This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No 856691.

5G-SOLUTIONS for European Citizens

D2.3B Zero-touch Automation Mechanisms for 5G Service Lifecycle (v2.0)

Document Summary Information

<table>
<thead>
<tr>
<th>Grant Agreement No</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>
</tr>
<tr>
<td>Start Date</td>
<td>01/06/2019</td>
<td>Duration</td>
</tr>
<tr>
<td>Project URL</td>
<td><a href="https://www.5gsolutionsproject.eu/">https://www.5gsolutionsproject.eu/</a></td>
<td></td>
</tr>
<tr>
<td>Deliverable</td>
<td>D2.3B</td>
<td></td>
</tr>
<tr>
<td>Work Package</td>
<td>WP2</td>
<td></td>
</tr>
<tr>
<td>Contractual due date</td>
<td>31/05/2022</td>
<td>Actual submission date</td>
</tr>
<tr>
<td>Nature</td>
<td>Report</td>
<td>Dissemination Level</td>
</tr>
<tr>
<td>Lead Beneficiary</td>
<td>LMI</td>
<td></td>
</tr>
<tr>
<td>Responsible Author</td>
<td>Anne-Marie Bosneag (LMI)</td>
<td></td>
</tr>
<tr>
<td>Contributions from</td>
<td>Anne-Marie Bosneag (LMI), Shubham Jain (LMI), Bader Mawasi (Nokia), Udi Margolin (Nokia), Panagiotis Papaioannou (UOP), Dimitris Giannopoulos (UOP)</td>
<td></td>
</tr>
</tbody>
</table>
Revision history (including peer reviewing & quality control)

<table>
<thead>
<tr>
<th>Version</th>
<th>Issue Date</th>
<th>% Complete</th>
<th>Changes</th>
<th>Contributor(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>01/04/2022</td>
<td>5%</td>
<td>Initial Deliverable Structure and ToC</td>
<td>Anne-Marie Bosneag (LMI), Shubham Jain (LMI)</td>
</tr>
<tr>
<td>V0.1</td>
<td>15/04/2022</td>
<td>10%</td>
<td>Initial contributions to All Sections</td>
<td>Anne-Marie Bosneag (LMI), Shubham Jain (LMI)</td>
</tr>
<tr>
<td>V0.1</td>
<td>18/04/2022</td>
<td>15%</td>
<td>Quality Check</td>
<td>Christos Skoufis (EBOS)</td>
</tr>
<tr>
<td>V1.0</td>
<td>01/05/2022</td>
<td>60%</td>
<td>Content added to All Sections</td>
<td>Anne-Marie Bosneag (LMI)</td>
</tr>
<tr>
<td>V1.0</td>
<td>15/05/2022</td>
<td>80%</td>
<td>Content added to All Sections</td>
<td>Shubham Jain (LMI)</td>
</tr>
<tr>
<td>V1.0</td>
<td>18/05/2022</td>
<td>80%</td>
<td>Quality Check</td>
<td>Christos Skoufis (EBOS)</td>
</tr>
<tr>
<td>V1.1</td>
<td>30/05/2022</td>
<td>85%</td>
<td>Content added to All Sections</td>
<td>Anne-Marie Bosneag (LMI)</td>
</tr>
<tr>
<td>V1.2</td>
<td>07/06/2022</td>
<td>100%</td>
<td>Quality Check and Peer Review</td>
<td>Christos Skoufis (EBOS), Andrea Di Giglio (TIM), Kostis Tzanettis (AppART)</td>
</tr>
<tr>
<td>V1.3</td>
<td>28/01/2020</td>
<td>100%</td>
<td>Final Version</td>
<td>Anne-Marie Bosneag (LMI)</td>
</tr>
</tbody>
</table>

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.

1 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
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 ............................................................................................................................................... 8

2 Introduction ......................................................................................................................................................... 9

2.1 Mapping to Project Outputs .......................................................................................................................... 9

2.2 Deliverable Overview and Structure ............................................................................................................. 10

3 Automation and 5G Service Orchestration in 5G-SOLUTIONS ........................................................................ 12

4 Zero-Touch Automation closed loop in 5G-SOLUTIONS .................................................................................. 15

4.1 ZTA Rule-based controller ............................................................................................................................ 16

4.2 Machine Learning for enabling good coverage of ZTA rules ....................................................................... 16

5 ZTA applications in 5G-SOLUTIONS .................................................................................................................. 19

5.1 ZTA in UC4.1 .................................................................................................................................................. 19

5.2 ZTA in UC4.4 .................................................................................................................................................. 25

5.3 ZTA in UC5.3 .................................................................................................................................................. 29

5.4 ZTA in LL1 .................................................................................................................................................... 30

6 Conclusions ....................................................................................................................................................... 33

References .............................................................................................................................................................. 34
List of Figures

Figure 3-1: High-level view of components involved in 5G-SOLUTIONS automation ....................................13
Figure 4-1: Closed Loop Automation in 5G-SOLUTIONS ...........................................................15
Figure 4-2: Standardised residual & correlogram plots for ARIMA (in UC4.1) .......................................18
Figure 5-1: Rule 1 defined for ZTA in UC4.1 ..............................................................................21
Figure 5-2: Rule 2 defined for ZTA in UC4.1 ..............................................................................22
Figure 5-3: Rule 3 defined for ZTA in UC4.1 ..............................................................................23
Figure 5-4: Downlink bitrate KPI forecast ....................................................................................24
Figure 5-5: CDSO workflow includes ZTA (left) and action events received from CDSO (right) ..........24
Figure 5-6: Total Rate changes as a result of ZTA actions that are triggered by changes in Downlink Bitrate ..................................................................................................................25
Figure 5-7: The quality of the service (shown by Lost Frames) stays within the required parameters even with a sharp increase in the traffic (Total Bytes) .................................................................25
Figure 5-8: Rules defined for ZTA in UC4.4 ...............................................................................27
Figure 5-9: Changing the gNodeB configuration to favour UL .....................................................28
Figure 5-10: Results of two separate trials for ZTA in UC4.4, which show that ZTA gets triggered when uplink_bitrate KPI falls under 5Mbps, resulting in gNodeB reconfiguration and therefore a higher total_bytes KPI for the service ..................................................................................................29
Figure 5-11: CDSO UI shows ZTA update action was received by CDSO ...........................................29
Figure 5-12: UC1.5 architecture, including ZTA for optimisations .................................................31
Figure 5-13: End-to-end orchestration across Core and Edge domains in 5G-VINNI Norway ..........31
List of Tables

Table 2-1: Adherence to 5G-SOLUTIONS GA Deliverable & Tasks Descriptions ................................................. 9
## Glossary of Acronyms

<table>
<thead>
<tr>
<th>Abbreviation / Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>CDSO</td>
<td>Cross-Domain Network Orchestrator</td>
</tr>
<tr>
<td>E2E</td>
<td>End to End</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FCAPS</td>
<td>Fault, Configuration, Accounting, Performance, Security</td>
</tr>
<tr>
<td>5G</td>
<td>Fifth Generation</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MANO</td>
<td>Management and Network Orchestration</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SDN</td>
<td>Software Defined Network</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>ZSM</td>
<td>Zero Touch Network and Service Management</td>
</tr>
<tr>
<td>ZTA</td>
<td>Zero-Touch Automation</td>
</tr>
</tbody>
</table>
1 Executive summary

Automation has been a goal for telecommunication systems management for a long time, with a view of decreasing the manual tasks, incorporating as much operational knowledge into the system as possible and increasing the reaction speed to any problems that might appear in the network. In the context of 5G networks, the requirement for automation becomes even more stringent, due to the increase in dynamicity, complexity, and scalability, which makes real-time manual resolutions infeasible in certain situations.

In this deliverable, our focus is on describing the Zero-Touch Automation (ZTA) mechanisms that we provided in the scope of the 5G-SOLUTIONS project. 5G-SOLUTIONS is a H2020 project aiming at enabling the deployment of various verticals on open 5G experimentation platforms such as 5G-VINNI and 5G EVE, as well as the Amarisoft private 5G node. A major focus of the project is on multi-domain orchestration, with a focus on optimizing service delivery and lifecycle management. In this context, ZTA can be used as needed to allow for automated optimisations during the run time of the services.

This deliverable presents the challenges and design choices made, including issues such as availability of real time monitoring data during the experiments, APIs from local orchestrators, types of changes that can be enacted, etc. Our ZTA framework is generic, lightweight, and scalable, being implemented as a service that can be adapted to the specifics of each use case and platform that we work with. This enables the creation of various ZTA mechanisms to cover the wide range of deployments and types of 5G services in our project. The ZTA solution is rule-based to allow for the incorporation of service-specific knowledge and can be combined with Machine Learning (ML) methods to enable more complex behaviours. For example, when combined with forecasting methods, we can enable proactive automations, rather than reactive.

After presenting the framework, we present two end-to-end trials that we performed with ZTA. They are different in type of activation (proactive vs. reactive), as well as in terms of the action taken (changes in the service or network configurations). These trials successfully showed the validity of our approach and paved the way for two more proposals of applying ZTA, one in the context of MLL and one in the context of 5G-VINNI. These approaches have been transferred to their respective work packages (WP7 & WP4, respectively) and the investigations and results will be reported in their respective deliverables for cycle 3.

The key achievements covered by this deliverable are:

- Exploration of challenges and design choices for automation mechanisms in 5G-SOLUTIONS;
- Presenting our generic lightweight and scalable Zero-Touch Automation solution developed in the context of 5G-SOLUTIONS. This ZTA approach can be used in a variety of 5G deployments and services;
- ZTA trials for (1) proactive scenario with changes to service configuration, and (2) reactive scenario with changes to network configuration;
- Proposals for further exploring the application of ZTA to the Multi-LL scenario, as well as working with another 5G platform (5G-VINNI) than the existing trials.
2 Introduction
Automation in 5G telecom systems is a feature that is much needed and researched. The first version of this deliverable, D2.3A, presented the large 5G orchestration and automation landscape, as well as specific challenges that we envisioned at that early stage in the project. This deliverable is the final version, presenting the Zero-Touch Automation solution that we designed, implemented and tested in 5G-SOLUTIONS.

5G-SOLUTIONS is focused on the deployment and running of various 5G verticals, from Factories of the Future to Smart Cities, Smart Energy, and Multimedia, on top of different open 5G platforms that come with their own orchestration and KPI reporting solutions. To enable various types of automations in such a vastly heterogeneous environment, we designed the ZTA solution as a generic rule-based framework, that can be combined with various Machine Learning models to enable both proactive and reactive types of optimisations.

This deliverable presents:
- design considerations for ZTA
- closed loop for automation, including all components from KPI VS, to CDSO, to ZTA, to 5G platform and 5G vertical
- implementation of rule-based ZTA framework, interfaces, usage with or without using ML models in combination with the rules that encode service-specific knowledge
- results of our successful end-to-end trials for both proactive and reactive optimisations, affecting both vertical and 5G platform configurations
- proposals for further inclusion of ZTA trials in cycle 3, in the context of MLL (WP7) and LL1 (WP4).

As our Zero-Touch Automation solution is generic, lightweight and scalable, we envision this component to be used in various settings in future projects.

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 2-1: 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>Task 2.4: Zero-touch automation mechanisms for 5G service lifecycle (v2.0)</td>
<td>Within the context of programs, such as ETSI Zero-touch Network and Service Management program or MEF Lifecycle Services Orchestration, there has been much industry discussion on automation and lifecycle concerns. Zero-touch automation</td>
<td>Sections 3-5</td>
<td>Section 3 discusses challenges that influenced our design of the Zero-Touch Automation solution, Section 4 presents our closed-loop</td>
</tr>
</tbody>
</table>
5G service lifecycle mechanisms are inherently linked to the increase of speed and removal of human interaction throughout the lifecycle of the processes, with a focus on pro-active management. In this task, we will examine the complex problem of managing services and machine learning features in deployed operations.

**Analysis of emerging Machine Learning lifecycle programs:** We will investigate initiatives on Machine Learning lifecycle programs, with a key focus on the viability of a DevOps environment that encompasses machine learning models in production systems.

**Enablement of Machine Learning and Automation:** In this subtask, we will explore novel vertical industry automations, based on adaptive automation of 5G features combined with predictive analyses of data coming from the vertical industry service. We will also explore network application tracking to support the next generation of Root Cause Analysis features. By this we extend the core value of automation and accountability to be fundamental parts of dependable automation.

<table>
<thead>
<tr>
<th>DELIVERABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D2.3B: Zero-touch automation mechanisms for 5G service lifecycle v2.0</strong></td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

### 2.2 Deliverable Overview and Structure

This deliverable is organized as follows:

- Section 3 presents a discussion on automation and 5G service orchestration in 5G-SOLUTIONS, discussing specific challenges, design decisions and high-level architecture including all components involved in zero-touch automations;
- Section 4 presents our closed-loop automation solution, ZTA rule-based framework implementation details, as well as the inclusion of ML into the zero-touch automations;
• Section Error! Reference source not found. presents our ZTA trials, including proactive and reactive adaptations of service and network configurations, as well as proposals for further applying ZTA to additional use cases in the project;
• Section 6 presents our conclusions.
3  Automation and 5G Service Orchestration in 5G-SOLUTIONS

Automation has been identified as a goal in many telecom management areas, and this need has become even more stringent in the context of 5G, with its added layers of abstraction, virtualisation and softwarisation introducing more complexity, as well as the need for a flexible management system that can adapt to various service needs. This need for automation has been addressed by various industry and standardization bodies, of which we most notably mention ETSI ZSM [1] that introduces both closed loop automation and Machine Learning elements at various levels in the 5G system. A detailed presentation of such industry and standardization bodies has been included in D2.3A [2] and we refer the reader to that deliverable for a more comprehensive view. We will note here, however, that it became apparent in the industry that, in order to support automation, we need both precise service-specific knowledge, as well as the ability to learn what is happening in the system during the runtime of a service, as context matters to decide what automation to run.

5G-SOLUTIONS positions itself as a H2020 project focused on running multiple various 5G verticals (ranging from Media applications to Factory-of-the-Future, to Smart Cities and Smart Energy applications) on top of 5G open platforms (5G-VINNI UoP, 5G-VINNI Norway, 5G EVE) [3]. As such, the cross-domain orchestrator, deployed as Nokia’s CDSO, plays a major role in the proper running of these verticals, being responsible for lifecycle management of the services, as well as any optimisations to be done on top of the 5G platform.

Another very important component of our system is the KPI VS [4], responsible for collecting KPIs in real time during the experiments, enabling statistics and ML for these KPIs and visualising them to the users.

Alongside these two main components developed and deployed in 5G-SOLUTIONS, we have designed a Zero-Touch Automation feature, that can work with the CDSO to enable automatic optimisations for the various types of services.

In designing the ZTA feature, we had to take into account particularities of the 5G-SOLUTIONS solution. Specifically, we took into account the following:

1. **KPI data availability in real-time:** we have investigated the reporting both from the vertical service and the 5G platform, as only platforms and services reporting KPIs in real-time during the execution of the experiment can take advantage of the ZTA feature. As such, 5G-VINNI UoP reports KPIs regularly in real time during service running, while 5G-VINNI Norway and 5G EVE are based on TaaS, which reports KPIs at the end of testing campaigns.

2. **Openness of underlying 5G platform and vertical to different types of optimisations:** for automatic actions to be enabled, APIs that will allow the automatic optimisations in the 5G platform or service are needed. Note that while some optimisations are already performed at the level of the local domain (e.g., local 5G orchestrator), we are looking into optimisations that can be performed at the cross-domain level, taking into account information from the service and platform. For this to happen, the CDSO needs to have a way to automatically trigger changes, i.e., to change different types of parameters related to slices and configurations. For example, we will show in later sections that in our testing of ZTA for UC4.1 (real time media streaming), the ZTA module could recommend changes to the video encoder rate used by the real-time video streaming service to allow for SLA maintenance; while in the case of ZTA for UC4.4 (high-quality content distribution), ZTA was allowed to trigger the change of the gNodeB configuration in the
5G platform. All of these changes required the implementation of the underlying mechanism, as well as the definition of an API towards the CDSO.

3. **Generic framework that can be easily integrated into various 5G systems**: we developed our ZTA solution as a generic framework that provides seamless integration into various types of systems, through a container-based lightweight implementation. This enables re-use of our solution in a multitude of 5G environments / levels, also providing ease of scalability. We note here that while we have applied ZTA to the cross-domain level, the framework can be re-used at different levels of the system in an ETSI ZSM fashion.

4. **Location of the ZTA container**: one important consideration for our design was the location of the ZTA container vis-à-vis the data that it needs, as a zero-touch optimization loop needs to constantly check the status of the system (via its reported KPIs) in order to decide when to trigger certain automations needed for the optimization of the service. It is a well-known fact in microservice design that the amount of data that needs to be transferred between microservices is one of the major performance bottlenecks. Since ZTA should return results in almost real-time to enable adaptation of the system to dynamic conditions, the timeliness of the results is a major factor. Therefore, we took the decision to deploy our ZTA container close to the KPI VS, which stores the real-time reported KPIs, as well as the ML-based containers, as reported in D3.2C [4]. Therefore, the ZTA is deployed on the same infrastructure as the KPI VS.

Figure 3-1 presents a high-level view of the various components involved in our solution:

**Figure 3-1: High-level view of components involved in 5G-SOLUTIONS automation**

As seen above, CDSO sits between the vertical service and the 5G platforms, enabling service orchestration across these domains. Therefore, CDSO collaborates with both domains and their orchestration solution for the definition of workflows that enable the particular service. One particular challenge for the CDSO is dealing with a variety of underlying orchestration solutions and vertical needs. Specific workflows and plugins have been developed to enable the service orchestration for the various verticals, as well as the proper integration with the different domains. These are explained in detail in
D2.2 [5] deliverables. The workflows also include steps for the ZTA start/stop, where we deploy ZTA as part of the scenario.

The KPI Visualisation System (KPI VS) interfaces with the 5G platform and vertical domains for collection of experiment KPIs via REST APIs. The data is collected with a high granularity (where possible) and stored in a MongoDB database, from where ML and ZTA containers can access them. The metrics produced by the monitoring service are displayed through the KPI VS user interface. The KPI VS also includes the ML containers, which allow users to get more insights into the experiment data and also enable more support for ZTA. These ML containers can be used in the closed loop automation for enabling more complex types of automation (e.g., proactive vs reactive), as well as for enabling a good coverage for the conditions to be checked. This is explained in more detail in the following section.

The ZTA container was developed alongside KPI VS, as explained above. This was developed as a rule-based controller. The rules encode service- and platform-specific knowledge to enable various types of optimisations, as needed by the particular service / platform deployment. This flexible and generic implementation enables a wide-range of optimisations, encoding specific knowledge but also working in collaboration with the ML containers where learning of the system behaviour is needed.

The next section delves into more detail into the ZTA design and implementation, as well as the closed loop containing all of the components mentioned in Figure 3-1.
4 Zero-Touch Automation closed loop in 5G-SOLUTIONS

In the context of 5G-SOLUTIONS, we implemented zero-touch automation at the level of cross-domain / service orchestration. In particular, we focused on optimization of service performance, based on training data sets that combine network counters and network-based KPIs with vertical-specific KPIs. This combines data from two domains that normally have no visibility to each other’s inter-workings, allowing for full-stack performance optimisations (within the limitations discussed in the previous section).

The closed loop automation, shown in Figure 4-1, works as follows: KPI VS continuously collects KPIs from the multi-domain 5G testbed, including the service and the 5G platform KPIs. ZTA is developed as a set of service-specific rules, which are continuously evaluated against the KPIs reported in real-time or against knowledge collected through the ML containers. The ZTA container ingests the KPIs through a data handler container (also used by the ML container), which was developed to take into account the particularities of KPI reporting (e.g., to ingest the time-series data while properly differentiating between the same KPI name reported by several components, etc.)

The ZTA loop is triggered and stopped by CDSO through a Start/Stop APIs. Once triggered, the rules defined are being continuously evaluated, at a specific pre-defined time interval. When the conditions evaluate to true, an action that was pre-defined and is specific for the type of optimization that we want to enable is sent to the cross-domain orchestrator CDSO. CDSO will then collaborate with the underlying domains to act on the recommendation and enable the changes in their respective domains. The recommended changes are optimisations that can be observed in the service KPIs.

Figure 4-1: Closed Loop Automation in 5G-SOLUTIONS
4.1 ZTA Rule-based controller

The ZTA controller provides a configurable way for deploying control loops that can retrieve data from the data handler / ML models, make a decision based on a condition specified and recommend actions to the CDSO, all using configurable REST calls. The control loop initiates with a START API that is configured through a JSON configuration in the request body. The configuration may contain one or more rules, where the name of the rule can be any URL-encoded string. Multiple rules are executed in random order where each rule contains the following fields:

- **query**: A URL for a container that provides a value to be monitored.
- **indexPath**: A list of indices where the monitored value can be found in the query response. For example, index_path = $n$ denotes that we look $n$ steps ahead in a forecast response.
- **condition**: The condition for triggering the control action. Multiple conditions can be combined into a conjunct condition (through “and” / “or” statements).
- **action**: The action to be performed if the condition is true. The action contains:
  1. **url**: The URL where the action is to be performed;
  2. **headers**: (optional) headers to be passed to the request;
  3. **body**: (optional) body to be passed to the request. This will transform the request into a POST request.
- **intervalLength**: defines how often the rules should be checked, in seconds.
- **duration**: defines the lifetime (in seconds) of the ZTA control loop, after which this loop is deleted.

Any number of such rules can be combined by “and” / “or” statements to cover more complex situations.


Additionally, more flexibility is also enabled by the use of KPIs straight from the KPI VS MongoDB, or by the use of the models provided by the ML included in KPI VS.

4.2 Machine Learning for enabling good coverage of ZTA rules

As explained in Section 3, the move towards ZTA in 5G is prompted by a variety of factors, such as:

- Increased technological complexity (accumulation of technologies, increasingly heterogeneous networks, new levels of abstraction/complexity, etc.);
- Dynamic behavior of the network prompted by dynamic demand for new functionality and capacity extensions;
- Variability and volume of network access (e.g., type and number of connections, diverse types of services and traffic patterns);
- Enablement of holistic management taking into account both network and service-level KPIs.

In this dynamic environment, *Machine Learning* is one important paradigm. While rule-based systems are very useful to enable the encoding of specific knowledge targeted towards certain aspects of the system, they are not very good at adapting to dynamic situations. This is where Machine Learning methods come
into play, and therefore a mixed approach using rules combined with ML models proves to be a very efficient approach in implementing zero-touch automations.

While understanding the benefits of using ML, we must also take into account the limitations, such as:

- **Availability of data needed to create accurate models**: the amount and quality of data available to the ML system, as well as the availability of labelled data, is of prime importance to ensure the accuracy of the models.
- **Lifecycle management of ML models**: the ML models must be updated in time, to reflect the new variations in data, changes in configurations, accessibility patterns, etc.
- **Understanding of results**: the results must be correctly evaluated in the context given by the system, including the error margins introduced by ML. A major impediment in the wide-scale adoption of ML-based automations in telecom management has also been the difficulty in understanding the solution provided by an automated learning system (e.g., where correlations learnt by ML were not immediately identifiable in the recommended solutions, or in the context of deep learning or reinforcement learning).

This means that work must be done to provide the right environment for inserting ML-based automation into the systems, including:

- A proper data layer;
- Correct ML tools for the data & features needed;
- Correct identification of automation use cases;
- Ability to understand and design ML algorithms that fit the data available and the domain problem, coupled with ability to understand where rule-based systems might be more beneficial;
- Correct management of ML models / use of updated models.

For our solution, we have worked in collaboration with KPI VS for understanding the data and how to properly process it, we have worked with CDSO, 5G platform and verticals for the identification of zero-touch automation use cases, and we have identified which ML methods are of use in the automation use cases identified.

To complement the ZTA rules, we have used ML algorithms to enable good coverage for the ZTA rules both in terms of:

- **action**: combining rules with time-series forecasting allows for proactive automations, because the rule can check if KPI trends suggest that allowed thresholds will be broken in the future. This allows the system to react before such trends happen. In the first example of applying ZTA, in Section 5, we will give an example of such behaviour.
- **KPI coverage**: ML can be used for getting better insights into the network & service KPIs and understand where we can for example, use a strongly correlated KPI that is already reported, instead of one that is difficult to be monitored. Specifically, if we detect a strong consistency across network & service KPIs, e.g., one service KPI is strongly correlated to a network KPI, and the monitoring & KPI calculation system of the network reports this network KPI in real-time to the KPI VS, we can check the network KPI rather than trying to implement additional service monitoring. Or a service KPI that is usually monitored and reported is highly correlated to a very complex network KPI where it is not feasible to continually calculate it; in this case, it makes sense
to check the service KPI. These correlations can be used in conjunction with the rules in ZTA to allow for good coverage of the rules.

The KPI VS includes ML containers for anomaly detection, KPI correlation, regression-based prediction, as well as time series forecasting. All of these containers are described in detail in deliverable D3.2C, including their implementation details and usage. We refer the reader to D3.2C [4] for all these details.

In this section we will focus on the time-series forecasting, as this was mainly used in our project in conjunction with ZTA to allow for proactive automated optimisations.

For our time-series prediction, we employed Auto-Regressive Integrated Moving Average (ARIMA), which relies on a combination of models that are fitted to time series data to better understand the data trends and predict future values based on a linear regression model. The model’s main parameters (Autoregression AR, Moving Average MA and Integrated (I)) heavily influence the results and vary significantly across different KPIs. For this reason, we decided to use auto-ARIMA based on the pmdarima Python package, retraining on new data every “n” number of seconds and calculating the optimal parameter combinations of (AR, I, MA) for a specific given KPI. We have tested this technique and noticed that it gave us very good results in our forecast of the downlink bitrate KPI for the 5G real-time media streaming service, as shown in Section 5.1. We further show this in Figure 4-2, by plotting residual plots. We can see that the residual errors have a constant variance around a mean of zero, showing good accuracy. The ACF plot (Correlogram) shows that the residual errors are not autocorrelated. Any autocorrelation would imply that the residual errors have a pattern that the model cannot explain. Additionally, we calculated the algorithm’s Symmetric mean absolute percentage error (SMAPE) denoted in the following equation, which calculates accuracy based on relative errors:

$$SMAPE = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{|A_t| + |F_t|/2}$$

![Figure 4-2: Standardised residual & correlogram plots for ARIMA (in UC4.1)](image)

In the next section, we also present the actual predicted values for our example of combining ML with ZTA rules to enable certain proactive automated behaviours.
5 ZTA applications in 5G-SOLUTIONS

While the previous sections explained the ZTA framework, the closed loop with all involved components, as well as inclusion of ML into the loop, this section presents the application of ZTA to different verticals in the project.

Specifically, in Sections 5.1 & 5.2, we present how we applied ZTA in the context of UC4.1 and UC4.4, respectively. These have been fully implemented closed loop automations, with the trials being performed end-to-end, involving all the components mentioned in Sections 3 and 4. The first example (Section 5.1) is an example of proactive optimisations involving vertical configuration changes, while the second example (Section 5.2) is an example of reactive optimisations involving network configuration changes. The results of the trials are included below in these sections.

Additionally, Sections 5.3 and 5.4 present our plans for further exploring ZTA in the context of Multi-Living Labs (WP7) and Factories of the Future Living Lab (WP4), respectively. The results of these investigations and associated trials will be reported in the deliverables associated with WP7 and WP4 for cycle 3.

5.1 ZTA in UC4.1

The first use case that we applied ZTA to in 5G-SOLUTIONS was UC4.1, which is a real time video streaming application. It focuses on the distribution of high quality Ultra High Definition (UHD) content over 5G networks. For this service to provide a satisfactory user experience, both latency and network consistency are considered key factors. The media application and equipment are deployed within the 5G-VINNI Patras testbed, similarly to a commercial deployment. Three encoders, able to stream content at multiple speeds, are used: two of them stream the signal, while the last one is used to multiplex the signals and create a demand for higher throughput. Furthermore, two decoders are also used: the first one is deployed in the core network, while the second one at the signal reception site. In this way, end-to-end application latency will be measured without including the impact of the public network. Our focus is on QoS of the video streaming, measured through KPIs such as Lost Frames and Downlink Bitrate.

The automation was implemented and tested end-to-end. This included the entire closed loop:

- CDSO starting the service, the KPI collection, as well as initiating ZTA with service-specific rules;  
- KPI VS collecting in real time the KPIs reported from the network and service;  
- ZTA starting the rule-checking and calling the forecast ML container during each iteration;  
- Action from ZTA being triggered and sent to CDSO;  
- CDSO sending the request to OpenSlice in UoP;  
- OpenSlice directing the request and change of the encoder’s rate.

The choices made, based on this service needs, are:

- **proactive adaptation**: we combine the rule-based approach of ZTA with ARIMA-based time series forecasting to enable looking ahead for trends in KPIs which indicate service degradation in the near future and react before the service degrades;  
- **changes in the service encoder rates**: our action will affect service equipment, specifically the encoders rate. The slicing or network configurations remain unaffected.

The rules that we defined check if the predicted value for the downlink bitrate KPI will go up by more than 10% in the next 50 seconds. If so, the CDSO is recommended to ask OpenSlice for a change up in the
encoder’s rate (where this rate can have the following values: 7.5 MBPS, 15.6 MBPS, or 26 MBPS).

Figure 5-1, Figure 5-2 and Figure 5-3 show the rules defined for ZTA in UC4.1. Here we see that we check the forecasted value (coming from the ML forecast container) 10 steps ahead (which corresponds to 50 sec) for downlink bitrate in the element identified by imsi-001010000045631. If this forecasted value is greater than 10% of current value, depending on the current rate, we calculate the new encoder rate, which will be sent to the CDSO. Because the action contains various new encoder rate, the policy was written as a set of “control policy1”, “control policy2”, “control policy3” to cover all possible encoder rates. The policy checks these rules every 10 seconds until the end of 7200 seconds, as given in interval_length & duration of the ZTA rules.

The control policies are sent together in one POST message to ml-control.kpivs-5gsolutions.eu/start.
Figure 5-1: Rule 1 defined for ZTA in UC4.1
Figure 5-2: Rule 2 defined for ZTA in UC4.1
Since the rule checking is based on forecast of the downlink bitrate KPI, we also present here the results from the ML forecast container. Figure 5-4 shows these results – the blue line indicates the actual KPI data, the orange line represents the forecasted values, where we used a 75:25 ratio of train to test data. The SMAPE score, as defined in the previous section is 91.2%, which shows a very good accuracy for the forecasted values. The green line represents the next forecasted values.

Figure 5-3: Rule 3 defined for ZTA in UC4.1
It should be noted here that although we got very good accuracy for our tests, the ML will always introduce a certain amount of uncertainty in the results, as well as maybe a lag in adjusting to new trends. This is unavoidable and it is a normal part of introducing ML into the system. In terms of working with the most updated model, we generate the ARIMA model every time the rule is checked, which works well for the size of data set that we have.

In terms of the action, ZTA will send back to the CDSO an update request including the new encoder rate. The CDSO is integrated with the underlying platform 5G-VINNI Patras by accessing OpenSlice through the TMF633 Service Catalogue Management API [6] and TMF641 Service Ordering APIs [7], and through the REST API for the ZTA flow. A use case specific workflow, including activation of the ZTA, has been developed to accommodate the service optimisation needs, as shown in Figure 5-5 (left). Figure 5-5(right) shows configuration change requests sent by CDSO, to change the encoder rate of the service.

When ZTA requests a change from CDSO, CDSO communicates with OpenSlice in 5G-VINNI Patras, by sending an HTTP request to the Openslice REST API, which then propagates the request to a custom developed VNF that issues the command to change the encoder configuration.

Figure 5-6 presents the Downlink Bitrate KPI and the Total Rate KPIs during the experiments. We can see that the variation in Downlink Bitrate is well coordinated with the triggering of the rules, around 12:00
and then a few times around 12:30, until it stabilised at the max rate of 26MBPS. Figure 5-7 shows the increase in traffic alongside the Lost Frames KPI. We can see that the lost frames for the 5G application (blue line) were declining. We can derive from this that even with increased traffic in the network, the network observes no loss in service quality due to configuration changes requested by ZTA.

![Figure 5-6: Total Rate changes as a result of ZTA actions that are triggered by changes in Downlink Bitrate](image)

![Figure 5-7: The quality of the service (shown by Lost Frames) stays within the required parameters even with a sharp increase in the traffic (Total Bytes)](image)

### 5.2 ZTA in UC4.4

UC4.4 is about live video professional production from high quality cameras. To ensure such high video quality, the requirements are consistently high bandwidth along with high reliability and low delay. Therefore, the focus is on the uplink performance of the 5G network. To enable this, a network configuration that favours uplink will be activated automatically when the quality of the service starts to degrade.

The way in which we used ZTA in UC4.4 is different than the previous example. The following choices were made, based on the specific needs of this use case:

- **We react** to a degradation in quality of service. This is done by checking the current value of uplink bitrate KPI. Specifically, when the UL bitrate KPI goes under 5Mbps once in 10 seconds, or under 15Mbps 3 times in 60 seconds, the action is triggered.
- The action will result in changes in **network configuration**, specifically the gNodeB configuration is changed: UC 4.4 mainly needs UpLink (UL) BandWidth (BW) in order to provide better quality of experience. The standard gNB configuration, however, provides higher DownLink (DL) BW
compared to UL, because most UCs utilize more DL than UL. This is achieved in the gNodeB by using more time slots for DL, i.e. processing DL packets more often per time unit. Therefore, our action will be to optimize the gNodeB configuration for UL by allocating more processing time slots in the gNodeB for UL than DL.

Similar to the ZTA in UC4.1, we implemented and tested ZTA in UC4.4 end-to-end, including all the components (CDSO, KPI VS, OpenSlice/RMS, ZTA).

The rules defined for ZTA are presented in Figure S-8 below:
In terms of the action, ZTA will send to CDSO the recommendation for gNodeB reconfiguration. CDSO, in turn, will send the request to OpenSlice in the 5G-VINNI Up platform. When OpenSlice receives the request from CDSO, it translates this request and forwards it to a Radio Management Server (RMS) in the UoP platform. The RMS then forwards the requested command to an Agent installed in the corresponding gNodeB. This allows the platform to control multiple devices, as seen in Figure 5-9.

The communication between the various instances is done over HTTP and follow TMF API standards. The components involved are shown below:
The results of the trial show that ZTA correctly triggers when the conditions are met, resulting in changed configuration and a high bandwidth for the application (Figure 5-10):
Figure 5-10: Results of two separate trials for ZTA in UC4.4, which show that ZTA gets triggered when uplink_bitrate KPI falls under 5Mbps, resulting in gNodeB reconfiguration and therefore a higher total_bytes KPI for the service.

The CDSO UI shows the triggering of the ZTA (Figure 5-11):

In the next sub-section, we will discuss how we will use these trials in the Multi-LL scenario.

5.3 ZTA in UC5.3

UC5.3 is a Multi Living Lab use case that runs UC4.1 and UC4.4 in the same slice in the 5G-VINNI UoP platform. In this case, we would like to understand how ZTA works in this combined scenario.
Our investigations will be fulfilled in the context of WP7, and the results reported in deliverable D7.3B. The initial thoughts in this area are:

- Investigate any dependencies introduced by running multiple UCs;
- Understand what results we get from running the individual ZTA optimization loops for the use cases involved, when the UCs are run together in the same slice;
- Investigate if additional rules and actions are needed in the Multi-LL scenario, that are not already covered by the optimization features implemented in the UoP orchestrator.

One initial observation is that the ZTA action that optimizes UC4.4 impacts the gNodeB configuration, which means that it can also impact UC4.1. Our initial experiments will start with exploring this scenario, where at some point UC4.1 needs to operate in a network configured to favour UL. The next step is to understand what we need to change in the ZTA rules to ensure that both UCs (5G services) can run in parallel in an optimal way. The trials are scheduled to start soon (in cycle 3) and all results will be reported in D7.3B.

5.4 ZTA in LL1

Another avenue that we are exploring is to apply ZTA in the context of the Factory-of-the-Future LL. Similar to the MLL case, and based on the results we got until now, all investigations and results will be reported in WP4, deliverable D4.2B. This application of ZTA is highly dependent on the status of the 5G-VINNI Norway platform. Since delays have been reported for this platform, at the time of this deliverable it is still unclear how much we can change in the platform through ZTA (e.g., is Day 2 configuration available?) Because of these reasons, this will be followed up on a best-effort basis.

UC1.5 focuses on remote control and rapid deployment, configuration and testing of new robots. It uses the facilities available at NTNU’s Manulab in Trondheim, Norway.

ZTA can be used in UC1.5 to improve resource allocation for the system components, i.e., automatically adjust the number of allocated resources based on the system needs. This can be done by monitoring the actual resource usage over time and adapting accordingly. Additionally, in the context of UC1.5, the system supports introducing new robots and automatically configuring them on the fly. New resources must then be allocated according to the estimated needs for those new robots. The role of ZTA is to automatically adjust the resource allocation to match the actual needs of those components.

The suggested system architecture is found in Figure 5-12 below.
There are two kinds of resources that could be automatically adjusted:

- **Edge Cloud Resources**: including CPU / memory usage, where ZTA could spawn more resources as needed;
- **The 5G SA Core platform** (shown in Figure 5.13 below as Day 2 config of the 5G SA Core): latency/bandwidth KPIs to be reported into KPI VS e.g., when new robot is coming into the factory floor, ZTA reacts by adjusting the allocated resources via CDSO. This option is only feasible if the required SA Core infrastructure becomes available in due time.

**Figure 5.12: UC1.5 architecture, including ZTA for optimisations**

**E2E Orchestration across Core and Edge for SA, Phase 4 Automation using NCOM**

**Figure 5.13: End-to-end orchestration across Core and Edge domains in 5G-VINNI Norway**
In terms of triggers for ZTA, the initial ideas that are explored are to allow the AAS, which keeps track of robots joining/leaving the system to report changes into KPI VS, e.g., by reporting a variable representing active nodes. This variable can be checked regularly by ZTA through its rules. One additional challenge for this particular UC is to better understand the integration between TaaS and KPI VS, to ensure that reporting can be done in real time when needed.
6 Conclusions

This deliverable has presented our design, implementation and testing of zero-touch automation mechanisms in 5G-SOLUTIONS. We have presented the specific challenges that we encountered in our project in terms of automation and designing an appropriate solution. The solution we present is a lightweight rule-based framework implemented as a Docker container, which can be combined with ML models for enabling proactive optimisations of services, as well as a good coverage of the rules. This solution is integrated into a closed loop, including KPI VS, cross-domain orchestrator CDSO, 5G platform and 5G vertical.

Further, we presented our end-to-end trials for a ZTA in two use cases, one implementing proactive optimisations through service configuration changes, and the other one implementing reactive optimisations through network configuration changes. We have also briefly presented further investigations for ZTA in MLL (WP7) and LL1 (WP4).

The main contributions of this deliverable are (1) exploring and presenting specific challenges for optimization of 5G verticals, as well as our design decisions (2) delivering a lightweight, scalable, easy to integrate solution for ZTA, based on combining a rule-based system with ML methods, to enable different types of optimisations (proactive vs reactive, vertical vs network configuration changes, etc.) (3) presenting the results of our end-to-end trials for two different types of zero-touch automations that we implemented in 5G-SOLUTIONS; these trials successfully proved the validity of our approach.

As our ZTA framework is generic, lightweight and scalable, we envision it as a component that can be used in various settings or projects in the future.
References


[3] 5G-SOLUTIONS, "D1.1B: Definition and analysis of use cases/scenarios and corresponding KPIs based on LLs (v2.0)," 2022.


