS)

Figure 6(b) illustrates the deployed VNFs at the 5Growth Nfvi-Pop addressing the targeted UC2. In addition to the VNF hosting both the Data Assembler and the Robot Link functions, there is a dedicated VNF for the AGV controller. Thus, the NSD and VNFD used in the UC1 are extended to instantiate the AGV controller VNF. The detailed definition (using JSON format) of the information elements contained in both descriptors for this UC2 is available in [8]. The following describes the key aspects of such descriptors:

VNFD

- VNF for the Data Assembler and Robot Link
 - o Compute requirements: CPU (4), Virtual Memory (16 GB), Storage (100 GB)
 - o Supported operation: Instantiation and Termination of the VNF
- VNF for the AGV Controller
 - o Compute requirements: CPU (4), Virtual Memory (8 GB), Storage (45 GB)
 - o Supported operation: Instantiation and Termination of the VNF

NSD

- A virtual link connecting the VNF (hosting the Data Assembler and Robot Link) with the UPF through the so-called n6_cmm_vl
- A virtual link connecting the VNF (AGV Controller) with the UPF through the same n6_cmm_vl
- The deployment flavour ("df_Inno_UC2") supports only a single instance of the VNF

2.2.2. COMAU Pilot

As described in document D3.4 [9], Use Case 3 (UC3) uses the integration of the 5Growth stack with the 5G EVE platform. A service, created for UC3 is called “Digital Tutorial and Remote Support” and provides high-definition videos and live streaming connections between “in-field” maintenance staff (i.e. staff located in the factory where the maintenance is needed) and skilled technicians located in a remote technical office. The scope of this service contains two areas:

- **Remote Support** - when machinery in a factory is affected by a fault, the “in-field” maintenance staff can contact a skilled technician in a remote office. Using the live and video streaming, the skilled technician has the “full picture” of the fault and can support the “in-field” staff to fix the fault, reducing the “Mean Time To Repair” (MTTR);
- **“Digital Tutorial”** - live and video streaming can be used for training purposes of new maintenance staff giving access to remote “step-by-step” digital tutorials.

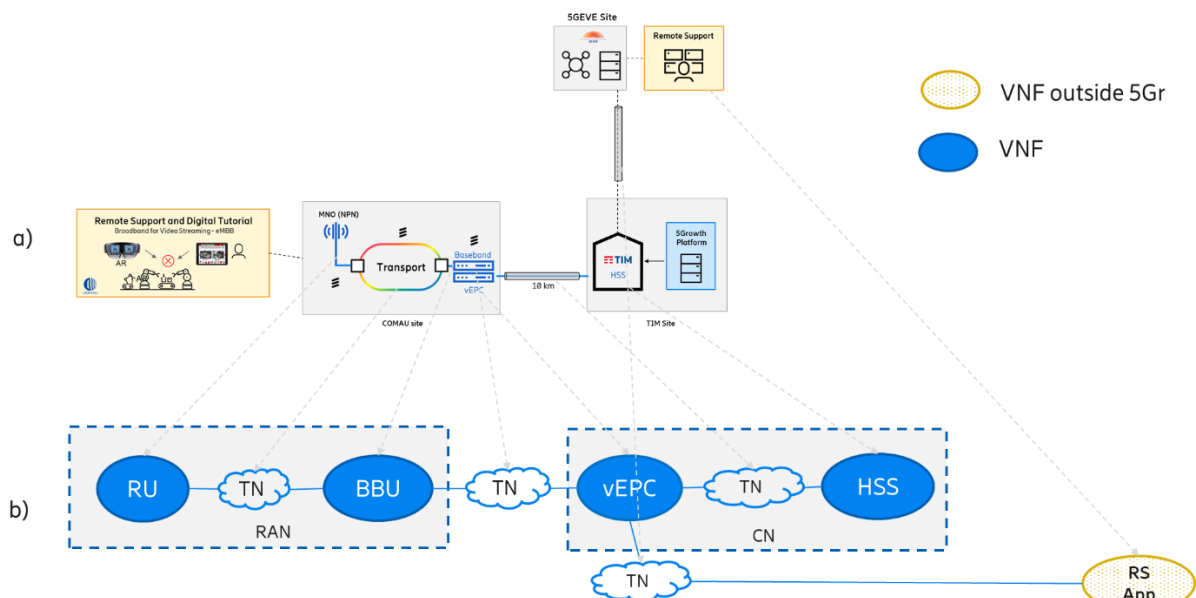


FIGURE 7: RELATIONSHIP BETWEEN UC3 BLUEPRINT AND DESCRIPTORS WITH THE PILOT INFRASTRUCTURE

Figure 7 shows the UC3 developed in the COMAU pilot highlighting the relationship between a) the Pilot infrastructure and b) the Service Descriptors. Specifically, the 5G EVE site is hosted in TIM premises located in Borgaro and is connected to the 5Growth site where the platform is hosted in the TIM premises located in Reiss Romoli. Conversely, the robotic, radio, and transport equipment are hosted on the COMAU site. UC3 service is formed by network functions that provide the connectivity between the robotic area and the remote skilled technicians, and the application used by the vertical to provide remote support. The network functions are divided into RAN functions, Core Network (CN) functions, and Transport Network (TN) functions while the vertical application used for remote support is called “RS App”. “RS App” is instantiated in the 5G EVE site (so it is handled by the 5GEVE platform), while RAN, CF, and TN functions are instantiated in 5Growth sites (so they are handled by the 5Growth platform). In such a scenario the interaction between 5Growth and 5GEVE is needed to connect the network function with the vertical applications.

Always referring to Figure 7, the service descriptor of UC3 is formed by the following files:

- UC3 Vertical Service Blueprint (VSB). It is a JSON file that gives a high-level description of the parameters and components of the service. This file describes the service from the customer (vertical) point of view.
- UC3 Network Slice Template (NST). It is an optional JSON file that gives the radio slice parameters as defined in [10].
- UC3 Network Service Descriptor (NSD). It is a JSON file that provides the UC3 service descriptor according to the ETSI model [11]. This file describes the service from the provider (operator) point of view. UC3 NSD is formed by two composite NSD files one used by the 5Growth stack to deploy the network functions and the other used by the 5GEVE stack to deploy the "RS App".
- Radio Unit (RU) Virtual Network Function (VNF). It is a JSON file describing the RU Network Function according to the ETSI model [12]. It represents the functionalities of the antenna (like the geographic area covered by the antenna).
- Base Band Unit (BBU) Virtual Network Function (VNF). It is a JSON file describing the BBU Network Function according to the ETSI model [12]. It represents the functionalities of the BBU equipment (like the digitalization of the radio signal).
- Virtual Evolved Packet Core (vEPC) Virtual Network Function (VNF). It is a JSON file describing the vEPC Network Function according to the ETSI model [12]. It represents the functionalities of the vEPC equipment (like the tracking area and the mobility management).
- Home Subscriber Server (HSS) Virtual Network Function (VNF). It is a JSON file describing the HSS Network Function according to the ETSI model [12]. It represents the functionalities of the HSS equipment (it is a database containing detailed information on User Equipment or UE)
- "RS App" Virtual Network Function (VNF). It is a JSON file describing the Vertical application used for remote support service according to the ETSI model [ref ETSI IFA011].

For details, JSON files used for the UC3 service can be found in [8].

3. 5Growth integration with 5G-VINNI

For the integration of 5Growth with 5G-VINNI, the CSMF at 5Gr-VS interacts with the Network Slice Management Function (NSMF) at 5G-VINNI, as described in D3.3 section 1.1.2. In particular, the integration between 5Growth and 5G-VINNI is done with the SONATA platform (a result of the 5GTANGO project), which is available at the 5G-VINNI Aveiro site. This integration comprises the adaptation of the 5Gr-VS requests to the SONATA Slice Manager (which has been designed and implemented before the 3GPP technical specification 28.531 [13]).

The flow between 5Gr-VS and 5G-VINNI's Network Function Virtualization Orchestrator (NFVO) follows the lines shown in Figure 2.

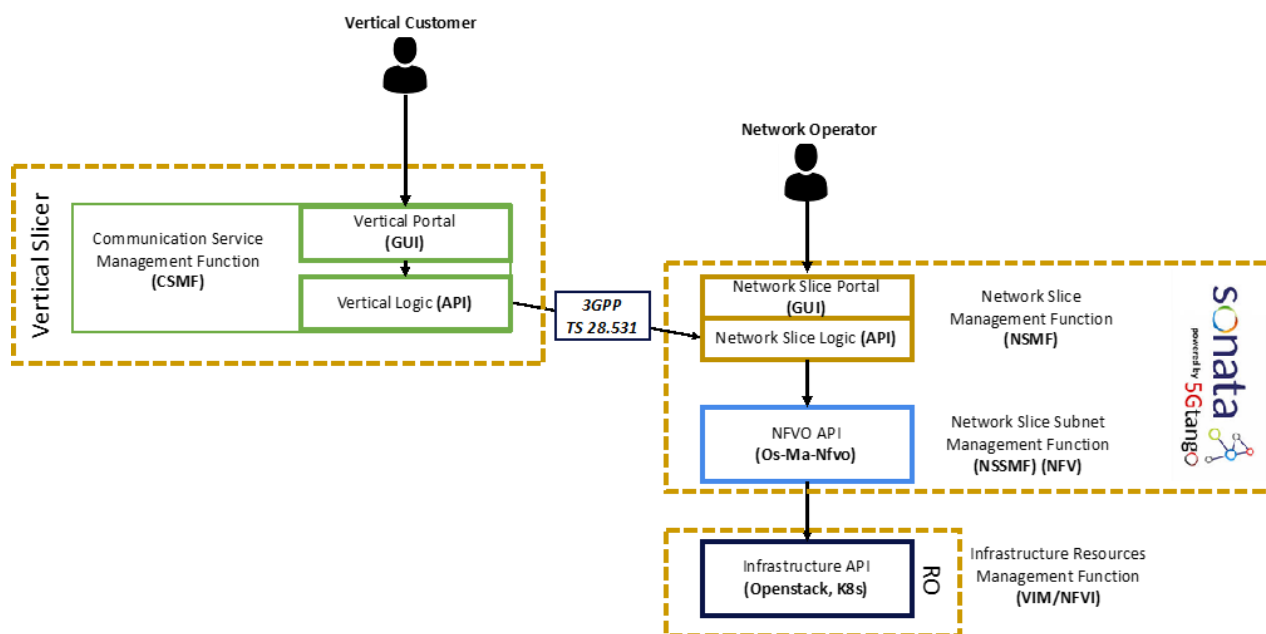


FIGURE 8: 5GROWTH INTEGRATION WITH 5G-VINNI

The software components required to enable the interaction between 5Growth and 5G-VINNI are the following:

- 5Gr-VS driver towards 5G-VINNI, in 5Growth platform, see Section 3.1.2;
- SONATA adaptor, in the 5G-VINNI platform, see Section 3.1.3.

All these components have been already released in D3.3, in sections 2.3 and 2.4, and they have been updated to support new functionalities, specifically to support Day-1 and Day-2 configurations.

This section provides an overview of the software artifacts that have been implemented, since D3.3, in the 5Growth project to enable its interaction with the 5G-VINNI platform (section 3.1). Moreover, it presents the blueprints and descriptors for the vertical services designed and developed for the Aveiro pilots (section 3.2), which are using the 5G VINNI Portugal site.

3.1. Software artifacts

This section describes the software provided in this delivery. It includes the code developed to integrate 5Growth with ICT-17 platforms. This section provides the list of the software components developed in the 5Growth project to enable the interaction between the 5Growth and 5G-VINNI platforms. As mentioned in section 3, the components have been described in the previous software release in D3.3. In the following, we focus on the software updates introduced since the last version (the reader can refer to D3.3 for the description of all the software modules).

The list of software components for the interaction with 5G VINNI is collected in Table 2: List of software components for interaction between 5g-VINNI and 5, reporting the link to their repository and the reference to the sections where they are documented.

TABLE 2: LIST OF SOFTWARE COMPONENTS FOR INTERACTION BETWEEN 5G-VINNI AND 5GROWTH

Name	Section	Repository
5Gr-VS driver towards 5G-VINNI	3.1.2	https://5growth.eu/redmine/projects/5growth/repository/5gr-vs
SONATA adaptor	3.1.3	https://github.com/5growth/sonata-drivers

3.1.1. 5Gr-VS modifications for interaction with 5G-VINNI

3.1.1.1. General description

As described in D2.3 [3], one of the key innovations introduced in the 5Gr-VS is the support for network slice level multi-domain deployments. This innovation enhanced the 5Gr-VS vertical service to network slice decomposition mechanisms enabling transparent decomposition of VSS into multidomain network slices, and automated translation of vertical service lifecycle actions into network slice lifecycle requests across multiple domains. Moreover, to transparently support different network slice management platforms, a driver-based approach was introduced to adapt the network slice lifecycle management requests to the particular API offered by the platform. This innovation is the key enabler for the 5G-VINNI integration, as described in D3.3 and further detailed in Section 3.1.2

This kind of scenario, where a vertical service is provisioned using slices across multiple domains, usually requires performing some configuration on the VNFs and PNFs for the end-to-end service to work. An example of this is the configuration of the IP addresses assigned to the endpoints of a network slice provisioned in a different domain. For this reason, the 5Gr-VS was enhanced to support automated Day-1 network slice configuration functionalities. In particular, the 5Gr-VS was extended with a catalogue of Network Slice Template (NST) configuration rules, which allow establishing the parameters and the message format to be used to configure slices based on certain NST. Furthermore, the vertical service lifecycle management logic was extended to use these configuration rules to automatically trigger the configuration requests.

3.1.1.2. Services and workflows

The NST configuration rules are onboarded during the vertical service design phase by an admin user, using the updated VSB onboarding workflow depicted in Figure 9.

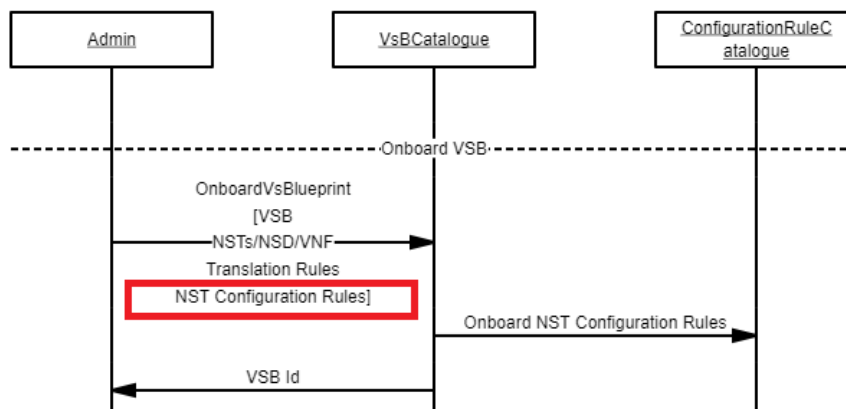


FIGURE 9: NST CONFIGURATION RULE ONBOARDING WORKFLOW

In Figure 10, we show the vertical service instantiation workflow, highlighting in red the new logic and interactions of this new feature. As shown in the figure, during the vertical service instantiation phase the vertical service lifecycle management (VS LCM) uses the NST configuration rules to automatically determine the required parameters, compute the values, and generate a configuration request when all the network slices have been instantiated. This configuration request is issued to the NSMF Domain Driver, which implements the NSMF Provider Domain-specific logic to request the configuration. It is important to highlight that since the main scenario for this feature is the automated configuration of the network slice endpoint addresses, an extra step was added to allow the 5Gr-VS to retrieve this information by issuing a network slice instance query.

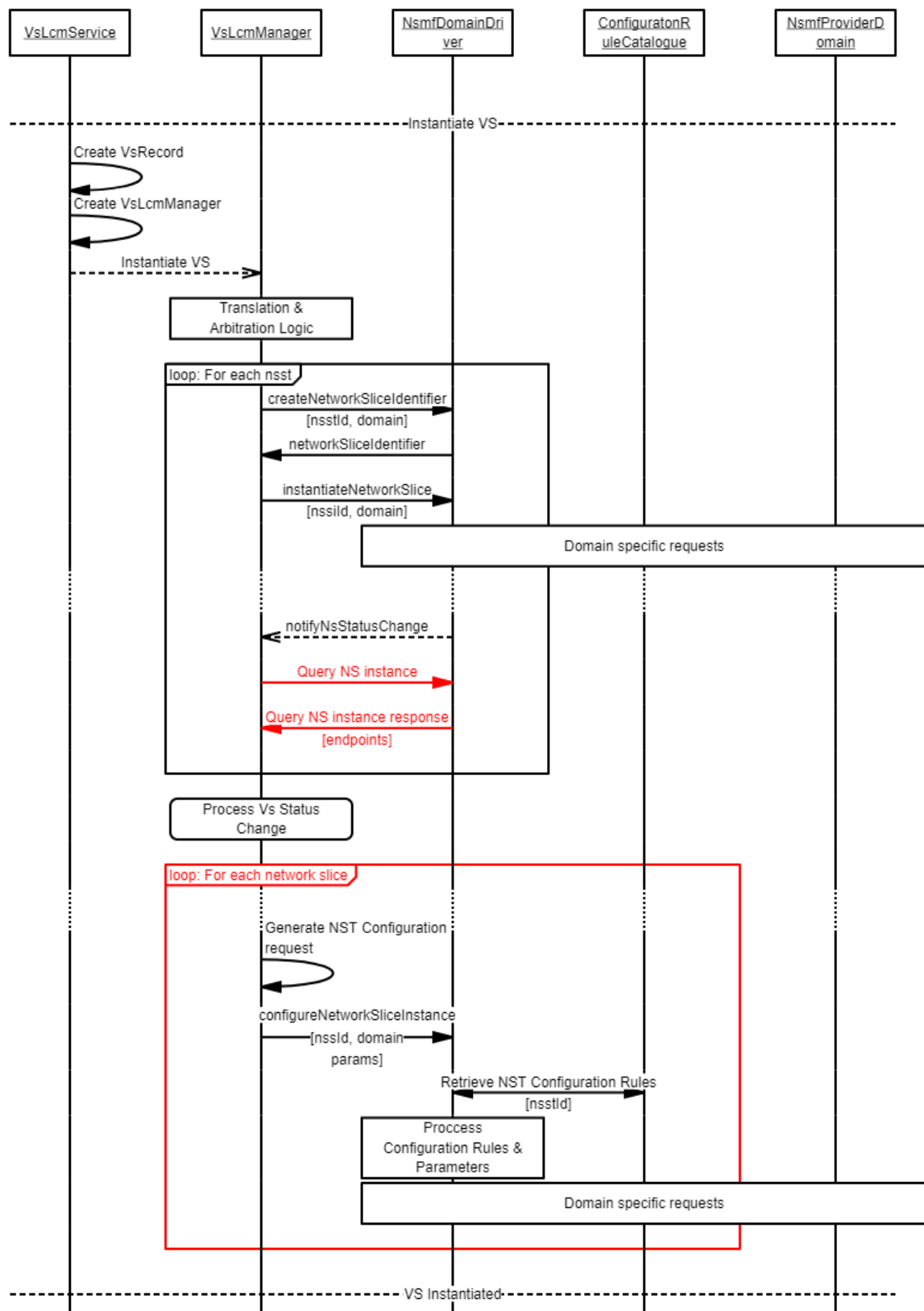


FIGURE 10: 5GR-VS ENHANCED INSTANTIATION WORKFLOW FOR DAY-1 NETWORK SLICE CONFIGURATION

3.1.1.3. Implementation

In Figure 11, we depict the functional blocks of the 5Gr-VS, highlighting the components that were updated or introduced for the support of automated Day-1 NST configuration mechanisms. We refer to D3.2 for further details about the different functional blocks. As depicted in the picture the NST configuration rule catalogue was introduced to store configuration rules, while the Vertical Service Catalogue (VS Cat REST) Controller was updated to support the workflow described in Figure 9. The VsLcmService and the Nsmf Interaction Handler were updated to support the workflow vertical service instantiation workflow detailed in Figure 10.

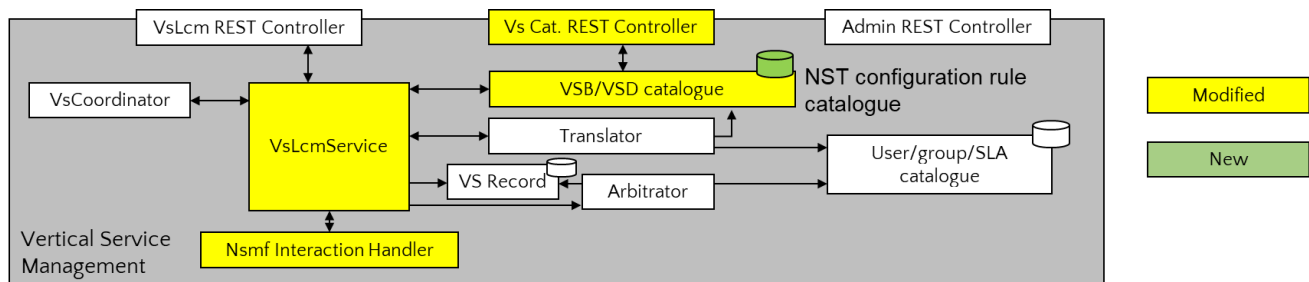


FIGURE 11: 5GR-VS VSMF IMPLEMENTATION UPDATES

In particular, the *NsmfLcmProviderInterface* implemented by the Nsmf Interaction Handler (and by the NSMF Domain Drivers) was extended to implement the method enabling the network slice configuration.

3.1.2. 5Gr-VS driver towards 5G-VINNI

This component was described in D3.3 and further detailed in Section 2.3. In the next subsection, we will refer to the new features and implementations for this component.

3.1.2.1. New features description

To support the new inter-domain innovation, as described in D2.3, this component was enhanced to support Day-1 and Day-2 configurations. For this purpose, we improve the process of Query Slice Instantiations to retry information of the Sonata Slice/Service/Function and add a new process to Configure Sonata Slice Instantiations after they are instantiated.

3.1.2.2. New services and workflows

The next figure shows the new operation supported by this component.

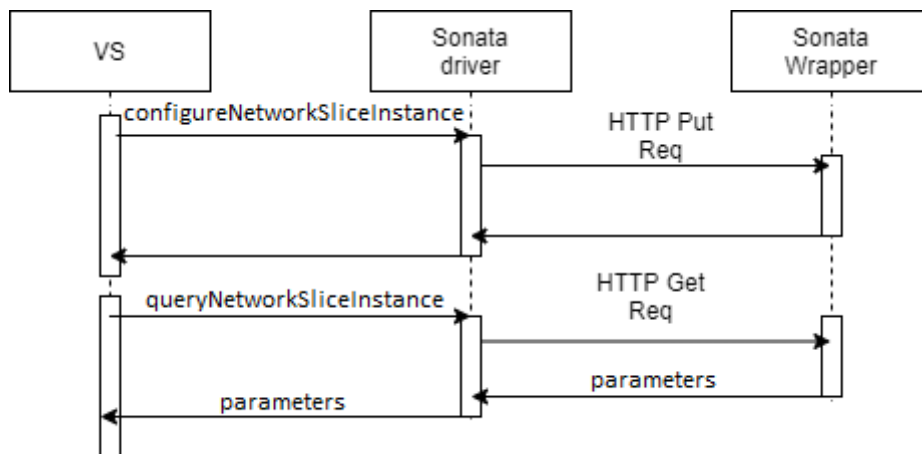


FIGURE 12: NEW OPERATIONS SUPPORTED BY THE 5GR-VS DRIVER FOR SONATA

The new operations implemented/ on this driver are:

- `configureNetworkSliceIdentifier` – operation to configure a network service slice on SONATA;
- `queryNetworkSliceInstance` – operation to obtain selected data about the running slice network service on SONATA (in this operation now is possible to retrieve specific info from the Slice/Service/Function).

As we can see in the following figure, we have depicted a vertical service configuration phase, the id obtained in the initiation process is used to identify the service to configure. The configuration is achieved when received the reply from the external system.

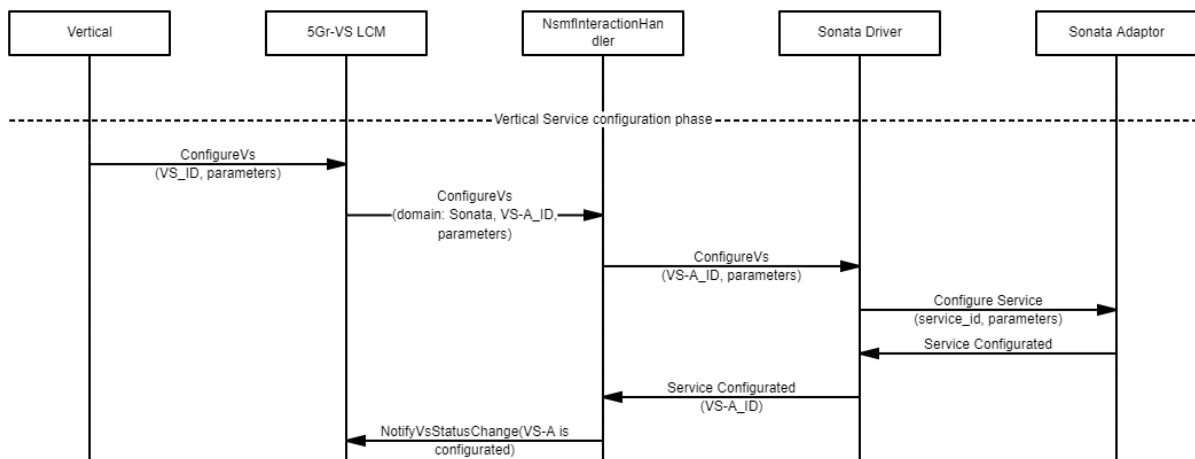


FIGURE 13: FLOWS FOR CONFIGURATION OF A SONATA SERVICE INSIDE 5GR-VS

Regarding the changes to the query only affect the content of reply in the instantiation phase, like was detailed in the previous section Figure 10.

3.1.2.3. Implementation

The implementation of the SONATA driver relies on the structure built inside the 5Gr-VS, depicted in Figure 14, already presented in D3.3, and further detailed in Section 2.3

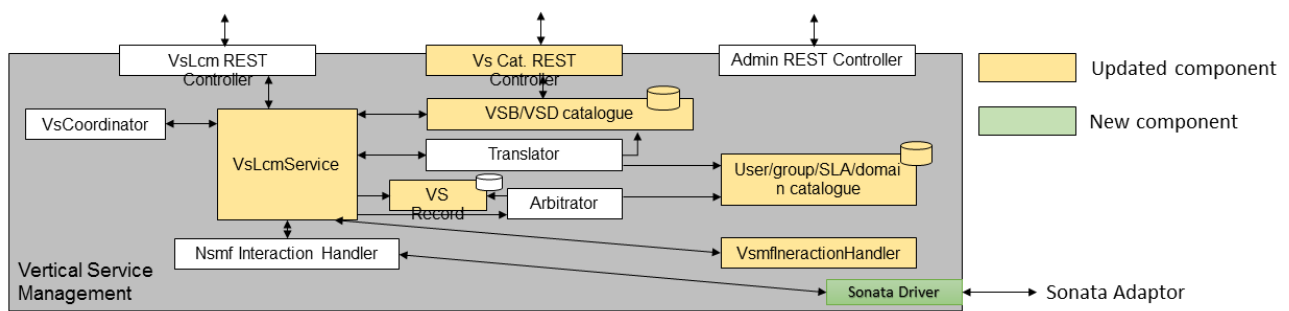


FIGURE 14: 5GR-VS SOFTWARE ARCHITECTURE FOR SONATA INTEGRATION

For these new features, the implementation consists of forwarding information relevant to the Sonata Adaptor.

3.1.3. SONATA adaptor

This component was described in D3.3 Section 2.4. In the next subsection, we will refer to the new features and implementations for this component.

3.1.3.1. New features description

To support the new inter-domain innovation, as described in D2.3, this component was enhanced to support Day-1 and Day-2 configurations.

For this purpose, we improve the message of Query Slice Instantiations to retry information of the Slice/Service/Function and add a new message to Configure Slice Instantiations after they are instantiated.

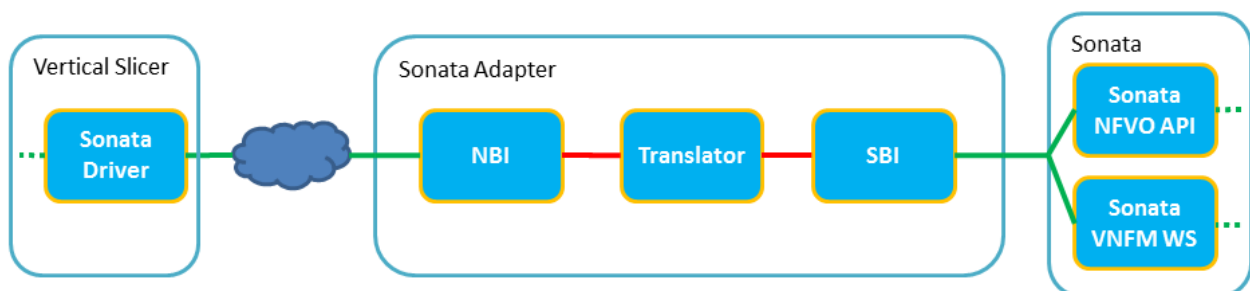


FIGURE 15: SONATA ADAPTER HIGH-LEVEL ARCHITECTURE

To fulfil the new messages, the Architecture described in Section 2.4.1 of D3.3 was enhanced with a new connection between the SBI (South Bound API) and the Sonata VNFM (Virtual Network Function Manager) by web socket communication.

3.1.3.2. New services and workflows

Figure 16 shows the flow diagram of the configuration process.

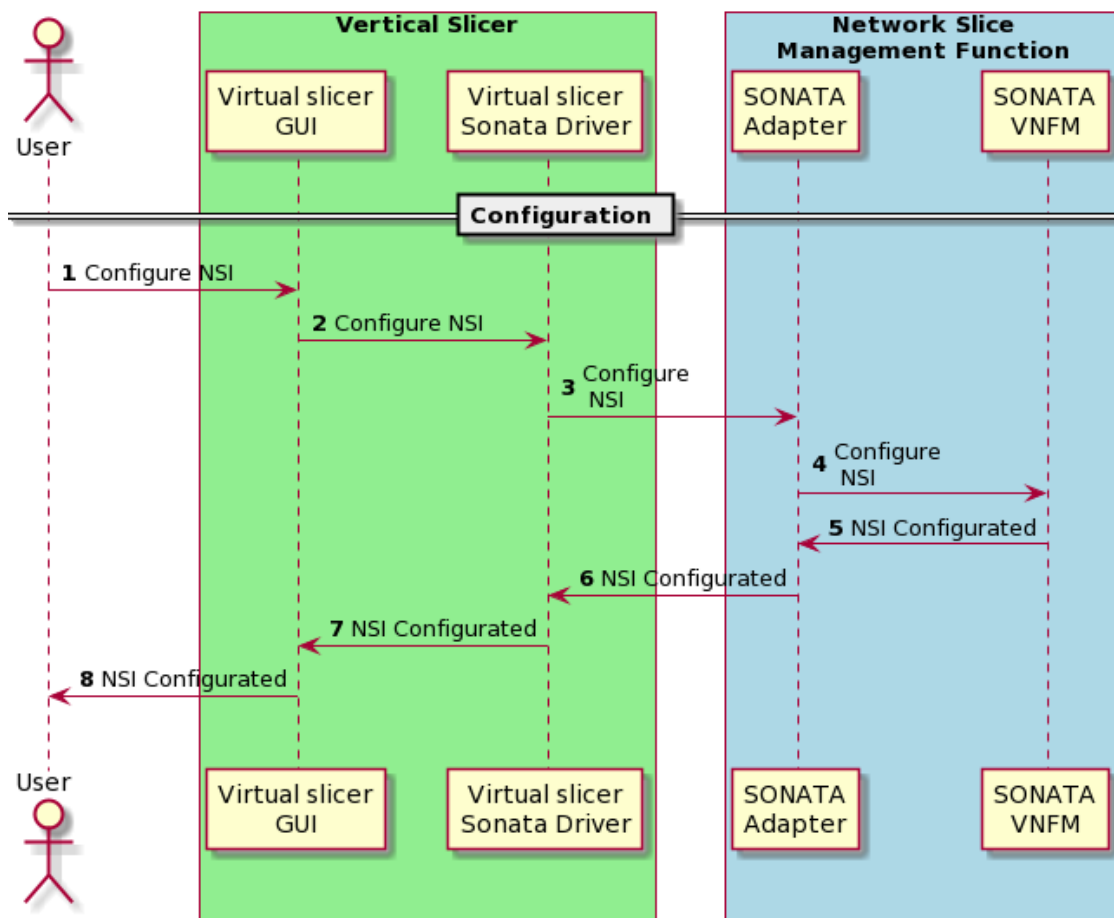


FIGURE 16: HIGH-LEVEL SEQUENCE DIAGRAM FOR CONFIGURATION

The sequence for the configuration process can be detailed as follows:

1. The Vertical User goes to the Vertical Slicer GUI and gives the instruction to configure a specific Network Slice Instantiation (NSI).
2. The 5Gr-VS knows that this NSI is for the SONATA Platform and forwards the instruction to the VS SONATA Driver.
3. The VS SONATA Driver sends the message to configure the Network Slice providing the Nsild and the parameters to the SONATA Adapter.
4. The SONATA Adaptor translates the message and sends that to SONATA VNFM to configure the slice.
5. The SONATA VNFM replies that the NSI was configured.
6. The SONATA VNFM replies that the NSI was configured.
7. The VS SONATA Driver, when receives the status for the finalized configuration, replies to VS with the information needed.
8. The 5Gr-VS GUI sends the information to Vertical User that the configuration was finalized.

3.1.3.3. Implementation

The API of the NBI (North-Bound Interface) has a new HTTP Method for configuring a Network Slice and an updated one to request the information of a Network Slice Instantiation, as summarized in Table 3.

TABLE 3: NBI API

Action	HTTP Method	Endpoint	Body	Output Status	Output Body
Configure a Network Slice	PUT	http://<sonata-ip>:<sonata-adapter-port>/api/v1/ns/<nsild>/action/configure	<ConfigureNsiRequest>	202	Null
Request the information of a Network Slice Instantiation	GET	http://<sonata-ip>:<sonata-adapter-port>/api/v1/ns/<nsild>		200	NetworkSliceInstance

The body content that goes in the messages from the previous table is described in Table 4

TABLE 4: MESSAGE BODY DEFINITION

ConfigureNsiRequest	NetworkSliceInstance
<pre>{ "nsild": "string", "parameters": { "param1": "string", "param2": "string", "param3": "string" } }</pre>	<pre>{ "name": "string", "description": "string", "nsild": "string", "nstld": "string", "nsdld": "string", }</pre>

	<pre> "nsdVersion": "string", "dflId": "string", "instantiationLevelId": "string", "nfVNsId": "string", "soManaged": "bool", "networkSliceSubnetInstances": "list<string>", "tenantId": "string", "status": "string", "errorMessage": "string", "nfVNsUrl": "string", "parameters": { "param1": "string", "param2": "string", "param3": "string" } } </pre>
--	---

The Actions supported by SBI and available in the SONATA WS are described in Table 5.

TABLE 5: SONATA API SUPPORTED BY SBI

Action	Endpoint	Body	Output Body
Configure a Network Slice	ws://<sonata-ip>:<sonata-ws-port>/ssm	<pre> { "name": "<name>", "id": "<uuid>", "action": "config", "parameters": { "param1": "string", "param2": "string", "param3": "string" } } </pre>	<pre> { "name": "<name>", "id": "<uuid>", "action": "config", "message": "Configuration OK" } </pre>

Request the information of a Network Slice Instantiation	ws://<sonata-ip>:<sonata-ws-port>/ssm	{ "name": "<name>", "id": "<uuid>", "action": "get_config" }	{ "name": "<name>", "id": "<uuid>", "action": "get_config", "parameters": { "param1": "string", "param2": "string", "param3": "string" } }
--	---------------------------------------	--	---

3.2. Blueprints for 5Growth pilot vertical services deployed in 5G VINNI

This section describes the use case and the blueprints used in the 5Growth integration with 5G-VINNI.

A generic VS Blueprint can be seen in Figure 17. This kind of descriptor provides information related to high-level definitions needed by the 5Gr-VS to manage the target service.

```
{
  "vsBlueprint": {
    "version": "1",
    "name": "vinni",
    "description": "vinni"
  },
  "nsts": [
    {
      "nstId": "vinni_slice",
      "nstName": "vinni_slice",
      "nstVersion": "1",
      "nstProvider": "ALB",
      "nsdId": "vinni_nsd",
      "nsdVersion": "1",
      "nsstIds": [
        "536fa477-119e-4fea-add5-05afc5b53e33"
      ]
    }
  ]
}
```

```
    },  
    {  
      "nstId": "536fa477-119e-4fea-add5-05afc5b53e33",  
      "nstName": "536fa477-119e-4fea-add5-05afc5b53e33",  
      "nstVersion": "1",  
      "nstProvider": "ALB",  
      "nsstIds": []  
    }  
  ]  
}
```

FIGURE 17: VSB FOR 5GR-VS CATALOGUE WITH SONATA NSTIDS

3.2.1. Interdomain use-case

Concerning the scenario already deployed and presented in deliverables D2.1 [14], D2.3 [3], D3.3 [2] and the testing deployments done, several blueprints and descriptors were created and onboarded into the 5G-VINNI and the 5Growth platforms to successfully deploy the use cases for Efacec Energia and Efacec Transportes.

The descriptors and blueprints needed in the 5G-VINNI are:

- A Virtual Network Function Descriptor (VNFD)
- A Network Service Descriptor (NSD)
- A Network Slice Template (NST)

The descriptors and blueprints needed in the 5Growth are:

- A Vertical Service Blueprint (VSB)
- A Network Slice Template (NST)
- A Network Slice Subnet Template (NSST)

All referenced entities must be aligned in order to successfully deploy the services needed by the defined use cases. On the 5G-VINNI side, the hierarchy of the NST composed by an NSD, which incorporates the VNFD, must be well configured. Furthermore, each layer of descriptors must act as a robust base for higher ones iteratively (VNFD, NSD, NST). Finally, the 5Growth VSB is the entity responsible for interconnecting and configuring all services. For that reason, several aspects in it are relevant to highlight, such as the quality-of-service parameters and their ranges, which are in line with the KPIs of the use cases. By leveraging those variables, the 5Growth platform creates a self-managed E2E Network Slice composed of one Network Slice Subnet, which in this scenario contains a Network Slice from another administrative domain, the 5G-VINNI.

The presented scenario already guarantees interdomain capabilities in the control plane by interconnecting different administrative domains. One corresponds to the vertical and contains services such as the 5Growth-VS and the 5Growth Web Portal, and the other corresponds to an operator domain, handled by the 5G-VINNI project which hosts all the vertical services.

The scenario is being extended to support interdomain capabilities in both the control plane and the services data plane. This second part will be achieved by making use of another NSMF driver and corresponding NSST at the VS level. The VS will again be responsible for instantiating this new Network Sub Slice over the desired domain. The descriptors and blueprints developed to enable this capability are available in [8].

4. Conclusions

This second code release of the platform software provides several corrections and enhancements of the software required for the integration of 5Growth and the ICT-17 platforms 5G-EVE and 5G-VINNI / SONATA. These were motivated mainly during the integration tests and connection with the real ICT-17 platforms and evolutions in the interfaces and/or internal procedures of the ICT-17 platforms themselves.

Although the implementation focuses on the support of the most common service lifecycle management operations, where creation, instantiation, status, and termination operations are fully integrated with functional workflows, the integration of on-boarding and scaling operations did not achieve the initially planned integration support. The reasons were mostly due to fact that interfaces to handle such operations were unavailable by the ICT-17 platforms. As an example, 5G EVE opted, as a design choice, to not enable ExpB onboarding via the 5G EVE Portal API, since it is considered as a management action that is not associated with the lifecycle management of the service. However, onboarding and synchronization of NSDs and VNF packages between the 5G EVE platform and 5Gr-SO are supported through the IWL catalogue driver, which has been developed in the context of 5Growth.

Upon the successful connection between 5Growth and ICT-17 platform sites, pilots were provided with the required environment to trigger multi-/inter-domain deployments, making use of 5G EVE and 5G VINNI. As such, the required descriptors and blueprints were defined for the different use cases, allowing them to be triggered by a user playing the role of the Vertical.

Overall, the implemented software modules, which are part of the 5Growth platform release, contribute to a more complete, mature, and robust codebase that can be leveraged to support many other vertical industries and their use cases that go beyond the ones targeted in the 5Growth project.

References

- [1] 5Growth, "D3.2, "Specification of ICT-17 in-house deployment"," [Online]. Available: https://5growth.eu/wp-content/uploads/2020/04/D3.2-Specification_of_ICT17_in-house_deployment.pdf.
- [2] 5Growth, "D3.3, "First version of software implementation for the platform"," [Online]. Available: https://5growth.eu/wp-content/uploads/2019/06/D3.3-First_version_of_software_implementation.pdf.
- [3] 5Growth, "D2.3, "Final Design and Evaluation of the innovations of the 5G End-to-End Service Platform"," [Online]. Available: https://5growth.eu/redmine/projects/5growth/repository/5growth/raw/execution/WP2/deliverables/D2.3/D2.3-Final_Design_and_Evaluation_of_5G_End-to-End_Service_Platform.pdf.
- [4] 5G-EVE, "D4.6, "Final version of the experimentation portal"," [Online]. Available: NO LINK AVAILABLE YET.
- [5] ETSI, "ETSI GS NFV-SOL 005, "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point"," [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/02.07.01_60/gs_NFV-SOL005v020701p.pdf.
- [6] ETSI, "ETSI GS NFV-IFA 013, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Os-Ma-nfvo reference point - Interface and Information Model Specification"," [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/013/03.04.01_60/gs_NFV-IFA013v030401p.pdf.
- [7] 5G-EVE, "D4.3, "Models for vertical descriptor adaptation"," [Online]. Available: <https://www.5g-eve.eu/wp-content/uploads/2020/05/5geve-deliverabled4.3-final.pdf>.
- [8] 5Growth, "Github: "Pilots repository"," [Online]. Available: <https://github.com/5growth/5gr-pilots>.
- [9] 5Growth, "D3.4, "Plan for pilot deployments"," [Online]. Available: https://5growth.eu/?smd_process_download=1&download_id=1076.
- [10] GSMA, "NG.116, "Generic Network Slice Template"," [Online]. Available: <https://www.gsma.com/newsroom/wp-content/uploads//NG.116-v5.0-7.pdf>.
- [11] ETSI, "ETSI GS NFV-IFA 014, "Network Functions Virtualisation (NFV); Management and Orchestration; Network Service Templates Specification"," [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/014/02.01.01_60/gs_NFV-IFA014v020101p.pdf.

- [12] ETSI, "ETSI GS NFV-IFA 011, "Network Functions Virtualisation (NFV); Management and Orchestration; VNF Packaging Specification",," [Online]. Available: https://www.etsi.org/deliver/etsi_gs/nfv-ifa/001_099/011/02.01.01_60/gs_nfv-ifa011v020101p.pdf.
- [13] 3GPP, "3GPP TS 28.531, "5G; Management and orchestration; Provisioning",," [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128500_128599/128531/15.00.00_60/ts_128531v150000p.pdf.
- [14] 5Growth, "D2.1, "Initial Design of 5G End-to-End Service Platform",," [Online]. Available: https://5growth.eu/wp-content/uploads/2019/11/D2.1-Initial_Design_of_5G_End-to-End_Service_Platform.pdf.
- [15] ETSI, "ETSI GS NFV-SOL 001, "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; NFV descriptors based on TOSCA specification",," [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/001/03.03.01_60/gs_NFV-SOL001v030301p.pdf.
- [16] 5Growth, "D2.2, "Initial implementation of 5G End-to-End Service Platform",," [Online]. Available: https://5growth.eu/wp-content/uploads/2020/05/D2.2-Initial_implementation_of_5G_End-to-End_Service_Platform.pdf.
- [17] 3GPP, "3GPP TS 28.530, "5G; Management and orchestration; Concepts, use cases and requirements",," [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128500_128599/128530/15.00.00_60/ts_128530v150000p.pdf.
- [18] ETSI, "ETSI GS NFV-IFA 005, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification",," [Online]. Available: https://www.etsi.org/deliver/etsi_gs/nfv-ifa/001_099/005/03.01.01_60/gs_nfv-ifa005v030101p.pdf.
- [19] 5Growth, "Redmine: "Pilots repository",," [Online]. Available: <https://5growth.eu/redmine/projects/5growth/repository/pilots>.