5G-SOLUTIONS for European Citizens

D1.2B Cross-domain Service Orchestration and Management Challenges Analysis (v2.0)

<table>
<thead>
<tr>
<th>Grant Agreement No</th>
<th>856691</th>
<th>Acronym</th>
<th>5G Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Title</td>
<td>5G Solutions for European Citizens</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Date</td>
<td>01/06/2019</td>
<td>Duration</td>
<td>42 months</td>
</tr>
<tr>
<td>Project URL</td>
<td><a href="https://www.5gsolutionsproject.eu/">https://www.5gsolutionsproject.eu/</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deliverable</td>
<td>D1.2B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work Package</td>
<td>WP1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contractual due date</td>
<td>30/11/2021 (M30)</td>
<td>Actual submission date</td>
<td>01/12/2021</td>
</tr>
<tr>
<td>Nature</td>
<td>Report</td>
<td>Dissemination Level</td>
<td>Public</td>
</tr>
<tr>
<td>Lead Beneficiary</td>
<td>LMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responsible Author</td>
<td>Anne-Marie Bosneag (LMI)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contributions from</td>
<td>Anne-Marie Bosneag (LMI), Shubham Jain B (LMI), Udi Margolin (NOKIA), Min Xie (TNOR), Sofiane Zemouri (IBM), Ramy Mohamed(IBM), Silvia Canale (Ares2T), Panagiotis Kontopoulos (NOVA), Ran Rahav (LiveU), Min Xie (TNOR), Andrea di Giglio (TIM), Dimitris Giannopoulos(UoP)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No 856691.
**Revision history (including peer reviewing & quality control)**

<table>
<thead>
<tr>
<th>Version</th>
<th>Issue Date</th>
<th>% Complete&lt;sup&gt;1&lt;/sup&gt;</th>
<th>Changes</th>
<th>Contributor(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>25/08/2021</td>
<td>5%</td>
<td>Initial Deliverable Structure and ToC</td>
<td>Anne Marie Bosneag (LMI), Shubham Jain B (LMI)</td>
</tr>
<tr>
<td>V0.1</td>
<td>10/09/2021</td>
<td>5%</td>
<td>Quality Review</td>
<td>Christos Skoufis (EBOS)</td>
</tr>
<tr>
<td>V0.2</td>
<td>18/10/2021</td>
<td>20%</td>
<td>Input towards Sections 3 &amp; 4</td>
<td>Andrea Di Giglio (TIM), Silvia Canale (Ares2T), Min Xie (TNOR), Panagiotis Papaioannou (UoP), Ramy Mohamed (IBM)</td>
</tr>
<tr>
<td>V0.2</td>
<td>20/10/2021</td>
<td>30%</td>
<td>Integrated input to Sections 3 &amp; 4</td>
<td>Anne-Marie Bosneag (LMI), Shubham Jain B (LMI)</td>
</tr>
<tr>
<td>V0.3</td>
<td>27/10/2021</td>
<td>40%</td>
<td>Contributions towards Section 5</td>
<td>Panagiotis Kontopoulos (NOVA), Silvia Canale (Ares2T), Ran Rahav (LiveU), Ramy Mohamed (IBM), Min Xie (TNOR)</td>
</tr>
<tr>
<td>V0.3</td>
<td>1/11/2021</td>
<td>50%</td>
<td>Integrated discussions output with different LLs &amp; CDSO</td>
<td>Anne-Marie Bosneag (LMI)</td>
</tr>
<tr>
<td>V0.4</td>
<td>10/11/2021</td>
<td>70%</td>
<td>Contributions towards all sections</td>
<td>Anne-Marie Bosneag (LMI), Shubham Jain B (LMI)</td>
</tr>
<tr>
<td>V0.5</td>
<td>20/11/2021</td>
<td>90%</td>
<td>Restructuring of content</td>
<td>Anne-Marie Bosneag (LMI)</td>
</tr>
<tr>
<td>V0.5</td>
<td>24/11/2021</td>
<td>100%</td>
<td>Formatting, references, abbreviations, generating tables for content / figures, etc.</td>
<td>Anne-Marie Bosneag (LMI)</td>
</tr>
<tr>
<td>V0.5</td>
<td>28/11/2021</td>
<td>100%</td>
<td>Technical review</td>
<td>Udi Margolin (Nokia), Ramy Mohamed (IBM), Albert Torres (IRIS)</td>
</tr>
<tr>
<td>V0.5</td>
<td>30/11/2021</td>
<td>100%</td>
<td>Quality Assurance Review</td>
<td>Christos Skoufis (EBOS), Andrea Di Giglio (TIM)</td>
</tr>
</tbody>
</table>

---

<sup>1</sup> According to 5G Solutions Quality Assurance Process:

1 month after the Task started: Deliverable outline and structure
3 months before Deliverable’s Due Date: 50% should be complete
2 months before Deliverable’s Due Date: 80% should be complete
1 months before Deliverable’s Due Date: close to 100%. At this stage it sent for review by 2 peer reviewers
Submission month: All required changes by Peer Reviewers have been applied, and goes for final review by the Quality Manager, before submitted
**Disclaimer**

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the 5G Solutions consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the 5G Solutions Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the 5G Solutions Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

**Copyright message**

© 5G Solutions Consortium, 2019-2022. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.
Table of Contents

Table of Contents ........................................................................................................................................ 4
List of Figures ................................................................................................................................................ 5
1 Executive summary .................................................................................................................................... 9
2 Introduction ............................................................................................................................................... 11
  2.1 Mapping to Project Outputs ............................................................................................................. 11
  2.2 Deliverable Overview and Structure ............................................................................................... 14
3 Local Domain Service Orchestration & Management in 5G-SOLUTIONS ............................................... 15
  3.1 The 5G-VINNI orchestrator Norway ................................................................................................. 16
  3.2 The 5G-VINNI orchestrator Patras .................................................................................................... 18
  3.3 The 5G-EVE orchestrator .................................................................................................................. 19
  3.4 5G private node .................................................................................................................................. 21
  3.5 Local vertical orchestrator ................................................................................................................ 22
4 Approach to Multi-domain Service Orchestration in 5G-SOLUTIONS .................................................... 24
  4.1 Planned deployments & orchestration requirements for 5G-SOLUTIONS use cases .................... 24
  4.2 Cross-domain orchestrator CDSO .................................................................................................... 26
    4.2.1 Functionality of CBND ............................................................................................................. 26
    4.2.2 The CBND Plugin Mechanism ............................................................................................... 28
    4.2.3 Role of CDSO in 5G-SOLUTIONS .......................................................................................... 29
  4.3 Approach to multi-domain service orchestration in 5G-SOLUTIONS ............................................. 30
5 Challenges of Virtualisation and Orchestration in 5G-SOLUTIONS ......................................................... 33
6 Service Orchestration Optimisation Features ....................................................................................... 37
  6.1 Extensions Towards Introducing More Automation in the Orchestration Area ............................ 37
  6.2 Zero-Touch Automation closed loop control ................................................................................... 38
7 Conclusions and Next Steps .................................................................................................................. 41
Annex 1: Zero-Touch Automation API / usage ......................................................................................... 42
Annex 2: KPIs reported by 5G-VINNI Patras towards KPI VS ................................................................. 46
References ...................................................................................................................................................... 48
List of Figures

Figure 1: 5G-VINNI Norway Facility Network Management and Service Orchestration Overview............ 17
Figure 2: Platform Orchestration Solution offered by 5G-VINNI Patras ........................................... 18
Figure 3: 5G EVE architecture interfacing towards 5G-SOLUTIONS.................................................... 20
Figure 4: 5G EVE REST-API exposure block......................................................................................... 20
Figure 5: Detailed architecture of UC3.4 and external APIs for the KPI VS and the CDSO ..................... 22
Figure 6: Alignment of CBND architecture with the ETSI NFC MANO view ........................................... 27
Figure 7: General case with the CDSO as cross-domain orchestrator ..................................................... 31
Figure 8: Zero-Touch Automation closed loop and automation actors.................................................. 39
Figure 9: Example of output from the ML forecast module – predicted values for lost_frames KPI measured at decoder 172.16.10.240 in UC4.1. Based on predicted values such as the one in this plot, the ZTA module can take a proactive decision and send the recommended action to CDSO ............... 39
Figure 10: Example of a correlation matrix that can be used for additional insights when designing the ZTA rules. This matrix comes from our trials in LL4. ......................................................................................... 40
List of Tables

Table 1: Adherence to 5G Solutions GA Deliverable & Tasks Descriptions .................................................. 12
Table 2: Mapping of use cases to ICT-17 platforms .......................................................... 25
Table 3: 5G-VINNI Patras monitored KPIs and respective details.......................................................... 46
## Glossary of Acronyms

<table>
<thead>
<tr>
<th>Abbreviation / Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>SG</td>
<td>Fifth Generation</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CBAM</td>
<td>CloudBand Application Manager</td>
</tr>
<tr>
<td>CBND</td>
<td>CloudBand Network Director</td>
</tr>
<tr>
<td>CDSO</td>
<td>Cross Domain Service Orchestrator</td>
</tr>
<tr>
<td>CEE</td>
<td>Cloud Execution Environment</td>
</tr>
<tr>
<td>CN</td>
<td>Core Network</td>
</tr>
<tr>
<td>CNF</td>
<td>Cloud-native Network Function</td>
</tr>
<tr>
<td>DC</td>
<td>Data Center</td>
</tr>
<tr>
<td>E2E</td>
<td>End-to-End</td>
</tr>
<tr>
<td>eMBB</td>
<td>Enhanced Mobile Broadband</td>
</tr>
<tr>
<td>eMTC</td>
<td>Enhanced Machine Type Communication</td>
</tr>
<tr>
<td>EPC</td>
<td>Evolved Packet Core</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>G-VNFM</td>
<td>Generic VNFM</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISG</td>
<td>Industry Specification Group</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>KPI VS</td>
<td>KPI Visualisation System</td>
</tr>
<tr>
<td>LCM</td>
<td>Life Cycle Management</td>
</tr>
<tr>
<td>LL</td>
<td>Living Lab</td>
</tr>
<tr>
<td>MANO</td>
<td>Management and Network Orchestration</td>
</tr>
<tr>
<td>MEC</td>
<td>Multi-access Edge Computing</td>
</tr>
<tr>
<td>MIoT</td>
<td>Massive Internet of Things</td>
</tr>
<tr>
<td>MMME</td>
<td>Mobility Management Entity</td>
</tr>
<tr>
<td>NBI</td>
<td>North Bound Interface</td>
</tr>
<tr>
<td>NCIR</td>
<td>Nokia Cloud Infrastructure Real-time</td>
</tr>
<tr>
<td>NEI</td>
<td>Network Element Interface</td>
</tr>
<tr>
<td>NF</td>
<td>Network Function</td>
</tr>
<tr>
<td>NFV</td>
<td>Network Function Virtualisation</td>
</tr>
<tr>
<td>NFVO</td>
<td>NFV Orchestrator</td>
</tr>
<tr>
<td>NS</td>
<td>Network Service</td>
</tr>
<tr>
<td>NSA</td>
<td>Non-standalone</td>
</tr>
<tr>
<td>NSD</td>
<td>Network Service Descriptor</td>
</tr>
<tr>
<td>NSG</td>
<td>Network Service Graph</td>
</tr>
<tr>
<td>NSI</td>
<td>Network Service Interface</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>NSMF</td>
<td>Network Slice Management Function</td>
</tr>
<tr>
<td>NST</td>
<td>Network Slice Template</td>
</tr>
<tr>
<td>OSM</td>
<td>Open Source MANO</td>
</tr>
<tr>
<td>OSS</td>
<td>Operations Support Systems</td>
</tr>
<tr>
<td>PNF</td>
<td>Physical Network Functions</td>
</tr>
<tr>
<td>PoP</td>
<td>Point of Presence</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAN</td>
<td>Radio Access Network</td>
</tr>
<tr>
<td>SA</td>
<td>Standalone</td>
</tr>
<tr>
<td>SBA</td>
<td>Service Based Architecture</td>
</tr>
<tr>
<td>SDN</td>
<td>Software Defined Network</td>
</tr>
<tr>
<td>SFC</td>
<td>Service Function Chain</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SON</td>
<td>Self-organizing Network</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider</td>
</tr>
<tr>
<td>TOSCA</td>
<td>Topology and Orchestration Specification for Cloud Applications</td>
</tr>
<tr>
<td>TR</td>
<td>Technical Report</td>
</tr>
<tr>
<td>UC</td>
<td>Use Case</td>
</tr>
<tr>
<td>UDR</td>
<td>Unified Data Repository</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>uRLLC</td>
<td>Ultra Reliable Low Latency Communication</td>
</tr>
<tr>
<td>VIM</td>
<td>Virtualised Infrastructure Manager</td>
</tr>
<tr>
<td>VNF</td>
<td>Virtual Network Function</td>
</tr>
<tr>
<td>VNFaaS</td>
<td>VNF as a Service</td>
</tr>
<tr>
<td>VNFM</td>
<td>Virtual Network Virtualisation Manager</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>ZTA</td>
<td>Zero-Touch Automation</td>
</tr>
</tbody>
</table>
1 Executive summary

One of the goals of 5G is to provide seamless support to various types of industries and applications. Orchestration of services pertaining to different types of verticals in 5G is a complex issue. Several factors contribute to this, including the complexity introduced by network function virtualization and slicing, deployments that span across multiple domains each with its own orchestration solution, and because of the different types of needs those different verticals exhibit. In the context of the 5G-SOLUTIONS project, we devised our orchestration solution taking into account the particularities of each type of vertical, its services, and also the characteristics of the platform that the service is deployed onto. The platform itself and, in certain cases, the vertical, have their own orchestration solution that we had to align with.

Our service orchestration solution has at its centre the Nokia CDSO orchestrator (Cross-Domain Service Orchestrator), which acts as inter-domain orchestrator and coordinates with the orchestrators of the underlying platforms 5G-VINNI Norway, 5G-VINNI Patras, 5G-EVE and the 5G Amarisoft private node, as well as any orchestrator in the vertical domain (e.g., OSM [1] in the case of UC3.4). In the case where the vertical domain does not have an orchestrator, CDSO could also orchestrate virtual network functions / cloud-native network functions in the vertical domain.

Developing this orchestration solution started with bridging the gap between the vertical and the 5G platform domains, understanding the limitations and focus for each of the use cases and platforms, as well as the technical details of orchestration in each of the relevant domains. In order to do so, ongoing workshops have taken place in the project between CDSO owner, use case owners and platform owners. Throughout this process, a set of challenges from all actors involved became clear. These challenges are presented in Section 5 from the different points of view of the underlying ICT-17 platforms, the 5G-SOLUTIONS orchestrator and the different verticals.

The capabilities of CDSO and the specific roles it plays in 5G-SOLUTIONS are presented in Section 4. CDSO interfaces with orchestrators in the ICT-17 platform and vertical domains, implements service lifecycle management, orchestration of workflows between these domains, as well as automatic dynamic optimisations through the Zero-Touch Automation feature. Based on the orchestration requirements of each service in 5G-SOLUTIONS, we classified the services under categories A (use case fully integrated as an experiment onto the underlying ICT-17 platform), B (both the underlying platform and the vertical domain have local orchestrators) and C (manual orchestration mainly for private in-factory deployments). When implementing orchestration for these categories, the general architecture with CDSO acting as inter-domain orchestrator is mapped differently, with CDSO playing slightly different roles. This is explained in Section 4.3.

Additionally, 5G-SOLUTIONS has implemented a Zero-Touch Automation (ZTA) feature, which works with CDSO to enable service optimisations in a closed-control loop fashion. This loop includes monitoring the KPIs in the different domains (e.g., to check when service quality KPIs are getting under a certain threshold), acting on use case specific rules (e.g., recommending to the CDSO to influence the service workflow / parameters), the action of the CDSO towards the underlying orchestrators and finally observing the change in the monitored KPIs. In this loop, CDSO triggers the ZTA controller, acts on recommendations coming from ZTA by sending the action recommendation to the underlying orchestrators and stops ZTA. ZTA is hosted as part of our KPI Visualisation System, which stores all
collected KPIs from both vertical and network domains, to ensure fast access to data. Section 6 presents ZTA, explains the mechanism and comments on our current work in applying it in the context of 5G-SOLUTIONS.

The key achievements presented in this deliverable are:

- Orchestration features of the underlying platforms and verticals, focused on features used in 5G-SOLUTIONS
- Our approach to cross-domain orchestration, using CDSO as inter-domain orchestrator that coordinates with orchestrators from the platform & vertical domains; the different roles that CDSO fulfills in the 5G-SOLUTIONS, as well as the general architecture and its mapping to different use case requirements
- The main challenges from different points of view, including underlying platform, CDSO and verticals
- Extensions towards automation introducing our zero-touch automation feature in service orchestration

This deliverable is based on the current status of our work at M30 in the project. We also refer the reader to deliverables D2.2 [2] [3] for specific implementation details related to orchestration.
2 Introduction

The main purpose of this deliverable is to summarise the results of the work conducted in Task 1.2, regarding the challenges towards providing a suitable orchestration solution to a diverse number of use cases in 5G-SOLUTIONS and our approach to solving these challenges. These use cases fall under several categories, including Factories of the Future, Smart Energy, Smart Cities & Ports, Multimedia & Entertainment, and are deployed on several ICT-17 platforms, 5G-VINNI Norway, 5G-VINNI Patras, 5G EVE and the private 5G Amarisoft node. Each of these platforms implements its own orchestration solution while, in some cases, the vertical domain also has its own orchestrator. The different use cases and deployments result in different orchestration requirements, which influence how the orchestration solution is implemented.

We explain the main orchestration features in each domain (platform, vertical, CDSO), that were used in 5G-SOLUTIONS, the roles for our inter-domain orchestrator CDSO, as well as the main architecture, including building blocks such as local orchestrators, CDSO, and KPI Visualisation System (KPI VS) [4] with Zero-Touch Automation (ZTA) [5] feature. ZTA is our rule-based control loop that can be triggered by the CDSO for services where an automatic optimisation loop can be implemented. ZTA allows for proactive optimisations, by taking input from ML-based forecasting services deployed as part of the KPI VS.

This deliverable is based on our efforts in 5G-SOLUTIONS until M30, taking into account that some of the use cases in the project are still evolving their understanding of deployments and orchestration.

The core sub-tasks of Task 1.2, as presented in this deliverable, are:

- Understanding what orchestration solution is used in each of the domains included in 5G-SOLUTIONS services, i.e., 5G-VINNI Patras, 5G-VINNI Norway, 5G-EVE, 5G Amarisoft private node and verticals that use local orchestration in their own domain.
- Categorizing 5G-SOLUTIONS use cases based on their orchestration needs, as some use cases are deployed and orchestrated using the ICT-17 platform orchestrator as sole resource orchestrator, while others rely on multiple local orchestrators (i.e., both platform and vertical orchestrators).
- Understanding the main challenges of multi-domain slicing, taking into account different points of view, including the platform provider, the different verticals service providers and the inter-domain orchestrator.
- Proposing directions for optimisations, going towards zero-touch automations which have the ability to adjust service performance on the fly.

2.1 Mapping to Project Outputs

This section maps the 5G-Solutions Grant Agreement commitments, both within the formal Deliverable and the Task descriptions, against the project’s respective outputs and work performed.
### Table 1: Adherence to 5G Solutions GA Deliverable & Tasks Descriptions

<table>
<thead>
<tr>
<th>Project GA Component Title</th>
<th>Project GA Component Outline</th>
<th>Respective Document Sections(s)</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TASKS</strong></td>
<td><strong>Task 1.2:</strong> Analysis of challenges and requirements for 5G E2E multi, concurrent and cross-domain service management and orchestration including interaction with ICT-17 facilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The evolution of NFV is faced with the reality that the initial MANO architecture is not adequate to address the service providers' underlying requirement to instil the agility of DevOps principles. Rather than setting up and managing services in network domains separately and manually, the expectation on 5G networks is that cross-domain E2E services will be provisioned, deployed, managed, and retired in a DevOps manner, with minimal manual intervention. In addition, where slicing capabilities are exposed by network domains, those E2E services will be deployable on network slices. We will also explore the integrity of OSS features being kept under cross data correlation from low level virtualisation and the vertical industry requirement, which is agnostic of the underlying telecom infrastructure. The core focus of Task 1.2 is to explore the above challenges in the context of a framework that aligns with industry standards, yet is innovative and provides leadership beyond what is included today in current approaches.</td>
<td>Section 3 - 6</td>
<td>Section 3 presents the orchestration solutions of the different domains involved in providing services in 5G-SOLUTIONS. The ICT-17 platforms and vertical orchestrators are all aligned with standard orchestration solutions and use standard interfaces towards our own inter-domain orchestrator. Our inter-domain orchestrator is defined in Section 4, including its main features, roles that it fulfills in the project, and also the general architecture, mapped onto different use case requirements. Specific challenges from different points of view are included in Section 5. Section 6 presents our automation solution, based control loop that allows the CDSO to optimize service delivery proactively on the fly.</td>
</tr>
<tr>
<td></td>
<td><strong>End-to-end service provisioning with slicing capabilities:</strong> In this sub-task, we shall design how to leverage the NOKIA orchestrator to support end-to-end inter-domain lifecycle management. We will make use of the available plugin interfaces to extend the functionality to allow management of closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sections 5&amp;6</td>
<td>Section 5 describes the features of CDSO that we leverage in 5G-SOLUTIONS, how it is integrated with the local orchestrators and different approaches to address the</td>
</tr>
</tbody>
</table>
loop services for network slicing. In addition, the plugins implementing the interfaces towards the network domain orchestrators will be designed to allow configuration and monitoring of the virtual and physical resources that are providing the network slices that the end-to-end services are running on. We will also investigate the issue of monitoring the verticals SLAs and analyse the data coming from the verticals to allow for efficient orchestration of the end-to-end services. The interaction and integration with the ICT-17 facilities (south-bound to 5G-SOLUTIONS) will also be accommodated, as well as north-bound to customer facing systems or portals.

**Proposed solutions of slicing issues between core and access in 5G mobile networks:** The radio and core networks differ in the manner in which they implement slicing and in which such slices are configured. We will examine how services using network slices across the RAN and core networks can be provisioned and managed transparently. We will investigate how common slices can be created across the two domains, and how traffic can be routed and shaped between these slice fragments whilst preserving a common view of the end-to-end slice. In addition, we will investigate how services running across slices can be optimised using predictive optimisation - we will try to understand how common slicing patterns can be predicted, in an effort to formulate a set of training patterns that can be used to train predictive and proactive configuration changes.

**Sections 4-6**

Section 4 presents our approach to cross-domain orchestration. Section 5 includes details on challenges met by the different providers, i.e. 5G platform / vertical / inter-domain orchestrator provider, including limitations coming from the APIs exposed to the CDSO. Section 6 presents our solution to introducing proactive automation in service delivery.
D1.2B: Cross-domain orchestration and management challenges analysis v2.0

Final (v2.0) report defining the management of the life cycle of services running across different 5G mobile networks including solutions for slicing and virtualisation beyond MANO

| PROJECT OBJECTIVE |
|--------------------|-----------------|-----------------|
| Objective 2:       | To analyse key challenges for performing multi and cross-domain service provisioning leveraging slicing and virtualisation technologies | Sections 4-5 |
| Technology         | Key challenges are presented in Section 5 from different points of view of the different actors involved. Our approach to cross-domain service provisioning is explained in Section 5. |
| development        |                                                              |
| and readiness      |                                                              |

2.2 Deliverable Overview and Structure

This deliverable is organized as follows:

- **Section 3** discusses orchestration features in the different domains included in 5G-SOLUTIONS, focusing on the orchestration offered by the underlying platforms, 5G-VINNI Norway, 5G-VINNI Patras, 5G EVE, and the private 5G node from Amarisoft. Additionally, it presents local vertical orchestrators, such as the one used in UC3.4.

- **Section** Error! Reference source not found. introduces the 5G-SOLUTIONS service orchestration solution, based on CDSO, including roles played by CDSO in 5G-SOLUTIONS and general architecture.

- **Section** Error! Reference source not found. presents challenges met in developing the 5G-SOLUTIONS orchestration solution from the different points of view, including the underlying platforms, CDSO, and the different LLs.

- **Section 0** discusses our approach to introducing orchestration optimisations based on Machine Learning techniques and the Zero-Touch Automation feature.

- **Section 7** summarizes the work done in Task 1.2 and presents our conclusions.

Additionally, Annex 1 includes a description of the ZTA endpoints with examples of usage, as well as giving an example on how to configure ZTA through use-case specific rules. Annex 2 presents the KPIs reported by 5G-VINNI Patras towards KPI VS that are used for ZTA purposes.
3 Local Domain Service Orchestration & Management in 5G-SOLUTIONS

Service orchestration is the execution of the operational and functional processes involved in designing, creating, and delivering an end-to-end service. Traditionally, these processes were handled by domain-specific, siloed operational support systems and tools built for static environments. 5G systems, though, come with a different architectural view (TS 23.501 [6]), which proposes that the system functionality is achieved by a set of Network Functions (NFs) that provide services to other authorized NFs to access their services. This Service-Based Architecture aims to decouple end-user services from the basic network and the platform infrastructure, in order to speed up the network deployment and thus the service delivery. It comes with a range of benefits, such as flexibility in the allocation of NFVs in general-purpose hardware, rapid implementation and deployment of new network services, support for multiple versions of services and multi-tenancy scenarios, reduction of CAPEX & OPEX through efficient energy usage and automation of operational processes. However, this creates a much more dynamic environment, in which the different NFs must be orchestrated. In parallel to this, the introduction of software driven capabilities and virtualization means that service orchestration must take into account a much more dynamic and complex service provider environment.

5G-SOLUTIONS aims to provide a variety of services operating on top of different 5G platforms. The services offered are grouped into 4 living labs (LLs): Factories of the Future, Smart Energy, Smart Cities & Ports, Media & Entertainment. The 5G platforms that these services are deployed on are: 5G-VINNI Norway, 5G-VINNI Patras, 5G-EVE, and Amarisoft Private Node. Additionally, some use cases in these LLs also include functionality that runs in their own premises. As such, an end-to-end service orchestration solution must take into account this diversity, including different orchestration solutions within the local domains, APIs offered from the platforms and verticals towards the cross-domain orchestrator, the different goals of each service, etc.

In order to achieve the 5G goals in terms of dynamicity, performance and ease of management, an end-to-end orchestration is required through the multiple domains. In the case of 5G-SOLUTIONS, this translates to slightly different orchestration solutions, based on how the use case is deployed, i.e., if the use case is fully integrated as a test on the underlying 5G-EVE and 5G-VINNI facilities, or whether it also uses functionalities that reside outside these platforms. We aim to provide a coherent service orchestration solution based on Nokia’s CDSO (Cross Domain Service Orchestrator), which can provide a range of features towards this. These are described in Section 3.5, with full details being included in deliverables D2.2 (versions A, B, C and D). However, based on the particularities of the service deployment and the goals of the service, a sub-set of these features might be used for each 5G-SOLUTIONS use case. For example, in the case where the use case is defined as a test on top of 5G-VINNI and 5G-EVE, the underlying platforms will handle platform end-to-end orchestration across their different technological domains. Therefore, in this case, the role of the CDSO will be to interact with the underlying platform for service lifecycle operations. However, in the more complex case where the service is deployed on both the platform and service provider infrastructure, the CDSO will have a more active role into orchestrating across these different domains, as explained in Chapter 4. Additionally, we have designed Zero-Touch Automation features (as part of Task 2.4), which enable the CDSO to have a proactive role in the service orchestration based on changing underlying conditions in the network / service KPIs. Also, in the context
of the Multi-LL use cases (LL5), CDSO will play a more active role orchestrating the workflow involving the different services of the Multi-LL use case.

In this section, we look at the main features offered by the domain orchestrators of 5G-VINNI Norway, 5G-VINNI Patras, 5G-EVE, and the 5G private Amarisoft node, as well as local orchestration in the vertical domain. We also describe the main features of CDSO. Please note that this description builds upon the earlier version of Deliverable D1.2A [7], where we included more details on the orchestration solutions of the underlying platforms and CDSO. In this version, we focus on the specific features used when deploying 5G-SOLUTIONS use cases, as well as interfaces both towards CDSO and KPI VS, which influences our Zero-Touch Automation solution presented in Section 6.

3.1 The 5G-VINNI orchestrator Norway

The 5G-VINNI Norway facility [8] provides 5G network services to LL1, LL3, and LL5 use cases, supporting 5G network slices [9], both stand-alone (SA) and non-stand-alone (NSA). Its orchestration system [10] is supplied by Nokia’s proprietary solutions, including (see Figure 1Error! Reference source not found.):

- VIM: NCIR (Nokia Cloud Infrastructure Real-time) based on OpenStack.
- VNF: CBAM (Nokia CloudBand Application Manager) that is responsible for onboarding and managing 3rd party application VNFs. Note that for the 5G core VNFs, 5G-VINNI Norway uses Ericsson’s Network Management (ENM) for the VNFs lifecycle management (LCM).
- NFVO: CBND (Nokia CloudBand Network Director) that is connected to VIM, VNFM, and SDN orchestrator.
- Service Orchestrator: FlowOne that is responsible for the slices LCM. It supports TMF Open APIs facing OSS and external customers like 5G-SOLUTIONS.
- SDN orchestrator for the cloud networking: Nuage SDN platform.

In addition, 5G-VINNI Norway facility deploys:

- An open-source OSS, called OpenSlice, that is acting as a portal to vertical customers like 5G-SOLUTIONS. OpenSlice supports TMF Open APIs, as explained below. It interacts with FlowOne to order slicing services and acquire the status of the ordered services.
- Test-as-a-Service (TaaS) platform that is located in the cloud of Keysight, as an automation platform for testing services. It provides proprietary APIs for managing the test cases/campaigns and retrieving the collected KPIs.
If 5G-SOLUTIONS UCs only need to consume the 5G network slices provisioned by the 5G-VINNI Norway facility, they mainly interact with OpenSlice (via CDSO) to order or subscribe to a network slice.

If 5G-SOLUTIONS UCs plan to add their own or 3rd party application VNFs into the 5G-VINNI infrastructure, then extra interactions would be needed. First, these 3rd party VNFs need to be prepared by UC owners in compliance with the requirements of Nokia VNFM (i.e., CBAM), which is acting as a generic-VNFM (g-VNFM) to manage the 3rd party VNFs. The preparation includes the following steps:

- Prepare the VNF descriptor (VNFD) in TOSCA
- Upload the image to VIM
- Package the VNF
- Validate the VNFD and the VNF package.

Once these steps are complete, the 3rd party VNF is considered as ready for onboarding. Then CDSO can request to order and activate the VNF, as well as other 5G network slices.

Note that for UCs requiring edges, additional efforts will be made to plan and decide which party will be responsible for orchestrating the edge sites, the options being either the 5G-VINNI orchestration system or the vertical local orchestrators themselves. This will be a joint decision among the UC owner, CDSO, and 5G-VINNI facility owner, preferably to be considered in late Cycle 2 or early Cycle 3.

In terms of APIs available from the 5G-VINNI Norway facility, CDSO can interact with the OSS (OpenSlice) via TMF Open APIs [11], as follows:

- TMF633: for service catalog
- TMF641: for service ordering
• TMF638: for service inventory

Both CDSO and KPI-VS can interact with TaaS via APIs, provided by Keysight. In terms of KPI reporting, this means that the KPIs are reported at the end of the experiment, not throughout the experiment. This might limit our options in terms of automatic reconfigurations/optimisations enabled through the ZTA feature presented in Section 6.

3.2 The 5G-VINNI orchestrator Patras

5G-VINNI Patras facility [8] supports the use cases of LL4, Multimedia & entertainment.

The orchestration solution in this platform comes with the following characteristics:

• OSS: Openslice. The Openslice instance allows external customers (e.g., CDSO) to order slicing services from the 5G-VINNI Patras facility.
• NFVO: OSM. The OSM platform handles the orchestration of virtual networks, VMs and Containers onto the underlining cloud infrastructure.
• VIM/CIM: OpenStack/Kubernetes.

![Figure 2: Platform Orchestration Solution offered by 5G-VINNI Patras](image)

The platform orchestration solution offers a number of features, which are essential to the 5G-SOLUTIONS verticals. These features include the following:
- Openslice, which is responsible for interfacing with the 5G-SOLUTIONS CDSO, in order to accept service orders and report back the service status, as well as for communicating with the northbound interface of the NFVO, for orchestrating Network Services.
- OSM, which plays the role of NFVO and is responsible for the orchestration of VNFs and CNFs at the respective infrastructure, as well as for running day-2 actions on orchestrated machines, as needed by specific use-cases, with the aim of changing their configuration at runtime, without having to redploy them, hence, reducing downtime.
- OpenStack provides the underlying orchestratable cloud infrastructure, which OSM interacts with, in order to deploy VNFs.
- Kubernetes, similar to OpenStack, provides the infrastructure for orchestrating CNFs inside a cluster.
- Various KPI collectors, which collect data from target machines and then aggregate that data into a cohesive database, before forwarding it towards the external visualization system.

One thing to note is that during Cycle 1 of experimentation in 5G-SOLUTIONS, the 5G-VINNI Patras infrastructure did not support multiple slices co-existence, and therefore, the tests were run on a single slice. However, from Cycle 2 onwards, multi-slicing is supported and therefore, different slice types will be used for the respective tests.

The platform orchestration solution offers an interface towards the CDSO to order and manage the 5G network services. More specifically, the API offered is the Openslice northbound API, which is a TMForum OpenAPI, similar to 5G-VINNI Norway.

In terms of reporting towards KPI VS, the API used to access the 5G-VINNI Patras KPIs is the Prometheus RESTful API. The types of KPIs that are reported are shown in Annex 2. These KPIs will be used in the ZTA automation, as described in Section 6.

3.3 The 5G-EVE orchestrator

5G EVE [12] is the facility proposed to provide field trial connectivity to the LL2 use cases. Figure 3 shows a detailed representation of the framework used in 5G-EVE.

We note the following features that are relevant to 5G-SOLUTIONS:

- 5G EVE provides end-to-end provisioning of 5G network services. The facility exposes a portal providing north-bound interfaces for CDSO and KPI VS. These interfaces allow third part modules to obtain information contained in the Interworking Framework.
- CDSO is connected to 5G-EVE facility, where the facility offers the capability to start/terminate the deployment experiment. This north-bound interface additionally allows to retrieve the list of running experiments, as well as deleting data for terminated experiments. It is not possible to define slices through this interface.
- KPI VS is used to collect network parameters and service parameters, in order to validate the vertical UC. This KPI VS exploits 5G EVE’s north-bound interface to obtain information contained in the Data Collection Manager (DCM).
- A Management and Orchestration (MANO) module is used to deploy cloud edge resources needed to perform vertical UC.
The 5G EVE infrastructure is connected via REST APIs at 5G EVE portal level, which is a global 5G EVE feature. By this interface, multiple slices can be requested, in any 5G EVE site facility, including the site in Turin, where LL2 use cases run. The local orchestrator will orchestrate local resources but does not expose any information about this to 5G-SOLUTIONS.

The CDSO is connected to the 5G-EVE REST API exposed by 5G-EVE to access to Experiment Lifecycle Manager (ELM) module as describe in Figure 4 Error! Reference source not found. below.
• Retrieve the experiment list.
• Retrieve the experiment status.
• Deploys the experiment.
• Terminates the experiment.
• Removes an experiment and its record from the system.

From the perspective of KPI reporting, the “Results Analysis and Validation” (RAV) module is used for the evaluation of the results: calculating KPI, raw metric, validating and test reporting. The “Data Collection Manager” (DCM) receives metric information from the infrastructure, the software architecture of the DCM is based on the Apache Kafka Bus. Subscribing on a specific Kafka topic allows the reception of system KPI and performance associated with the topic. The subscription is allowed only to the modules of 5G EVE platform.

KPI-VS collects and filters the KPI data from 5G-EVE immediately after the stop of the experiment. This has implications in terms of considering ZTA in the context of LL2, as ZTA needs real-time reporting of KPIs throughout the experiment.

3.4 5G private node
The 5G private node supports use cases in LL1 and LL3.

This private 5G node from Amarisoft supports both the 5G Core and RAN functionalities. The 5G RAN has a built-in WebSocket API that allows the configuration of the 5G slice. However, its 5G Core consists of a closed source binary code used as a monolithic deployment in the 5G base station. This creates a big problem in terms of the orchestration and the integration with the CDSO, as the notion of VNFs does not exist in this implementation and no custom configuration APIs are exposed by the 5G Core.

This issue was mainly investigated in the context of UC3.4, and the solution is planned to be extended to some of the LL1 UCs in Cycle 2 and Cycle 3 of the 5G-SOLUTIONS experimentation. The alternative option explored during Cycle 1 was based on an open source 5G Core named Open5GS. This 5G Core is implemented as a virtual machine (VM) in its current version, which is not ideal. However, we are following closely the progress of this opensource project/progress by University of Patras in that respect, for any new versions supporting separate 5G Core VNFs. Initial tests to integrate Open5GS with the 5G RAN of the private 5G node were performed and the integration was successful.

In this case, CDSO will enable orchestration across the RAN of Amarisoft and application VNFs, by interfacing with the vertical OSM, which will allow the orchestration of the application VNFs and 5G RAN through the WebSocket API. The WebSocket API defined in RFC 6455 [13] exposes a number of configuration parameters that will allow the Nokia CDSO to authenticate itself, retrieve or configure radio parameters of the eNB and gNB of the base station (such as the Modulation Coding Scheme, cell gain and TX gain) as well as starting and stopping the services of the network slices to be deployed (in the case of UC3.4, slices of type mMTC, eMBB, or both).

The integration between the 5G private node, the local vertical orchestrator and CDSO is shown in Figure 5 below.

When it comes to the KPI Visualisation System, the integration has been completed in Cycle 1 and consists of a REST API that is exposed by the KPI VS and allows the network and application KPI data to be pushed periodically from the 5G facility. We refer the reader to Deliverable D3.2B for a detailed description of the API.
3.5 Local vertical orchestrator

In order to investigate the issue of the limited virtualization capabilities in the private 5G node, open5GS [14] – an open-source implementation of the 5G Core – was used to replace the native 5G Core provided in the private node. This opened up the possibilities to use a local orchestrator for the orchestration of the application VNFs and potentially 5G core using Open Source Mano (OSM) [15]. This local orchestrator will be integrated with the CDSO through the OSM northbound API or using OpenSlice (to be confirmed in the final drop of Deliverable 2.2). Figure 5 shows the framework used in UC3.4, which is the first use case to investigate this direction, as well as the APIs for the CDSO and the KPI VS.

This figure shows two integration points for the CDSO and one for the KPI VS. While the integration of the KPI VS has been completed in Cycle 1, the CDSO integration is currently in progress and is due in Cycles 2 (initial integration of the 5G RAN) and 3 (full integration of the application VNFs).

In terms of the integration with the CDSO, this will be subdivided into two main parts, the integration of the 5G RAN part through a WebSocket API, as explained in the previous section and the integration of the 5G Core part using the OSM API. The WebSocket API is well defined and allows the CDSO to retrieve or configure radio parameters of the eNB and gNB of the base station, as well as start and stop the services of the network slices to be deployed.
The integration of the CDSO with OSM will be decided and detailed in the last drop of Deliverable D2.2D. Current thinking is along the lines of integrating through Openslice, similar to the integration between CDSO and 5G-VINNI Patras.
4 Approach to Multi-domain Service Orchestration in 5G-SOLUTIONS

This section presents our approach towards multi-domain orchestration in the context of the various LLs and underlying platforms support. We first show the orchestration requirements for the different UCs in 5G-SOLUTIONS. These requirements are grouped into 3 categories, which will result in slightly different roles to be fulfilled by CDSO.

The CDSO's main features are presented, its roles in 5G-SOLUTIONS, as well a general architecture that shows CDSO's role as an inter-domain orchestrator. A discussion follows on the differences between instantiating the architecture for the different categories of orchestration requirement, as presented for each UC.

4.1 Planned deployments & orchestration requirements for 5G-SOLUTIONS use cases

In order to understand what type of orchestration we need to support the different verticals in our Living Labs (LL), we present below a concise table to show for each use case what platform it will be deployed onto and what type of orchestration requirements it has. Based on high-level requirements, the UCs have been classified into the following categories:

1) Category A: Use cases that are hosted in an existing ICT-17 facility and are already orchestrated by the orchestrator in that project (i.e., 5G-VINNI Norway, 5G-VINNI Patras, 5G EVE). In these use cases, we assume that the ICT-17 orchestrator is already integrated (or will be integrated) with the use case and we will work with the platform owners to integrate the 5G-SOLUTIONS orchestrator with each facility. To this end, we'll perform the following service lifecycle actions:
   a. **Activate Service** – will trigger the service deployment for the given service.
   b. **Get Service Status** – will query the local orchestrator for the status of the deployed service.
   c. **Subscribe to notifications** – In case the orchestrator supports this, we will subscribe to notifications and display them in our event stream.
   d. **Terminate Service** – will trigger termination of the service.

   It is expected that in all ICT-17 projects we integrate with, the API exposed is HTTPS/REST based, there is connectivity from the 5G-SOLUTIONS orchestrator to that API (directly or via VPN) and credentials are provided.

2) Category B: Use cases not hosted in an existing ICT-17 facility. The use case may be associated with an ICT-17 project and even include some 5G components provided by that project (e.g., 5G radio [16]) but the use case is considered as Category B if the ICT-17 orchestrator is not handling this use case end to end.

   In this category, there are several sub-cases:
   a. The use case has its own MANO orchestrator orchestrating it – in this case, we should treat this like Category A and trigger the lifecycle actions as done in Category A use cases.
b. The use case needs to be orchestrated but does not have an orchestration solution. In this case, the 5G-SOLUTIONS orchestrator should provide end-to-end orchestration (as a MANO NFVO). In this case, we may also have to create the NSD and add it to our catalog.
c. The use case is orchestrated manually – in this use case, we should collaborate with the use case owner to decide how this use case is reflected in the 5G-SOLUTIONS orchestrator.

3) **Category C**: Use cases that are not hosted by an ICT-17 facility and do not require any orchestration as they reside inside a factory and have a standalone 5G network. These use cases will have to perform all the experiments manually and will not be connected to CDSO.

These planned deployments and requirements represent the status at M30 for the use cases. As mentioned before, some UCs are still in the phase of actively defining more virtualized functions and orchestration needs, and therefore we expect that in the next 12 months these categories might change for some of the UCs. For example, for LL1 there are active ongoing discussion in the area of orchestration, where some UCs were defined as category C and the investigation is trying to find a solution towards moving them to category B. However, this snapshot at M30 is very important in understanding the interaction and boundaries between 5G-SOLUTIONS and the underlying ICT-17 platforms that we use in terms of orchestration, and in assessing the current status for each UC.

Table 2: Mapping of use cases to ICT-17 platforms

<table>
<thead>
<tr>
<th>LL</th>
<th>UC#</th>
<th>Use Case Title</th>
<th>Responsible Partner(s)</th>
<th>ICT-17 Project</th>
<th>Location</th>
<th>Orchestration category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>Time-critical process optimization inside digital factories</td>
<td>P&amp;G, IRIS</td>
<td>5G-VINNI Patras</td>
<td>Private node in P&amp;G Belgium</td>
<td>TBD (C/B)</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>Non-time-critical communication inside the factory</td>
<td>GLAN, IRIS</td>
<td>5G-VINNI Patras</td>
<td>Private node in Glanbia Ireland</td>
<td>TBD (C/B)</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>Remotely controlling digital factories</td>
<td>NTNU</td>
<td>5G-VINNI Norway</td>
<td>Trondheim</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>Connected goods</td>
<td>P&amp;G, IRIS</td>
<td>5G-VINNI Patras</td>
<td>Private node in P&amp;G Belgium</td>
<td>TBD (C/B)</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>Rapid deployment, auto/re-configuration, testing of new robots</td>
<td>NTNU</td>
<td>5G-VINNI Norway</td>
<td>Trondheim</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
<td>Industrial Demand Side management</td>
<td>A2T, IREN</td>
<td>5G EVE</td>
<td>Turin</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>Electrical Vehicle Smart Charging</td>
<td>A2T, IREN</td>
<td>5G EVE</td>
<td>Turin</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>2.3</td>
<td>Electricity Network Frequency Stability</td>
<td>A2T, ENEL X</td>
<td>5G EVE</td>
<td>Turin</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>3.1</td>
<td>Intelligent Street Lighting</td>
<td>NURO, TNOR</td>
<td>5G-VINNI Norway</td>
<td>Oslo</td>
<td>A</td>
</tr>
</tbody>
</table>
4.2 Cross-domain orchestrator CDSO

5G-SOLUTIONS cross-domain service orchestration solution is based on Nokia’s CBND (acting as CDSO cross domain orchestrator). Its functionality relevant to 5G-SOLUTIONS is presented in the following sections.

4.2.1 Functionality of CBND
CBND automates the lifecycle management of VNFs/CNFs and Network Services and the corresponding virtualized network resources in full consistency with defined business priorities and relevant SLA parameters. The dynamic network level elasticity management aims to align the resource consumption with the actual business priorities. CBND is aligned with the NFV Orchestrator functionality as defined by ETSI NFV MANO GS [17] and comes with an easy-to-use interface for defining and provisioning the network services. Another advantage of using CBND is the use of well-defined open interfaces. Figure 6 shows the CBND components alignment with the ETSI NFV MANO [16] view.

The functionalities provided by CBND can be broadly classified into the following:

- VNF lifecycle management – deploy, configure, update, terminate.
- NS lifecycle management – deploy, configure, update, terminate.
- Closed loop automated operations (such as NS LCM, custom workflows, VNF LCM etc.) based on the changing dynamic conditions of NS, VNF, managed resources and additional external event.
- Catalog of Virtual Network Functions Descriptors, Network Service Descriptors, and virtual networking components described using TOSCA.
- Inventory of NFVI resources, deployed network services and VNFs.
- VNF and NS scaling to handle varying network loads.
- VNF and NS monitoring for fault and performance.
- Disaster recovery management.
- Policy management framework for rule-based lifecycle management and orchestration.
- Multi-vendor, multi-technology and multi-cloud stack support for orchestration.
- RESTful open interfaces for efficient multi-vendor Telco cloud management.
CBND has been designed based on plugin architecture principles, keeping the core business logic of orchestration intact, while supporting a varying ecosystem for orchestration. This is one of the reasons why the integration of CBND with the underlying platforms solutions is feasible.

CBND maintains a global view of all available and used NFVI resources across multiple data centers and clouds. It is normally integrated with multiple VNF Managers from different vendors to orchestrate network services composed of network functions from several vendors. This is dependent on how open the underlying platforms are, in terms of allowing access to their VNFMs.

CBND is based on the concept of Network Service and Network Service Graph (NSG). The NSG describes a set of closely inter-working VNFs, which together realize higher level Network Services. Some NSGs may be directly associated to business services, and some may provide additional network resources that are required for the implementation of the business services. The NSG Template lists the VNFs composing the Forwarding Graph (a Forwarding Graph is subset of NSG closely inter-working VNFs) and also specifies how VNFs are interconnected.

As the evolution of the mobile network towards a fully virtualized architecture and 5G is envisioned to take place in a phased approach, the connection of still existing traditional physical resources is also described. The CBND stores the available Network Service Packages in the Network Service Graph Template Database.

CBND needs to interact with other products which are needed to realize the solution. The list of other products which form Nokia’s CloudBand solution are the following:

- CloudBand Application Manager (VNF Manager).
- CloudBand Infrastructure Software (including Virtualized Infrastructure Manager (VIM)).

CBND is a Multi-Vendor solution, there are business cases when CBND needs to interact with the VNFM of other vendors to instantiate and manage multi-vendor VNFs. CBND needs to interact with different VIMs like VMWare, OpenStack, etc. For this interaction, specific plugins can be defined. All these interactions are dependent on the APIs coming from the local domain orchestrators.

### 4.2.2 The CBND Plugin Mechanism

In CBND, there is a plugin mechanism that allows easy customization of the entities managed and orchestrated by CBND. By default, the following types of entities are supported:

- **VIM** – Virtual Infrastructure Manager, a cloud infrastructure management such as Openstack or VMWare.
- **VNFMI** – Virtual Network Function Manager, a management system to manage the lifecycle of virtual network functions (networking applications running in the cloud).
- **SDN Controller** – a software defined networks controller.
- **NF** – Network Function, a generic entity with network connectivity.

The plugins can be created for any of the above types or there is also an option to define new types though most new types can be modeled via the generic NF type. A plugin consists of the following components:

- **XML file** (mandatory), an XML defining the plugin.
• **Plugin Code** (optional), a piece of SW code implementing REST APIs to support the plugin. Code can be implemented in any programming language but most CBND plugins are in Java thus the code examples are also in Java. Not all plugins require code. Code is needed for example (1) when you want to provide monitoring for the entity or (2) when you need to implement an interface with a non ETSI compliant VNFM.

• **Mistral Workflow** (optional), a workflow to support the specific plugin behavior.

• **TOSCA block** (optional), a Tosca-based definitions block to support the specific plugin behavior.

The following flows can be supported by the plugin mechanism:

1. **Define new inventory type**: Define the managed entity to be supported.

2. **Status Monitoring**: Define how to monitor the status of the new entry defined. Requires definition of a REST API to return the status according to a known format. (Note: some types support additional monitoring metrics (e.g., VIM).

3. **Connect to Network Service**: Define how to connect the entry to be part of an orchestrated service. This requires the definition of the parameters that will be requested when deploying the network service.

4. **Define a new type**: When existing types are not sufficient, can define new types (will require code). Can define the parameters needed to create the entity. Can validate the parameters and provide options.

5. **Define a new Mistral Workflow**: May be needed if the flow of operations when deploying the network service is different from the default.

6. **Physical Network Function (PNF)**: By default, the TOSCA block supports definition of a PNF but all needed functionality to support it has to be provided (e.g., the supporting code).

In 5G-SOLUTIONS, specific plugins have been developed for CBND, two examples being the TMF API plugin, for the interface with TMF 633 and TMF 641 APIs, and the Websocket plugin, for the interface with the 5G private Amarisoft node. More details of specific plugins and their implementations are described in deliverables D2.2 (A, B and C).

### 4.2.3 Role of CDSO in 5G-SOLUTIONS

Nokia CBND, acting as the CDSO (Cross Domain Service Orchestrator) fulfils the following roles in 5G-Solutions:

• Interact with the ICT-17 facility orchestrators to launch and configure 5G services.

• Configure network slices (as supported by the ICT-17 facility orchestrator’s API).

• Configure VNFs/CNFs via the ICT-17 facility orchestrator’s API (including on-the-fly configuration during the experiment, supporting the ZTA feedback loop).

• Activate the 5G services to launch the experiment.

• Terminate the 5G services when the experiment is finished.
• Poll 5G service status.
• Configure vertical applications residing outside the ICT-17 infrastructure (using their supported API, e.g., specific service parameters exposed by the service or interfacing through a local orchestrator’s API).
• Activate/Deactivate such vertical applications.
• Poll application status.
• Orchestrate/Automate MLL experiment to make sure the concurrent running of experiments has consistent timing and is in sync.

In order to support the above roles, joint workshops have been ongoing between CDSO and the ICT-17 facility owners together with the UC owners to define the following (as reflected in D2.2C):

• Define the APIs supported by the ICT-17 facility (in some cases these are defined by the facility owners while in other cases this requires mutual work).
• Define the architecture of the UC in a way that orchestration will be possible.
• Define the APIs supported by each UC.
• Define the experiment flow for each UC and map it to the above APIs.
• Create specific workflows and plugins in Nokia CBND to support the above flow.
• Work with all actors to finalize the integration and make sure the defined flows are working properly and the experiment can be performed.
• Integrate with the KPI VS to support notifications and the ZTA feedback API.

4.3 Approach to multi-domain service orchestration in 5G-SOLUTIONS

In its most general form, CDSO acts a cross-domain service orchestrator between the underlying ICT-17 platforms and the vertical domain. Because the platforms and/or verticals have an orchestration solution that is closed to the outside (i.e., they don’t allow external orchestrators to interfere with the internal solution), CDSO acts only as an inter-domain orchestrator. Its main responsibilities are service lifecycle management and service workflow orchestration, for which CDSO interacts and collaborates with the underlying orchestrator(s) for onboarding and configuration of services, service activation/deactivation and polling for status. CDSO does not have access to the resource orchestration in the local domains, but instead interfaces through standard interface to these domains to fulfill the roles specified above.

As explained in before, there is a very big variation between the different deployments and requirements from the different LLs and UCs. This means that the overall orchestration solution will have different flavours for the different categories of UCs, resulting in the CDSO playing a subset of the roles mentioned above in certain cases. This is because for many UCs, they will be completely integrated with the underlying platform and rely only on the platform’s orchestrator for resource orchestration. The orchestration inside the ICT-17 platforms is closed to external orchestrators, in the sense that the external orchestrator must rely on the resource orchestration performed by the local orchestrator. In this case, CDSO will mainly take the role of service lifecycle orchestrator. In other cases, where the vertical domain also has an orchestrator for its VNFs/CNFs, CDSO will take on the role of inter-domain orchestrator, coordinating with both the platform and the vertical orchestrators. Moreover, if the vertical resides on
the edge/cloud platform and is not orchestrated, CDSO could play the role of resource orchestrator for this particular case.

Additionally, CDSO plays a role as part of Zero-Touch Automations by enabling proactive reconfigurations to services, based on near real-time monitoring. This is explained in more detail in Section 6.

In Figure 7 below, we present the general case, and we explain in the following paragraphs which components are involved if the use case belongs to category A or B or C (as defined in Section 4.1) in terms of orchestration.

![Figure 7: General case with the CDSO as cross-domain orchestrator](image)

The following modules are involved:

1. 5G platform orchestrator (for 5G-VINNI Norway, 5G-VINNI Patras, 5G-EVE, Private 5G Amarisoft Node) – responsible for the orchestration of resources and services within the platform
2. Local orchestrator in the vertical domain. Some UCs have their own local orchestrator in their domain (e.g., UC3.4, local orchestrator based on OSM, as described in D2.2C).
3. CDSO used as cross-domain orchestrator (its specific role depends on the particularities of each UC)
4. KPI VS, the visualization system of that project, which collects all KPIs from platforms and verticals, provides statistical and ML methods for data analysis, and also hosts the Zero-Touch Automation module used by CDSO for runtime optimisations during the experiment. The functionality related to Zero-Touch Automation is detailed in Section 6 of this document. The ML and the Zero-Touch Automation modules are hosted within the KPI VS framework, as they are heavily dependent on
the collected KPIs and having them integrated into KPI VS, where the KPI collections exist, lowers transmission times for KPIs towards ML/ZTA modules.

Therefore, for category A, CDSO acts as service lifecycle orchestrator, being responsible for the service initiation, updates during lifetime and termination. In this case, CDSO only interacts with the one underlying orchestrator (most often ICT-17 platform orchestrator), which takes on the resource orchestration and VNF management responsibilities and presents towards CDSO an API for service lifecycle management. At this time UCs in LL4 and LL2 fall under this category, where CDSO interfaced with 5G-VINNI Patras and 5G EVE for this purpose.

However, even in this case, CDSO can affect the workflow and setting parameters of the service at runtime, by accessing the ZTA functionality from the KPI VS, as explained in Section 6 below. This is done mainly for optimization reasons, for example when quality of service KPIs or network KPIs indicate that the quality falls under certain SLAs. This feature is currently being trialed in the context of LL4 and will be extended to LL5 and possibly LL1.

For UCs that fall under category B, CDSO acts as cross-domain orchestrator, interacting with both the vertical and the ICT-17 platform orchestrators domains and ensuring that the overall workflow and service orchestration takes place according to the UC service specifications. This is now under investigation in the context of LL3, where in UC3.4 the plan is to create 2 concurrent RAN slices in the private node and one slice for video analysis in the vertical – these will be orchestrated across the two domains by CDSO. In future experimentation cycles, more complex situations involving cross-domain orchestration will be investigated in the context of LL5 and possibly LL1.

In the most general case, CDSO will interface with two orchestrators (using the APIs explained in Section 3) and ensure the correct workflow for the service is implemented. However, investigations are ongoing into the APIs that are available towards CDSO. If there is a lack of availability in this area, one possible solution is also to investigate the option of having the platform orchestrator manage the orchestration in the edge domain as well, and using CDSO as a service lifecycle orchestrator and / or for optimisations through ZTA, similar to the category A cases.

For category C, the orchestration is manually performed and no integration with the CDSO takes place. The number of UCs that consider this option is small and they use a private 5G note integrated into their factory environment. However, ongoing workshops between CDSO and UC owners in this category are in place, to discuss all options and whether moving towards a category B environment makes sense.

Additionally, an interesting direction of investigation that will take more shape in Cycle 2 and Cycle 3 is related to better understanding how a variety of underlying setups influences the orchestration options, taking into account also business roles. This mainly comes out from the LL1 & LL3 UCs supported by 5G-VINNI Norway. For example, some of the options are:

- Deployment using the public 5G system, with support for Private Logical Network (slice) as a Service (to the extent that the facility is able to support such LNaaS), with or without edge cloud for local break-out (i.e., UPF in edge). This is planned for all LL1 & LL3 5G-VINNI Norway supported UCs.
• Deployment using local 5G private network, with a full 5G system deployed in Edge Cloud. This is planned mainly for UC1.3, with the possibility to extending to UCs 3.5 and 3.6.
• Edge Cloud for 3rd party 5G Network enabled virtualized Application Functions (VNF/CNF/NAF).

The results coming out of these investigations will be included in orchestration and use case specific deliverables (D2.2D and D2.4B).

5 Challenges of Virtualisation and Orchestration in 5G-SOLUTIONS

In working towards providing an orchestration solution, it became quickly clear that the 5G-SOLUTIONS orchestrator must accommodate the diversity in LL services, including needs and goals, as well as deployment options for the LL service, and also the diversity in the orchestration solutions in the underlying platforms. This was addressed through a collaborative effort between (a) UC owners, (b) underlying platforms, and (c) the 5G-SOLUTIONS CDSO to bridge the gap between application-level service description and Network Slice / Network Service Descriptor definition. In order to advance in this important area for integrating 5G-SOLUTIONS with 5G-EVE and 5G-VINNI platforms, Living Labs sessions have been organised with the involvement of all stakeholders (LL owners, UC owners, underlying platform owners, CDSO owners). In this section, we present some of the challenges to virtualisation and orchestration, from the different points of view of the different stakeholders.

For the underlying platforms (5-VINNI Norway, 5G-VINNI Patras, 5G EVE, Amarisoft 5G Private Node), these challenges included (1) understanding how to support different verticals in platforms that were built for the main purpose of 5G experimentation and KPI validation and (2) supporting the 5G-SOLUTIONS use cases while still developing their platforms. Even at present, some of the underlying platform orchestration solutions are under development. For example, 5G-VINNI Norway is having delays in their core deployment and the attached orchestration solution, and most probably will report KPIs at the end of an experiment, which is not supportive of Zero-Touch Automation features and means that we must only rely on vertical KPIs for this feature. 5G-VINNI Patras is still working towards supporting multi-slice deployments, which will happen in Cycle 2 and Cycle 3 of the 5G-SOLUTIONS experimentation. For 5G-EVE several issues have been brought up during the collaboration with 5G-SOLUTIONS, such as ease of troubleshooting when a vertical deployment was not working according to the plan, timestamp misalignment between various systems in the 5G EVE facility (interworking layer, orchestration, KPI Analysis System, web portal), as well as the inability to report KPIs during the experiment. For the 5G Amarisoft private node, a specific challenge we had was the fact that the node does not offer the possibility of virtualization in the core part, but only in the RAN.

From the CDSO point of view, the challenges consisted in (1) dealing with the variety in various platform orchestrators, as well as the variety in working individually with each use case to define the orchestration flows depending on the needs of the use case and (2) limited access and visibility to underlying platforms orchestration. For (1), different plugins have been defined for each of the use cases that have been integrated with the CDSO; this is well described in deliverable D2.2C. In terms of the interaction with the underlying platforms, the integration with 5G-VINNI Patras has been achieved by accessing OpenSlice through the TMF633 Service Catalogue Management API [18] and TMF641 Service Ordering [19] APIs. The integration with 5G-VINNI Oslo is now under discussion, using a similar approach of using the same
OpenSlice TMF APIs as the ones exposed in 5G-VINNI Patras. For 5G-EVE, the integration has been achieved with 5G-EVE only allowing CDSO to request the onboarding of the experiment and the execution of it (i.e., start & stop).

Challenge (2) refers to the fact that CDSO has no control over the basic orchestration actions of the actual slices and VNFs deployed in the underlying platform for the service (including which resources it will use, where they are located and how they interconnect). These orchestration actions and lifecycle actions are not exposed by the local facility orchestrator; therefore, an external orchestrator has limited options in terms of the actions it can take. Usually, the API exposed by the facility orchestrator includes the ability to order and configure a service, activate and terminate a service and monitor the service status. These APIs are standard APIs.

This became very important especially when investigating the option of implementing Zero-Touch Automation features alongside the orchestrator, where the options of affecting the actual orchestration in the platform are limited. The Zero-Touch Automation feature is also affected by the capability of the underlying platform to report KPIs throughout the duration of the experiment, rather than only at the end of the experiment. This is discussed in more detail in Section 5.

Additionally, CDSO is an inter-domain orchestrator, without direct access to configuration and deployment policies within the platform. This might affect performance, especially in the case where the service runs under tight time constraints.

The Living Labs met different challenges (1) depending on the goals and needs of each UC in the LL and also (2) depending on the specific characteristics of the underlying platform and what features the platform has decided to offer towards the vertical. Also, it is notable that the different LLs have different maturity levels in terms of orchestration. While LL sessions on orchestration have been held to bridge the gap between the orchestrator and the UC needs and goals, some use cases are more advanced in this area because their focus has been on orchestration from the beginning. In this category, we can list use cases UC3.4 and UC 4.4, UC4.6. Other use cases have been focused more on 5G capabilities since their primary goals were more related on making sure that their service relies on 5G transmission rates. Here we can list use cases from LL1 that rely on in-factory deployments, where the effort was put primarily into ensuring that the communication infrastructure and capabilities are in place. For these use cases, the effort to redesign the service using virtualization was not justifiable unless the performance of the deployment and communication aspects were up to the standard needed by the use case. These cases are now in the process of planning the introduction of virtualization and they draw from the experience of other 5G-SOLUTIONS use cases that have done it before them. This is expected most probably in Cycle 3 of the 5G-SOLUTIONS experimentation.

In terms of Living Labs’ challenges related to the underlying platforms features, the challenges listed above under underlying platform challenges (i.e., delays and flexibility) affected also the service orchestration options for each LL / UC.

More specifically, if we look per LL, LL1 reported challenges in the areas of integrating the concept of virtualisation into their existing service, especially when the primary focus was on the performance of the communication, as well as delays coming from 5G-VINNI Norway for UC1.3 and UC1.5. However, looking forwards there are plans to introduce virtualization and speed up this process by building on the
experience that we have had with other UCs in this area. LL1 is also considering investigating which of
these virtual functions can then be offered at the edge across multiple industries. The results in
the virtualization and orchestration direction for LL1 will be included in detail in deliverable D2.2D.

LL2 reported challenges mostly in terms of deploying and troubleshooting the deployment on 5G EVE.
Issues such as limitations on CDSO (CDSO not allowed to order the deployment of the experiment, but
only allowed to execute it), or timestamp misalignment between various systems in the 5G EVE facility
(interworking layer, orchestration, KPI Analysis System, web portal), as well as difficult troubleshooting
due to non-availability of logs were reported. Additionally, the interaction between the RunTime
Configurator and the Interworking Layer Function Repository caused various issues regarding the
collection of the Network KPIs from 5G EVE, which would impact any monitoring and automation features
that are highly dependent on the availability of data.

LL3 UCs span across two 5G facilities used in 5G SOLUTIONS, namely the 5G VINNI Norway facility and the
private node 5G facility (Amarisoft) in Dublin. Therefore, the orchestration differs depending on the
facility that each UC is using. Furthermore, orchestration requirements depend on the advancement of
each UC in Cycle 1.

The 5G VINNI Norway facility uses TaaS (Test as a Service), which is used to run test cases and collect
network and application KPIs. The TaaS integration with the CDSO is planned and will happen as part of
Cycle 2, and therefore all UCs that use the 5G VINNI Norway facility (UC3.1-3 & UC3.5-6) will use CDSO
orchestration in the subsequent cycles. For example, UC3.1 and 3.2 have started developing the vertical
applications code which will later be packaged as third party VNFs and will need to be orchestrated by the
CDSO.

The UC that uses the private 5G node (UC3.4) is also working towards a full integration with the CDSO. As
explained in Section 3.4, the private 5G node supports both the 5G Core and RAN functionalities, but only
exposes a WebSocket API for RAN orchestration, while the 5G Core consists of a closed source binary code
used as a monolithic deployment in the 5G base station. This creates a big problem in terms of the
orchestration and the integration with the CDSO, as no orchestration of slices is available at this level and
no custom configuration APIs are exposed by the 5G Core. VNF orchestration is especially needed in UC3.4
as this UC explores two scenarios with two different network slices (mMTC and eMBB), and both may be
deployed concurrently, and also as part of the MLL (Multi-Living Lab). Work is being carried out now to
orchestrate the application VNFs (centered around IoT and video analytics) using Open Source MANO
(OSM). However, the orchestration and configuration of the RAN part in this facility is not supported by
OSM. This is where the CDSO comes into play, as this will allow for an end-to-end orchestration spanning
the RAN and application VNFs all from a central location. The integration with the CDSO will be enabled
by the built-in WebSocket API for the RAN and the northbound API of OSM for the Core and third party
VNFs.

LL4 use cases relied on a close collaboration with 5G-VINNI Patras and CDSO for orchestration. Most
challenges related to understanding all the components functionality and APIs have been addressed
through ongoing workshops. Additional challenged have been met in terms of virtualization, where for
example LiveU (UCs 4.3, 4.4, 4.6) uses many operators and it is impractical to install the gateway in all of
these operators. Other challenges reported were in the area of complexity of integrations, which had an
impact on the readiness of the UCs in the 5G-SOLUTIONS experimentation cycles. Also, for many short
tests, the overhead incurred by the setup and tear down of all the components was non-negligible.
LL4 is also the first LL to trial Zero-Touch Automation in the framework of UC4.4 and these results will be included in D2.2D.

**LL5** focuses on the definition and trials of Multi-LL use cases. In this case, when multiple UCs run concurrently, orchestration becomes a critical feature. CDSO enables the MLL trials to run smoothly and effectively, by automating the orchestration for the multiple LLs across the facilities. As each facility has its own orchestrator, it is fundamental to integrate CDSO with the facility orchestrator before any MLL trials run. The integration process is challenging as CDSO and the facility orchestrator need to agree on the connectivity, the interfaces, the capabilities to be exposed, and whether/how/what the requests from CDSO can be executed by the facility orchestration system. Additional plug-ins may be developed by CDSO to facilitate the demands of integration.

Similar to the single UC trials, CDSO needs to coordinate with all UC partners who participate in the MLL scenarios and decide the MLL workflow, based on the individual workflows produced by CDSO and each UC. If one or more use cases contain 3rd party VNFs, CDSO takes responsibility for managing the 3rd party VNFs and their lifecycles. To what extent CDSO can manage the 3rd party VNFs depends on the split of responsibility between CDSO and individual facility orchestrator. This option is first investigated in the context of UC3.4 and will extend to UC5.1. The results will be presented in deliverable D2.2D.

Also, in the case of the Multi-LL UCs, as CDSO is integrated with TaaS (Test-as-a-Service) to manage the tests, it is important that CDSO retrieves the metadata of each test case and communicates with KPI-VS to differentiate the collected KPIs for each use case. This becomes important for the Zero-Touch Automation aspects of LL5.
6 Service Orchestration Optimisation Features

As part of this task, we looked at the possibility of introducing more automation into the system, through the area of optimizations that the inter-domain orchestrator CDSO can achieve. This extends the role of the CDSO towards a more active role, cooperating with the underlying local domain orchestrators for run time optimisations. Our approach is based on a Zero-Touch Automation closed loop that can achieve reconfigurations when the state of the service / system is degrading. To support this, KPI VS (including ML features and ZTA feature), CDSO and the local orchestrators must all collaborate. This is explained in detail in the next subsections.

6.1 Extensions Towards Introducing More Automation in the Orchestration Area

Automation has been a desired goal in telecom network management for many years, although in the network management domain it was traditionally slow to adopt. The reasons behind this included human operator acquired experience, which was not easily translated into automated processes, variability in network deployments and settings, and need of control over the network from the side of the operators. However, with the increasing complexity and dynamicity brought on by 5G, there is an increased need for automation and flexible management, as manual processes become very time consuming and even unfeasible. In 5G-SOLUTIONS, we are investigating the introduction of Machine Learning & rule-based approaches as a means to optimise the end-to-end orchestration of Use Cases in 5G-SOLUTIONS.

In general, in the area of automation there are two main approaches:

1) Rule-based approaches: these approaches aim to address well defined automation needs in the network by use of predefined rules and policies. These systems rely on domain expert knowledge that can be easily translated into machine-readable rules. Traditionally, telecom systems have relied on rule-based systems and therefore for specific areas, we can still capitalize on this knowledge, especially if we combine it with data-driven approaches. The upside to having a rule/policy driven automation is that the solutions are usually robust and well defined to address specific tasks. However, they won’t be able to scale with the variation of problems in the network or complexity of the network.

2) Data-driven approaches: the second approach is to use data-driven techniques such as Machine Learning (ML) to learn the behavior of networks using statistics and measurements that are continuously generated and collected at different stages of the lifecycle in the network / vertical domains. The upside of using ML approaches is that they can readily scale and accommodate the dynamic nature of the network, i.e. the automation rules and policies are not required to be tailored for specific issues in the network. Further, these solutions are dynamic, i.e. they are not required to be modified or amended as the network evolves or grows in complexity, since the methods will ensure that the new behavior is (re-)learnt over time. This only ensures that new problems that have not been examined before using previously implemented solutions can be addressed. The downside compared to rule-based approaches is that since these solutions are not hand-crafted for individual problems, they may not always be as robust as rule-based solutions that are designed by domain experts and this is shown through a measure of accuracy on how well they reflect and predict network behavior.
These approaches can also be combined to drive automation in the 5G system [20].

In 5G-SOLUTIONS, we provide lightweight containerized solutions for the following:

- ML-based solutions for KPI correlations, predictions of KPIs and time series forecasting.
- Zero-Touch Automation which is our rule-based control loop for enabling service optimisations / reconfigurations.

A combination of these methods can be used to enable service optimisations, for example using output from the ML-based time series forecasting service to trigger action recommendations towards the CDSO.

Several important considerations regarding Zero-Touch Automation are:

- **Location of the containers**: transferring large quantities of data between containers will have an impact on the timeliness of the results. Since ZTA should return results in almost real-time, to enable adaptation of the system to dynamic conditions and optimisations, we took the decision to deploy our ML-based containers within the KPI VS, where all collections of KPIs collected from the experiments are hosted.

- **Type of data available to ZTA**: since the various platforms have a closed solution in terms of orchestration and CDSO is only acting as an inter-domain orchestrator, data regarding network resources allocated to the different slices is not available to the CDSO or KPI VS. This means that our optimisations can influence service workflows and specific service parameters, but not directly the allocation of resources to slices in the different domains.

- **Availability of KPI data in real-time**: in order to be able to assess in real-time how the system/service is doing, we need real-time KPI reporting from the verticals and underlying platforms. This is not always available, depending on how the platform / vertical report their KPIs. For example, 5G-VINNI Patras reports KPIs in real-time throughout the duration of the experiment, whereas 5G EVE and 5G-VINNI Norway report KPIs through TaaS, which allows reporting only at the end of the experiment. This dictates which UCs will be eligible for optimisations based on ZTA and / or the type of optimisations that will be possible if only vertical data is considered as input.

6.2 Zero-Touch Automation closed loop control

ZTA was designed as a rule-based control loop, which continuously checks pre-defined rules and once a rule is evaluated to true, a predefined action is sent to CDSO which can then act on this recommendation and collaborate with the underlying platform and / or vertical orchestrator(s) to perform the reconfiguration.

As shown in Figure 8 below, KPI VS continuously collects KPIs from the (multi-domain) 5G testbed. These KPIs can be directly investigated by ZTA, or ZTA can evaluate the rules on input coming from our ML containers deployed as part of the KPI VS (e.g., the time series forecasting analytics container). The rules are defined in collaboration with the use case owners and the underlying orchestrator(s), which provide the domain knowledge that enables the reconfiguration / optimization. The ZTA loop is triggered by CDSO through the Start API and stopped by the CDSO through the Stop API. Once triggered, the rules defined in the ZTA’s configuration are being evaluated in a cyclic fashion (the time interval is also defined in the
configuration file for ZTA). When the conditions in the 5G testbed that are checked in ZTA evaluate to true, an action is sent to our inter-domain orchestrator CDSO. CDSO will now collaborate with the underlying orchestrator(s) to act on the recommendation, which will change the state of the 5G system. The recommended changes are optimisations which can be observed in the service KPIs.

Figure 8: Zero-Touch Automation closed loop and automation actors

Figure 9: Example of output from the ML forecast module – predicted values for lost_frames KPI measured at decoder 172.16.10.240 in UC4.1. Based on predicted values such as the one in this plot, the ZTA module can take a proactive decision and send the recommended action to CDSO.

One example that we are trialing at the moment is in the context of LL4, where predicted changes in data change or lost frames indicate a foreseen degradation in the video service. Figure 9 shows an example
from one of our trials, with the past values and forecast for the lost frames KPI; the green line represents the predicted KPI, and the grey area around it indicates the error margin. Predicted KPIs are monitored and evaluated by the ZTA controller, who will send a use case specific reconfiguration action to CDSO once these KPIs change by more than a certain predefined value. CDSO then will ask the 5G-VINNI Patras orchestrator to change service parameters related to the video encoder.

In order to inform and give better insights on vertical and platform KPIs, which might help in designing the rules to configure the ZTA module, additional ML modules included in KPI VS could be used. One example of this is the correlation module. Figure 10 shows an example from our LL4 trials, where vertical and network KPIs are correlated. This can be used to add additional rules for ZTA or for the case when certain KPIs are not reported regularly from one of the domains.

We also plan to extend the use of ZTA to use cases in LL5, where we expect ZTA to play an important role in the multi-LL orchestration case. Additionally, we will evaluate its use in the framework of LL1 in Cycle 3 of experimentation. These results will be reported in deliverables D2.3B and D2.2D, as well as D7.3B.

Annex 1 contains information on how to use the ZTA module, including endpoint usage examples and instructions and example on writing use case specific rules for ZTA.
7 Conclusions and Next Steps

This deliverable presents our insights at M30 of the project, on the orchestration challenges and approach to multi-domain service orchestration in 5G-SOLUTIONS.

The challenges were presented from the points of view of the different stakeholders, including 5G platform providers, vertical service providers and inter-domain orchestrator provider. Our approach to end-to-end service orchestration has at its core the Nokia CDSO orchestrator, who coordinates with the 5G platforms & vertical orchestrators for onboarding and configuring services, activating and deactivating them and polling for status updates. The resource orchestration is performed by the local orchestrators and it is not exposed to CDSO, therefore CDSO plays mainly the role of an inter-domain orchestrator. All local orchestrators are based on standard orchestration solutions (e.g., OSM) and use standardized (TMF-based) APIs to communicate with CDSO. In the case where the vertical is not fully integrated with the ICT-17 platform and does not have its own orchestrator, CDSO could orchestrate VNF/CNF functions in the edge domain. In the majority of use cases in the project up to this point this is not the case, but this remains an open direction of investigation at this point.

Automation is an important aspect towards having a solution that recognizes and reacts to the dynamic nature of a 5G service. Our Zero-Touch Automation solution represents a lightweight control loop that enables automatic reconfigurations with the goal of optimising service delivery and performance at runtime. Combined with our KPI forecasting services that are part of KPI Visualisation System, ZTA can become a powerful tool that allows CDSO to proactively optimize service delivery on the fly.

More information on specific implementations in the orchestration area will be covered by WP2 deliverables D2.2D and D2.3B.
Annex 1: Zero-Touch Automation API / usage

The ZTA controller is available at https://ml-control.kpis-5gsolutions.eu. All documentation is available at https://gitlab.kpis-5gsolutions.eu/kpi-visualisation-system/ml-optimisation/ml-control/-/blob/master/README.md. In this annex, we include information about the endpoints and examples of using this module.

The ZTA module offers the following endpoints to manage control loops:

- /start (POST)
- /stop/name (DELETE)
- /list (GET)

/start (POST)

The /start endpoint allows the user to set up a new ZTA control loop. It must be configured through a JSON configuration in the request body. The configuration must follow this minimal structure:

```json
{
    "name": str,
    "rule1": {
        "query": str,
        "index_path": list,
        "condition": {
            "type": int/list
        },
        "action": {
            "url": str
        }
    },
    "interval_length": int,
    "duration": int
}
```

The name field can contain any URL-encoded string, e.g. "control-policy-1". The configuration may contain one or more rules, where the name of the rule (e.g. "rule1") can be any URL-encoded string. Multiple rules are executed in random order. Each rule contains the following fields:

- query: A URL from where a value should be monitored
- index_path: A list of indices where the value can be found in the query response. For example, if the query returns ("forecasts": [0.5, ...]), the index_path ["forecasts", 0] will return the value 0.5.
- condition: The condition for triggering the control action. Multiple conditions can be combined into a conjunct condition. Currently, the following condition types are implemented:
- ">", "gt": Greater than comparison
- ">=", "gte": Greater than or equal to comparison
- "<", "lt": Lesser than comparison
- "<=", "lte": Lesser than or equal to comparison
- "==", "=": Equal to comparison
- "in": Contains operator

- **action**: The action to be performed if the condition is true. This field contains a `url` through which the monitored system can be configured.
  - `url`: The URL where the action is to be performed
  - `headers`: (optional) headers to be passed to the request
  - `body`: (optional) body to be passed to the request. This will transform the request into a POST request. The `interval_length` field defined how often the rules should be executed, i.e. every x seconds, and the `duration` field limits the runtime (in seconds) of the ZTA control loop.

In addition to the normal rules that are executed independently in random order, rules can also be combined into a logical "and" conjunction using the `and` field as demonstrated below. Any number of normal rules (individually executed) can be combined with any number of "and" rules. Note that all "and" rules share one common `action` field.

```json
{
  "name": str,
  "and": {
    "rule2": {
      "query": str,
      "index_path": list,
      "condition": {
        "type": int
      }
    },
    "rule3": {
      "query": str,
      "index_path": list,
      "condition": {
        "type": int
      }
    },
    "action": {
      "url": str
    }
  },
  "interval_length": int,
  "duration": int
}
```

**/stop/name (DELETE)**

The `/stop/name` endpoint will stop a running ZTA control loop before the next iteration. It will not interrupt an ongoing query or action. Control loops that were stopped or finished by themselves are automatically cleaned up. The name refers to the `name` field in the configuration that was used to set up the running ZTA control loop.
/list (GET)

The /list endpoint lists all currently running ZTA control loops and their metadata, including rules, total runtime and elapsed time.

Example configuration

An example configuration for a ZTA control loop in 5G-SOLUTIONS, querying the system to perform a forecast on a KPI and when some conditions are met on that KPI, actions will be taken on the CDSO. This control loop policy contains one normal rule ("rule1") and two rules that are conjunct with a logical "and" ("rule2" and "rule3"). The loop will run every 30 seconds for 10 minutes (600 seconds).

```json
{
    "name": "controlpolicy1",
    "rule1": {
        "query": "https://ml-prediction.kpivs-5gsolutions.eu/forecast/dl_bitrate?collection=LivingLab3UseCase4Network",
        "index_path": ["forecast", 0],
        "condition": {
            ">": -10,
            "<": 10
        },
        "action": {
            "url": "https://10.0.92.66:9443/api/cbnd/v1/networkservices/{{nsId}}/operations",
            "headers": {
                "Content-Type": "application/json",
                "Authorization": "Bearer {{access_token}}",
                "Refresh-Token": "{{refresh_token}}"
            },
            "body": {
                "operationType": "UPDATE",
                "parameters": ["param1": "new_value"]
            }
        }
    },
    "and": [
        "rule2": {
            "query": "https://ml-prediction.kpivs-5gsolutions.eu/forecast/dl_bitrate?collection=LivingLab3UseCase4Network",
            "index_path": ["forecast", 0],
            "condition": {
                ">": -10
            }
        },
        "rule3": {
            "query": "https://ml-prediction.kpivs-5gsolutions.eu/forecast/dl_bitrate?collection=LivingLab3UseCase4Network",
            "index_path": ["forecast", 0],
            "condition": {
                "<": 10
            }
        }
    ]
}
```
D1.2B Cross-domain orchestration and management challenges analysis (v2.0)

{
    "action": {
        "url": "https://10.0.92.66:9443/api/cbnd/v1/networkservices/{{nsId}}/operations",
        "headers": {
            "Content-Type": "application/json",
            "Authorization": "Bearer {{access_token}}",
            "Refresh-Token": "{{refresh_token}}"
        },
        "body": {
            "operationType": "UPDATE",
            "parameters": [{"param1": "new_value"}]
        }
    },
    "interval_length": 30,
    "duration": 600
}
Annex 2: KPIs reported by 5G-VINNI Patras towards KPI VS

Table 3: 5G-VINNI Patras monitored KPIs and respective details. Below shows the KPIs reported by 5G-VINNI Patras, including name, description, dimension it refers to and additional details. These KPIs can be used by the ZTA feature.

<table>
<thead>
<tr>
<th>Metric ID</th>
<th>Metric</th>
<th>Dimension</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>netdata_STATS_GNODEB_MHz_average</td>
<td>CPU USAGE</td>
<td>CPU usage for RX</td>
<td>gNodeB Wide</td>
</tr>
<tr>
<td>Sample Rate</td>
<td>CPU usage for TX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_STATS_GNODEB__of_Transport_blocks_average</td>
<td>Successful Transmission</td>
<td>uplink tx</td>
<td>Per Cell</td>
</tr>
<tr>
<td></td>
<td>Sample rate in RX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_STATS_GNODEB__of_Transport_blocks_average</td>
<td>Sample rate in TX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_STATS_GNODEB_bps_average</td>
<td>bitrate</td>
<td>downlink bitrate</td>
<td></td>
</tr>
<tr>
<td>netdata_STATS_GNODEB_bps_average</td>
<td>uplink bitrate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_STATS_GNODEB_db_average</td>
<td>SignalToNoise ratio</td>
<td>SNR</td>
<td>Per UE per Cell</td>
</tr>
<tr>
<td>netdata_STATS_GNODEB_db_average</td>
<td>uplink tx</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>downlink tx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_STATS_GNODEB_db_average</td>
<td>uplink retx</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>downlink retx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_UE_STATS_GNODEB__of_Transport_blocks_average</td>
<td>Successful Transmission</td>
<td>uplink tx</td>
<td></td>
</tr>
<tr>
<td></td>
<td>downlink tx</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>uplink retx</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>downlink retx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_UE_STATS_MME_B_average</td>
<td>total bytes</td>
<td>downlink total bytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>uplink total bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_UE_STATS_GNODEB__of_Transport_blocks_average</td>
<td>total bytes</td>
<td>downlink total bytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>uplink total bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_net_net_kilobits_persec_average</td>
<td>kbps/sec</td>
<td>Received</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_netdata_server_cpu_milliseconds_average</td>
<td>CPU usage</td>
<td>System</td>
<td>System Wide</td>
</tr>
<tr>
<td></td>
<td>Load1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>load5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>load15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_system_load_load_average</td>
<td>System Load</td>
<td>System load for last 1, 5 and 15 minutes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Load1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>load5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>load15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_system_net_kilobits_persec_average</td>
<td>Network</td>
<td>Received</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_system_Uptime_seconds_average</td>
<td>Uptime</td>
<td>Seconds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System wide</td>
<td></td>
<td></td>
</tr>
<tr>
<td>netdata_net_packets_packets_persec_average</td>
<td>Packets</td>
<td>multicast</td>
<td>Per host interface</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------</td>
<td>-----------</td>
<td>--------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Received</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmitted</td>
<td></td>
</tr>
<tr>
<td>netdata_sctp_packets_packets_persec_average</td>
<td>SCTP packets</td>
<td>Received</td>
<td>System Wide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmitted</td>
<td></td>
</tr>
</tbody>
</table>
References


