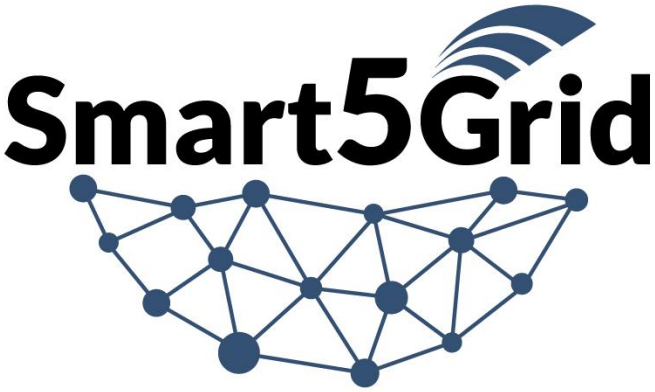

Demonstration of **5G** solutions for
SMART energy **GRID**s of the future

Deliverable D.4.2

Verification and validation framework based on DevOps practices

Version 1.1 - Date 30/06/2023

5G PPP
The 5G Infrastructure Public Private Partnership

This project has received funding from the European Union's *Horizon 2020 research and innovation programme* under grant agreement n° 101016912

D4.2 – Verification and validation framework based on DevOps practices

Document Information

Programme	Horizon 2020 Framework Programme – Information and Communication Technologies
Project acronym	Smart5Grid
Grant agreement number	101016912
Number of the Deliverable	D4.2
WP/Task related	WP4
Type (distribution level)	PU Public
Date of delivery	[30-06-2023]
Status and Version	Version 0.23
Number of pages	27 pages
Document Responsible	Hélio Simeão – UW
Author(s)	Hélio Simeão – UW Gonçalo Leal – UW Roni Sabença – UW Sarigiannidis Antonios - SID Karypidis Paris Alexandros - SID Fountoukidis Eleftherios - SID Karamitsou Thomai - SID Lytos Anastasios - SID Nanos Ioannis - SID Grigoriou Elisavet - SID Saoulidis Theocharis - SID Moukoulis Achilleas - SID Hatjigeorgiou Ioannis - SID Andronikidis Georgios - SID Kyranou Konstantinos - SID

	Chrysagis Konstantinos- SID
	Andrés Cárdenas – I2CAT
Reviewers	Hélio Simeão - UW
	Saoulidis Theocharis – SID
	Kyranou Konstantinos - SID

Revision History

Version	Date	Author/Reviewer	Notes
0.1	12/04/2023	Hélio Simeão, Gonçalo Leal, Roni Sabença	Table of contents (ToC) proposal
0.2	13/04/2023	WP4 partners	Updated ToC
0.3	01/05/2023	WP4 partners	Section 2.1 and 2.2, section 3, 3.1, 3.2, 3.3, 3.4 and 3.5
0.4	21/06/2023	Hélio Simeão	Introduction, Executive Summary and Conclusion
1.0	23/06/2023	Hélio Simeão	First version ready for review
1.1	30/06/2023	Hélio Simeão, Saoulidis Theocharis, Kyranou Konstantinos	Final version with review changes

Executive summary

Network Apps are a way forward in addressing the challenges of the modernisation journey that energy grids are undertaking, by running software solutions in the same (network) infrastructure and in a more distributed manner (edge computing). The advantages are clear, with lower latency, data privacy and less network load, particularly when 5G is the choice for wireless communication, making these technologies very attractive.

In this deliverable we explore how DevOps practices can be put to good use, in the development, testing and deployment of Network Apps across the edge to cloud continuum. DevOps is a widely used methodology in the development of applications that run in cloud environments and provides a path to deploy applications closer to the field devices and equipment. From rugged servers to gateways, actuators or the tiniest sensors that interface with the physical world, edge computing challenges are different from those in traditional computing environments, by combining challenges from operational technology (OT), information technology (IT) and cloud computing environments [1]. The advantages of adopting DevOps practices and tools are enumerated, such as shorter release cycles, easier collaboration and efficiency across development, integration and operation teams, high quality assurance, scalability and flexibility, essential to cope with the fast and demanding pace as networks expand and evolve, while reducing the risk of outage, service degradation or other issues that can severely impact critical infrastructure, like energy grids.

This deliverable also focuses on tools that can be used in the automation of the testing, integration and deployment of Network Apps. The results of the development of the Verification and Validation framework (V&V) are provided in detail, with descriptions of the several components of the solution, its functions, and workflows in verifying and validating a Network App, in the context of Smart5Grid. Network App developers can use the V&V to verify and validate their Network Apps, taking advantage of the work developed in WP4 Network Apps Technology Readiness, particularly T4.3 NFV automatic testing & validation framework through continuous integration (V&V CYCLE).

Table of contents

List of figures

List of tables

1. Introduction

1.1. Scope of the document

This deliverable follows D4.1 Development and deployment of Network Apps [2] for the energy vertical sector, where the concept of Network App in the context of the Smart5Grid project was described. While in D4.1 [2] the focus was on the Network Apps that were developed to address challenges and problems of energy grids, D4.2 focus on the necessary tools that will allow quicker development cycles, alleviate integration and deployment complications. D4.2 studies DevOps practices and why are these are important in the development and deployment of Network Apps, particularly targeting energy grids.

Automating tests and deployment routines are important steps to deliver value through software solutions to the several sectors of the economy, like the Energy sector, contributing to the modernisation and the transformation from traditional grids to smart grids. In order to achieve those steps, the Verification and Validation (V&V) framework developed in WP4 Network Apps Technology Readiness is a tool that can be used by 3rd party experimenters to verify and validate Network App, with the pre-requisite of being developed using the Smart5Grid (S5G) Network App descriptor.

D4.2 describes in detail the V&V main building blocks, namely the API Handler, Verification Engine, Validation Engine and the Results Manager and depicts the several steps and checks that are performed to verify, validate a Network App and store the respective results.

1.2. Document Structure

D4.2 is split into two major sections. Section 2 explains the DevOps methodology, its practices and why these are relevant in the development, testing and deployment of Networks Apps using 5G infrastructure. Concepts such as Edge Computing, that enable to bring more computational power closer to end devices, such as energy grid equipment, are briefly explored and tied to the advantages of DevOps methodology. The main purpose for the automation of the Integration and Deployment routines and how these routines bring innovation and new solutions to the Energy domain are also highlighted.

Section 3 focuses on describing the V&V framework. The V&V components are described in detail along with the workflows that allow a Network App to be verified and validated, from the moment it is uploaded to the Open Service Repository (OSR) in the Smart5Grid Platform. The checks performed to the Network Apps by the verification and validation engine during the V&V cycle are also described in section 3.

1.2.1. Notations, abbreviations and acronyms

Item	Description
API	Application Programming Interface
CD	Continuous Deployment
CI	Continuous Integration
CI/CD	Continuous Integration/Continuous Deployment
CP	Connection Point
ETSI	European Telecommunications Standards Institute
LF	Linux Foundation
LCM	Lifecycle Manager
NAC	Network Application Controller
NSD	Network Service Descriptor
NFVO	Network Functions Virtualisation Orchestrator
NBI	Northbound Interface
NS	Network Services
OSM	Open Source MANO (Management and Orchestration)
OSR	Open Service Repository
SLO	Service Level Objective
S5G	Smart5Grid
VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
V&V	Verification and Validation
WP	Work Package

Table 1: Acronyms list

2. DevOps practices: why do we need continuous integration and deployment

2.1. DevOps methodology

In today's rapidly evolving landscape, the DevOps methodology plays a central role in providing seamless collaboration and integration between software development and operations teams. As software workloads continue to rise in requirements, complexity, and infrastructure becomes continuously more software-defined and thus implementing this methodology is fundamental.

In this manner, DevOps encourages a culture of shared responsibility, where both teams work together throughout the software development lifecycle. One of the essential principles of this methodology involves the substitution of manual and repetitive tasks with automated processes. Thus, it is possible to mitigate human errors since the automatisation process can occur in various deployment stages and in different sections such as infrastructure provisioning, testing, continuous integration and continuous delivery (CI/CD). The software can be easily tested and deployed using those practices. Furthermore, DevOps focuses on evaluation through the implementation of monitoring and logging systems, allowing active monitoring and real-time notification in the event of application downtime or other issues.

In D4.1 [2] the concept of a Network App was explained and Smart5Grid is particularly interested in solutions where the instantiation of Network Apps occurs at the edge, i.e., closer to the energy grid devices. The Linux Foundation (LF) has developed a widely adopted taxonomy that visualizes edge computing across the spectrum of physical infrastructure. This taxonomy offers a representation of edge computing along the continuum of physical infrastructure that encompasses the internet, spanning from centralized data centres to individual devices. Figure 1, provides an overview of the edge computing continuum, ranging from centralized data centres to distributed devices that are resource constrained, highlighting the key trends that define the boundaries of each category. [1]

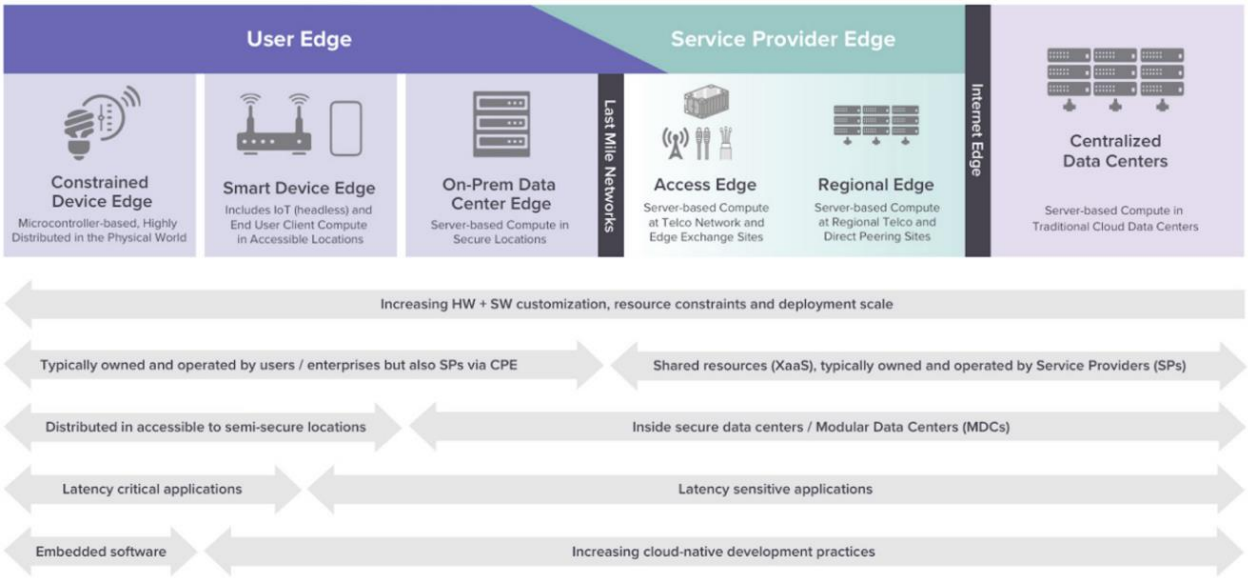

Figure 1: Edge Continuum according to Linux Foundation Edge

Edge computing collaborates with 5G networks, since 5G enables higher transfer rates and lower latency communications, when compared with previous generation of radio technologies. By deploying services at strategic points along the cloud-edge continuum, developers can effectively address the latency requirements of their applications, reduce the amount of data sent to the cloud and assure the privacy of the collected data, by processing it at the edge.

These trends encompass the increasingly complex design trade-offs that architects face, as compute resources are brought closer to the physical world. The DevOps methodology emerges as a framework that enables fast optimization, deployment and operation of the services for the fast-paced 5G network and edge environment on which the Network Apps are deployed. This environment, along with the edge nodes which are servers or devices located at the network's edge, has intermediaries that process data locally before sending it to a centralised data centre.

This methodology is integral to the V&V process of Network Apps in the rapidly evolving edge ecosystem. It aligns with the principles of validation and verification and enables efficient operation and management of services in the 5G network and edge environment. DevOps principles, such as continuous integration and deployment (CI/CD), automation frameworks, and monitoring and management tools, play a crucial role in the V&V of a Network App. Since the CI/CD pipelines ensure that the Network Apps alterations are tested and deployed quickly and reliably, as to be tested via the V&V.

The collaborative and automated nature of DevOps promotes effective collaboration between development and operations teams, leading to continuous improvement in the V&V process. By leveraging DevOps, organizations can optimise performance, ensure reliability, and meet the demands of the fast-paced 5G network and edge environment.

In summary, the DevOps methodology acts as the driving force behind the V&V process, enabling developers to effectively deliver and operate services in the cloud-edge computing landscape.

2.2. DevOps for Network Apps

Developing software solutions for critical services and systems like the Energy sector requires constant adaptation and flexibility. CI/CD practices can empower development and operation teams in delivering faster, more secure reliable software.

The main gains in adopting a DevOps approach in the development of Network Apps are the following:

- Rapid release cycles: Network apps often require frequent updates and feature releases to keep up with evolving network needs and technologies. CI/CD automates and simplifies the process of building, testing, and deploying these updates, enabling a fast and reliable release cycle. This allows the ability to apply changes and new features to the network apps quickly and seamlessly.
- Collaboration and efficiency: CI/CD practices foster collaboration and efficiency among development, operations, and network teams. Network apps can be seamlessly shared and tested across teams by automating the integration and deployment process. This streamlines communication reduces errors, and accelerates the overall development and deployment process.

- Quality assurance: CI/CD enables network apps to undergo continuous testing throughout the development and deployment pipeline. Automated tests can be executed after each integration to ensure that the app functions as expected and remains stable. Issues and errors are detected early, allowing for swift resolution and maintaining overall app quality.
- Scalability and flexibility: As network requirements evolve and expand, network apps need to adapt accordingly. CI/CD provides the scalability and flexibility required to accommodate changing network environments. Automating the build and deployment processes makes it easier to scale and deploy network apps across different environments, ensuring consistency and reducing manual errors.
- Risk reduction: CI/CD practices help mitigate risks associated with network app deployment. By automating testing, continuous integration, and deployment, issues can be identified and resolved early in the process. This minimizes the risk of deploying faulty or incompatible network apps, reduces potential downtime, and improves overall network reliability.

Overall, CI/CD practices for network apps in DevOps enable faster, more efficient, and higher-quality development and deployment cycles. They allow network teams to adapt to changing needs, collaborate effectively, and deliver robust and reliable network apps that align with the evolving requirements of the network infrastructure.

2.3. Benefits and advantages for Energy Vertical solutions

The deployment of vertical services in the smart grid domain can greatly benefit from the V&V approaches and the related adoption of new emergent paradigms that supports the easy and flexible deployment of vertical application as such the one put in place in Smart5Grid.

Mainly in this section we will focus on two key aspects: the benefit brought by 1) the synergy between 5G, Edge computing and Network Apps and 2) the benefit of the adoption of DevOps strategies in the Smart5Grid ecosystem for deploying Network Apps.

In the Infrastructure as a Service (IaaS) practices combined with Continuous Integration and Continuous Deployment (CI/CD) methodologies there is an exploitation of IaaS practice of CI/CD for deploying vertical services in the smart grid domain.

The advantages are the following:

Scalability and Flexibility: By utilizing virtualized infrastructure and cloud resources, the smart grid can easily adapt to varying workloads and demand patterns. CI/CD practices further enhance this scalability by automating the deployment process, enabling seamless scaling of services based on real-time demand. This ensures that the smart grid infrastructure can be tailored for new vertical services and effectively better manage huge data volumes without compromising performance or reliability.

Rapid Development and Time-to-Market: CI/CD methodologies as the such enabled by the V&V provide a framework for rapid development and deployment cycles, allowing for faster time-to-market for new smart grid services. Developers can continuously integrate and test code changes, ensuring early detection of potential issues and also as already discussed to be only focused on the application side, by keeping

the automation to take care of the underlying complexity of the 5G network. By automating the deployment pipeline, vertical services can be quickly provisioned and released into production, reducing the time required for manual configuration and verification. This agility makes it possible to adapt to customer needs and also to keep up with regulatory requirements in the dynamic energy domain.

Cost Optimization: A platform like the one offered by Smart5Grid, allows CI/CD practices to further optimize costs by streamlining the development and deployment processes, reducing human intervention and the potential for errors. This results in efficient resource utilization, minimized downtime, and overall cost savings for both infrastructure and operations.

Enhanced Security and Resilience: Deploying vertical services in the CI/CD practices, enhances the overall security by automating security tests and vulnerability assessments, enabling rapid response to emerging threats. Additionally, the cloud's distributed nature and built-in redundancy enhances the resilience of smart grid services, minimizing the impact of potential disruptions.

Interoperability and Integration: The smart grid ecosystem comprises diverse stakeholders, including utilities, grid operators, renewable energy providers, and consumers. S5G Open Platform provides integration capabilities that enable seamless interoperability between various systems and services. CI/CD practices facilitate the integration of new vertical services with existing infrastructure, ensuring compatibility and smooth operation. This interoperability allows for effective data exchange, improved collaboration, and the creation of innovative solutions in the smart grid domain.

3. Verification and Validation (V&V) Framework

The S5G Open Platform, highlighted in Figure 2, provides standardization and openness for software solutions, referred to as Network Apps, that use the 5G network. For developers this standardization is beneficial since it provides an open architecture and standardized interfaces that facilitate seamless integration with different network elements and components.

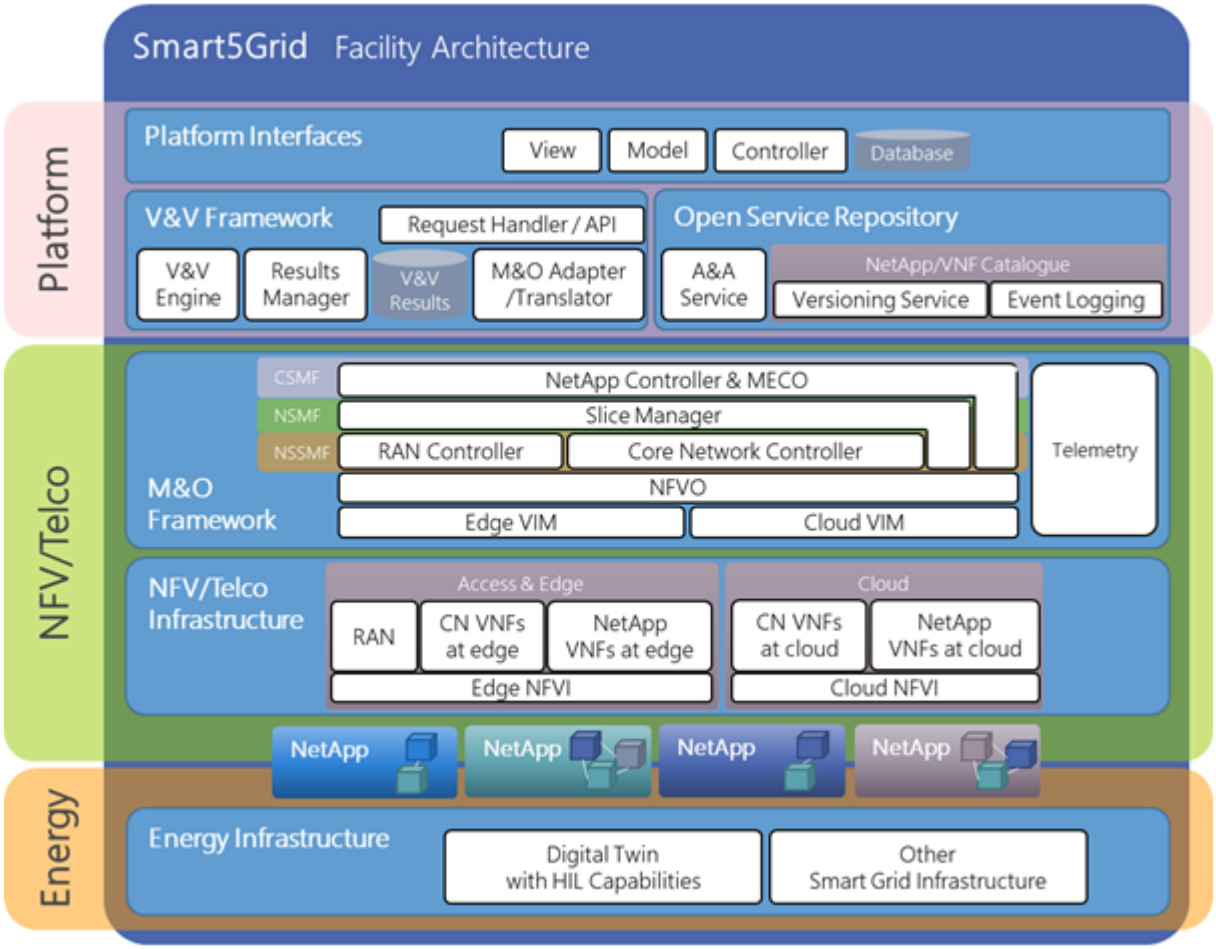

Figure 2: Smart5Grid Open Platform [3]

The V&V aims to be a facilitator for the deployment of Network Apps in the 5G infrastructure, offering testing and deployment capabilities with the 5G infrastructure. Developers can analyse and validate their solutions to ensure that they meet the expected standards and consequently making the deployment process faster since errors and anomalous behaviours can be detected.

For a developer to have access to the V&V, it is solely required to upload a Network App to the OSR (Open Service Repository), initiating the process for the verification and validation, which is transparent to the user. The verification process checks the integrity and correctness of the YAML files, ensuring correct structure and syntax. Additionally, it checks for any errors, inconsistencies, or missing information within the YAML files. On the other hand, the validation process focuses on assessing the overall functionality and behaviour of the Network App. It involves executing the validated YAML files and evaluating the actual results against the expected outcomes. This step assures that the solution performs as intended and meets the specified requirements. Both validation and verification are crucial and distinct operations in the

development and evaluation of Network Apps. While verification ensures the YAML files are valid and well-formed, validation tests the functionality and verifies if the solution behaves correctly in live or production scenarios.

3.1.V&V Engine

The process of verification and validation is encompassed by the V&V platform, which itself consists of two engines: the verification engine and the validation engine. Both engines play crucial roles, the Verification Engine encompasses checks to the Syntax, Integrity, and Topology of the Network App. It scrutinizes the syntax of the files to identify any structural or formatting issues, ensuring they comply with the expected standards. Additionally, it verifies the integrity of the data, ensuring its consistency and accuracy. Additionally, it also verifies the overall architecture and connectivity, known as topology, of the system or software being assessed.

As to the Validation Engine, it focuses on the practical aspects of the system or software. It verifies the successful onboarding and instantiation process, ensuring that the system can be deployed and initialized without any issues. It can also validate the Key Performance Indicators (KPIs) by retrieving relevant data and comparing it against predefined criteria, ensuring that the system operates as intended and meets specific requirements. Once all the checks are finished, the results are stored in a dedicated database, the Result Manager.

To facilitate the visualization of these results, the Argo Workflows platform was used and configured to allow all the different parts of the V&V process to be illustrated, as shown in the Figure 3.

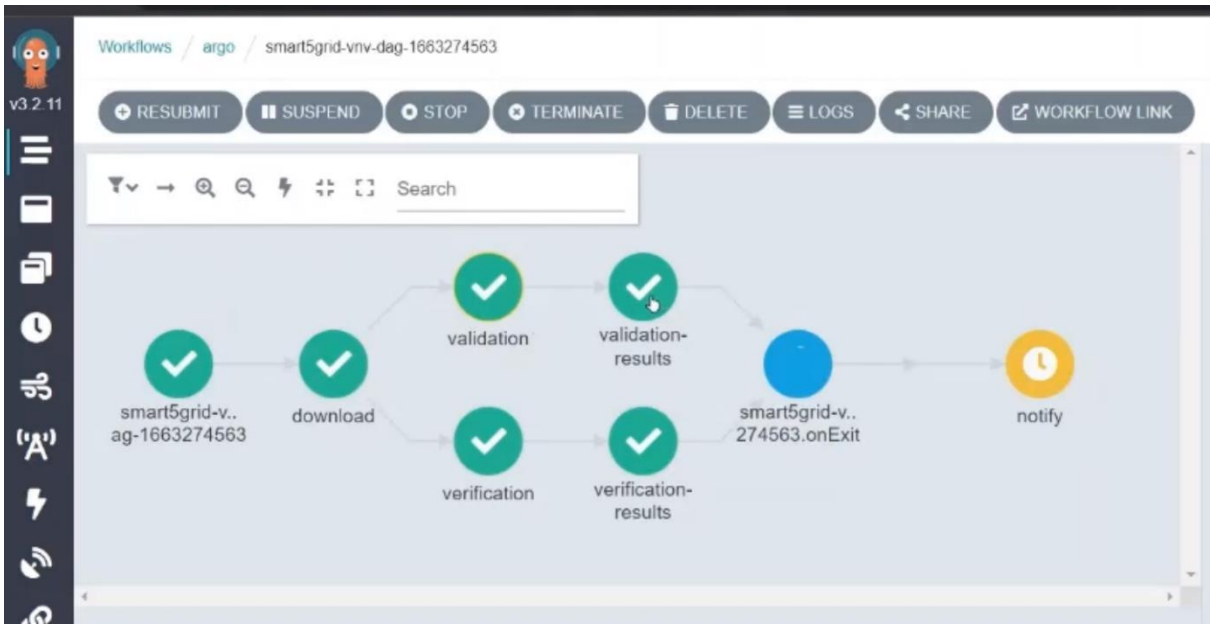

Figure 3: Example of the V&V Cycle

3.2. Request Handler / API

The Request Handler API, highlighted in Figure 4, serves as the entry point for submitting the Network Apps for verification and validation process. This API receives two parameters which are the name of the Network App to be analysed and whether the analysis is for verification, validation, or both. As a pre-requisite, the Network App must be already stored in the OSR, for the Request Handler to proceed with downloading the corresponding Network App.

During the verification and validation process, the Request Handler receives a zipped file of the Network App for verification, while for validation, it will receive the JSON representation of the YAML file. After the V&V cycle is finished, the results are inserted into the database, specifically the Result Manager collection. The following figure depicts a high-level workflow of the solution.

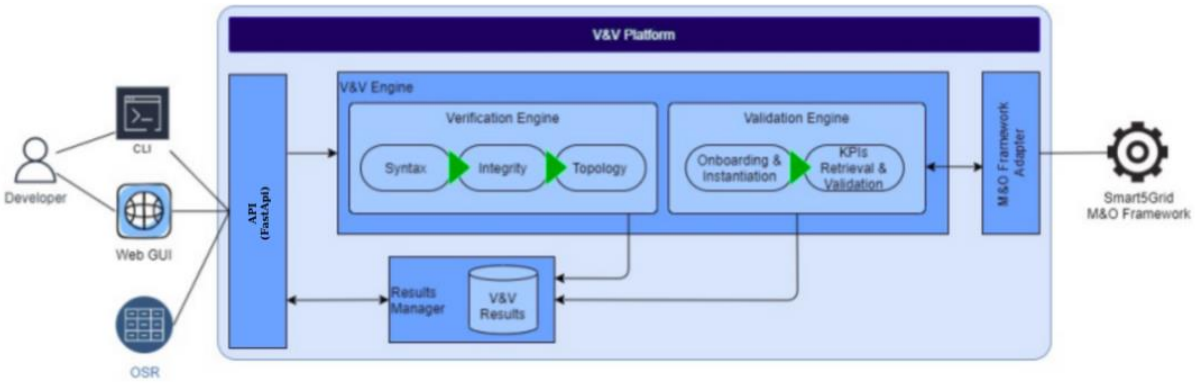

Figure 4: V&V High-level Architecture

3.3. Verification Engine

The requirements for the verification Engine have been aligned with the UCs needs [3], following the objective of accelerating the Development and Operations (DevOps) of the developer, enabling a continuous improvement of services, thus achieving a continuous integration and development workflow. It introduces a first phase of Network Application analysis used for checking the software code for errors introduced in the coding phase prior to its deployment. In this initial phase, the V&V can perform static checks over the end-to-end Network Application without the need of a target infrastructure or input data.

This component provides the capabilities to create a quality assurance pipeline to use on the artefacts of a Network Application technology chain, which includes, in the most complex type; the Smart5Grid Network Application descriptor, Network Functions descriptors (OSM release>=9), Helm charts and Docker container images. Several scenarios are possible depending on the type of Network Application being checked, purely Cloud-Native represented by the upper chain in Figure 5, or a more telco approach, highlighted in the bottom chain of the same figure. In either case, the verification engine deals with the following elements.

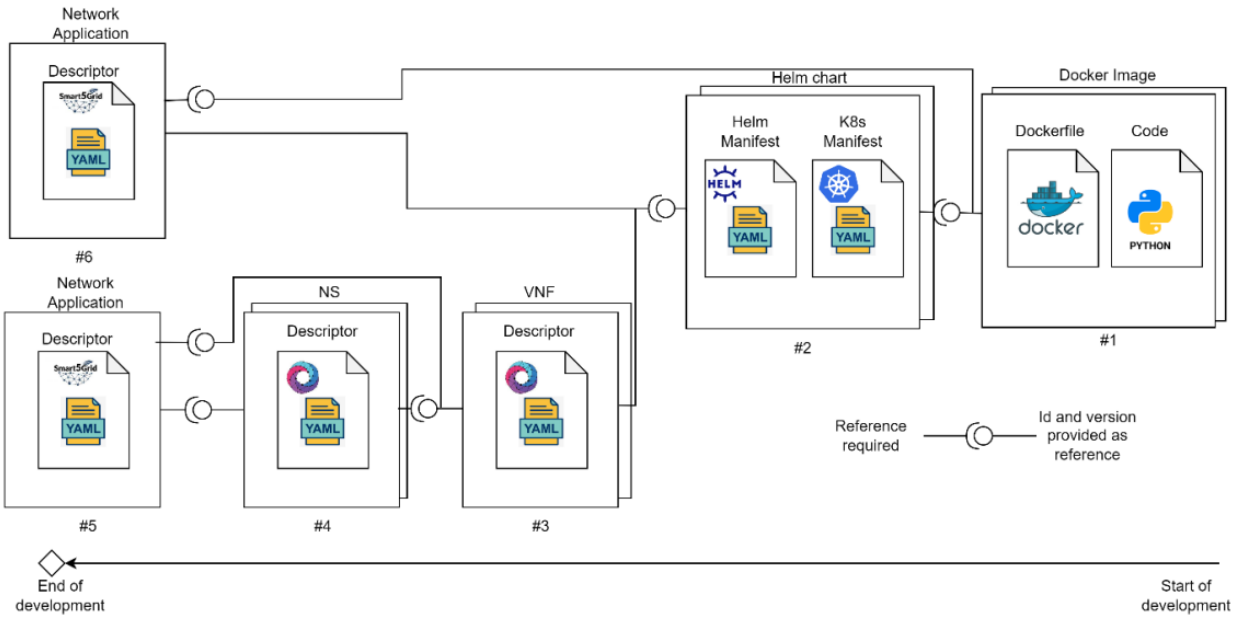

Figure 5: Network Application technology chain [D4.1]

Syntax verification: Each piece of the Network Application chain is syntactically contrasted with the corresponding information model defined either by an external community or by Smart5Grid.

Integrity verification: This check detects bugs in the overall structure of descriptors through the inspection of references and identifiers, both within and outside the individual descriptors. The integrity check ensures that the references are valid by checking the existence of the next component in the chain as well as the connection points (CPs) when applicable.

Topology verification: This check goes a step beyond the integrity verification by executing a set of mechanisms to verify the individual and end-to-end network connectivity graph. Unlinked and unreferenced components are detected per descriptor as well as unintentional functional blocks based on the definition of the objective of each key in the corresponding information model, e.g., invalid internal to external CPs mappings, unexposed internal CPs that are not marked as management CPs, network loops and cycles etc.

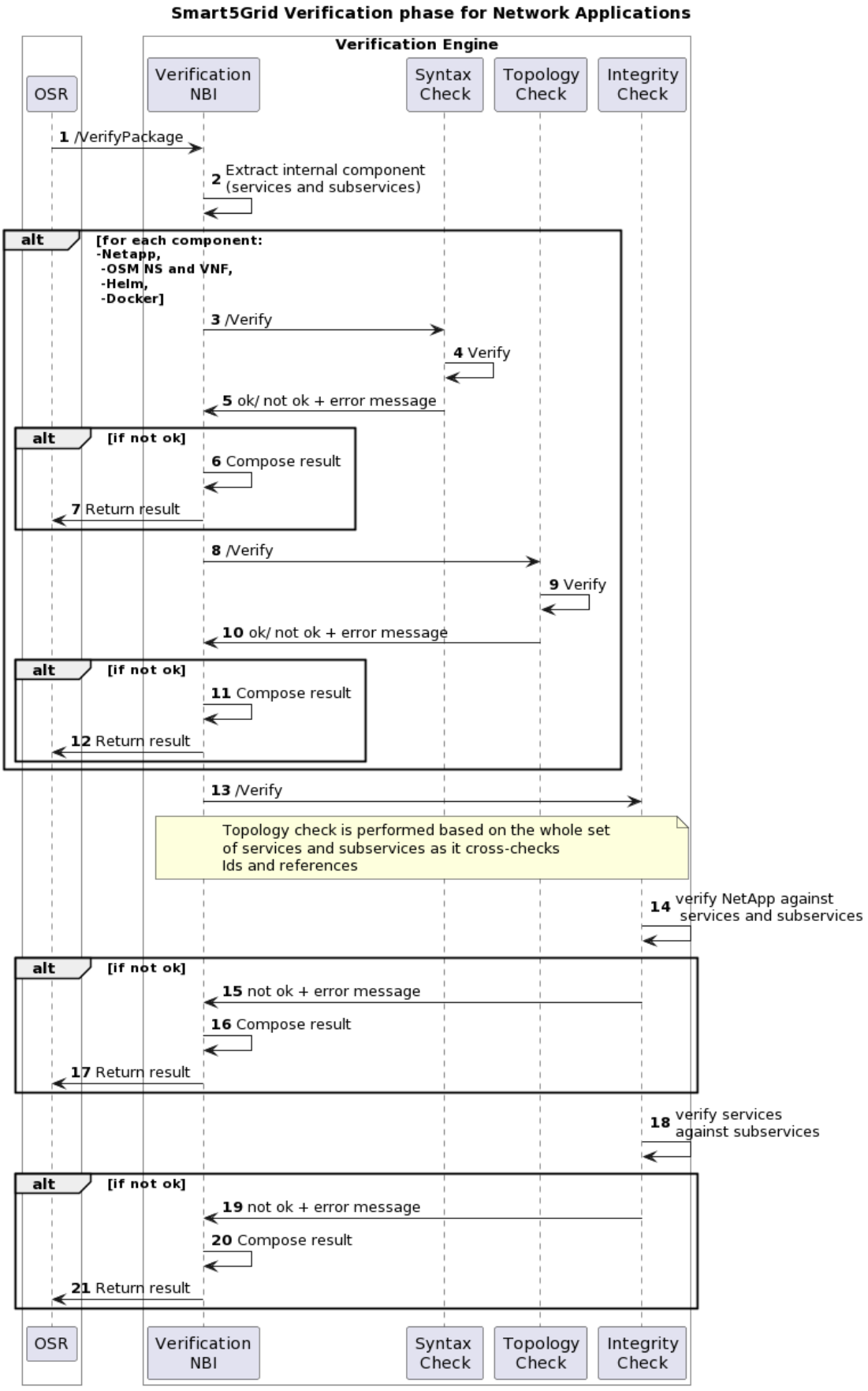

Figure 6: Workflow of the Verification Engine

3.4. Validation Engine

To carry out the Validation phase, the V&V onboards the Network App on the NAC and requests the deployment of a Network App instance. Once the instance is deployed, the V&V can request its status to the NAC. Once a successful status is reached, the V&V can proceed with the teardown by terminating the instance and deboarding the Network App.

3.4.1. Validation in UC2

The validation phase for UC2 makes use of the Telco-oriented NAC. The main feature of this NAC is that it can interact with telco platform composed of management and orchestration components of access, computational and service networks. For this purpose, the Network App descriptor includes in its information model descriptors of Network Services (NS) and Virtual Network Functions (VNFs) from the European Telecommunications Standards Institute (ETSI) Open-Source MANO (OSM). Part of the telco platform with which NAC interacts for application deployment has been described in D2.2 [2] and D3.2 [4].

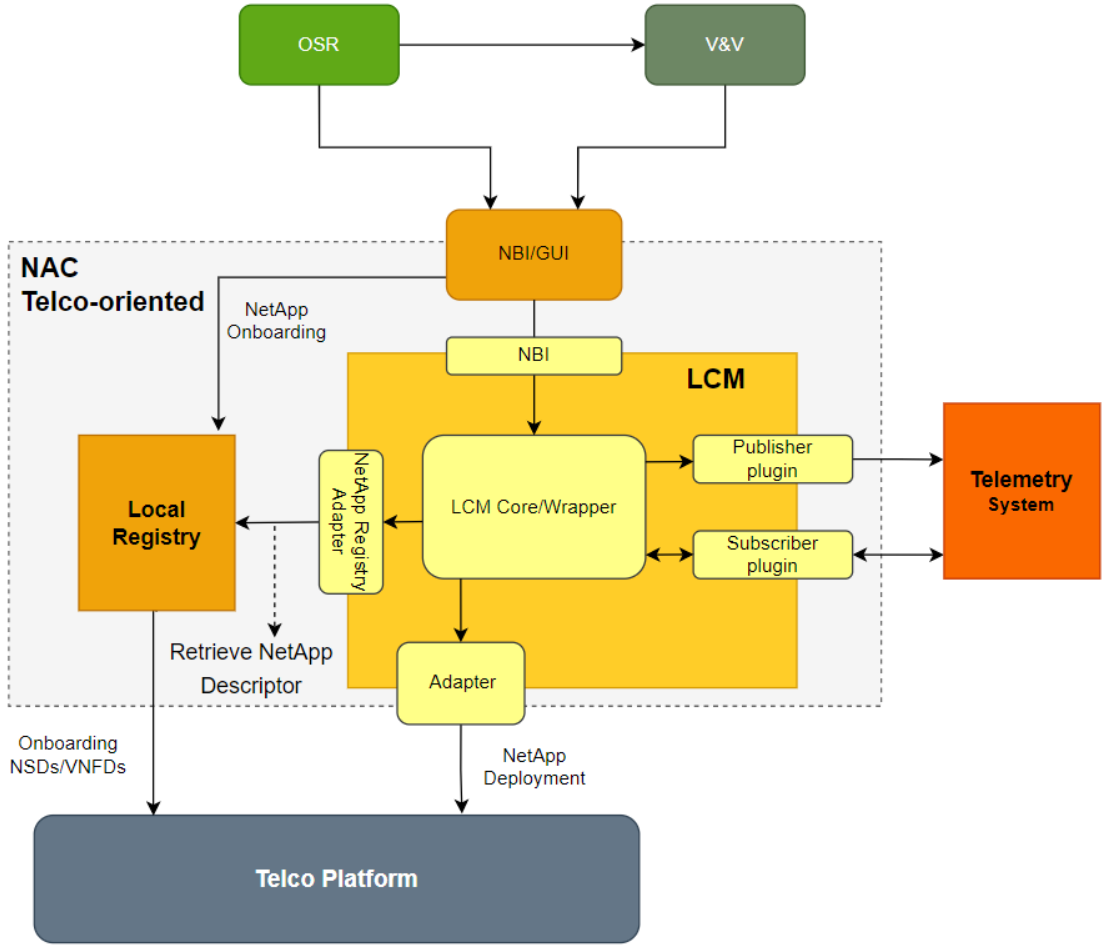

Figure 7: UC2 NAC Architecture

UC2 NAC, as shown in Figure 7, is composed of three main modules:

- **Northbound Interface (NBI):** It is in charge of exposing the API towards end users and external components.
- **Lifecycle Manager (LCM):** Manages the stages of an application's lifecycle, such as: deploying, reading, updating and deleting applications.
- **Local Registry:** It is in charge of storing the different descriptors (Network App descriptor, Virtual Network Function Descriptors (VNFDs) and Network Service Descriptors (NSDs)) required for the deployment of a Network Application.

Each of these modules interact with each other depending on the lifecycle management stage of an application. The validation steps with the UC2 NAC are described below:

1. **Network App Onboarding:** The V&V component sends the command to load the packages in NAC by interacting with NAC's NBI. NAC collects the files and packages them to be stored inside the Local Registry module. On the one hand the Network App descriptor will be available to the LCM in order to do the requirements translation, while the NSD and VNFD descriptors are loaded into the telco platform. For this purpose, the Local Registry component interacts directly with the Telco platform.

2. **Network App Deployment:** Once the loaded descriptors are stored and available in the Local Registry module, the LCM can make use of them to read and translate the specifications and requirements of the Network Apps. In this phase, the NAC receives the request for instantiation of a Network App through its NBI. The NBI module redirects this request to the LCM by interacting with the LCM API. The LCM performs the necessary translations of the Network App descriptor and communicates with the Telco platform to execute the Network App instantiation. At the same time, it sends the relevant aspects of monitoring and SLOs (service level objectives) to the telemetry system, which will start monitoring certain metrics of the system.

3. **Runtime Actions:** If the application has been deployed correctly, the LCM takes care of checking its execution status. In addition, it is constantly listening to the telemetry system in case there is any variation in application performance. If there is a variation of SLOs, the LCM receives a notification from the telemetry system to either scale applications or to reduce replicas. In this case, the LCM follows the traditional application deployment or removal procedure.

4. **Decommissioning:** Finally, when the application needs to be removed, the end user or an external component (e.g., V&V) sends the request to the NAC. This in turn, through its NBI, redirects the request to the LCM. Subsequently, the LCM interacts with the telco platform and removes the application from the computational infrastructure. It also sends a message to the telemetry system that the App has been successfully removed.

The NBI interacts with various components of the NAC such as the LCM and local registry along with components outside of the NAC such as the V&V and OSR. The first part of the NBI is to receive a request from the V&V to deploy the Network Application, thus beginning the communication process of the NAC which is described in this section. This communication occurs for the purpose of onboarding the Network

Application package. That package can have several forms and it can be received in various ways. The NBI after initializing a communication with the OSR (prerequisite: stored Network Application in the OSR), requests all the sub-packages required to instantiate the Network Application (Docker images –optional-, Helm chart, VNFs, NSs). Once the OSR sends all those packages and the NBI receives them, a bundled Network App artefact (.tar.gz file) is created by the NBI. That bundled file is sent to the local registry which stores everything internally and onboards the required descriptors to OSM. The NBI, after receiving a confirmation from the local registry that the package was correct and the communication was successful, sends a request to the LCM in order to begin the instantiation of the Network Application.

3.4.2. Validation in UC1, UC3 and UC4

For what concerns NearbyComputing's NearbyOne as NAC, and thus UC1, UC3, and UC4, the API was initially described in D2.2 [5], refined in its the reference repository [6] and presented in D3.2 [4]. This API is used by the V&V Framework to interact with the NAC during the validation process. This API is offered by a specific component of this NAC developed for the Smart5Grid project - the Smart5Grid Network App Adapter. As the name suggests, this adapter translates the concept of Smart5Grid Network App to their equivalent NearbyOne services. Since NearbyOne is a commercial product that already exists and has internal API which cannot be divulged, when using it as a Smart5Grid NAC, this adapter and the translation it carries out were required. In fact, as Figure 8 shows, the adapter is exposed in the frontend of the platform so that it can be directly reached by external entities (i.e., the V&V Framework)

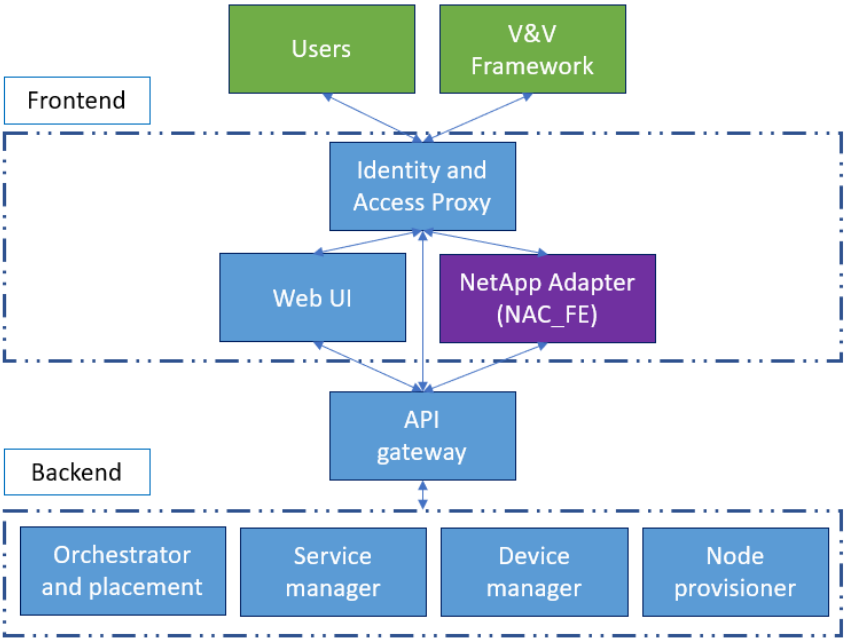

Figure 8: Internal Architecture of NearbyOne as Smart5Grid NAC

As described in detail by the UML Sequence Diagram in Figure 8, the validation process of NearbyOne NAC consists of four consecutive steps:

1. Onboard the Network App: This is where the Adapter translates the Network App into a NearbyBlock, the equivalent representation of a Network App into NearbyOne. This step is covered by arrows 1.x of the UML diagram.
2. Deploy a Network App instance: In this step the requests travel across the NAC and reach the NFVO in order to deploy the instance itself. In turn, the NFVO internally enqueues this request. Possible errors that might occur during this step are propagated back to the V&V. This step is covered by arrows 2.x of the UML diagram.
3. Get Instance status: This step occurs over time and with different actors working in parallel.
    I. On one side, the NAC keeps polling the NFVO for the status of the instance. This sub-step is covered by arrows 3.1.x of the UML diagram.
    II. Meanwhile, the V&V keeps polling the NAC for the instance status as well. Arrows 3.2.x of the UML diagram.

Having these two parallel flows allows the NAC to reuse the same flow to check an instance status during production operations outside of the V&V flow. Furthermore, the NAC can use the cached status of the instances to swiftly reply to the V&V framework without having to synchronously fetch the status of all components of the Network App.

4. Undeploy the Network App: This final step consists of the teardown of the Network App instance. The request is propagated until the NFVO and every component does the proper cleanup. This step is covered by arrows 4.1 to 4.7 of the UML diagram.

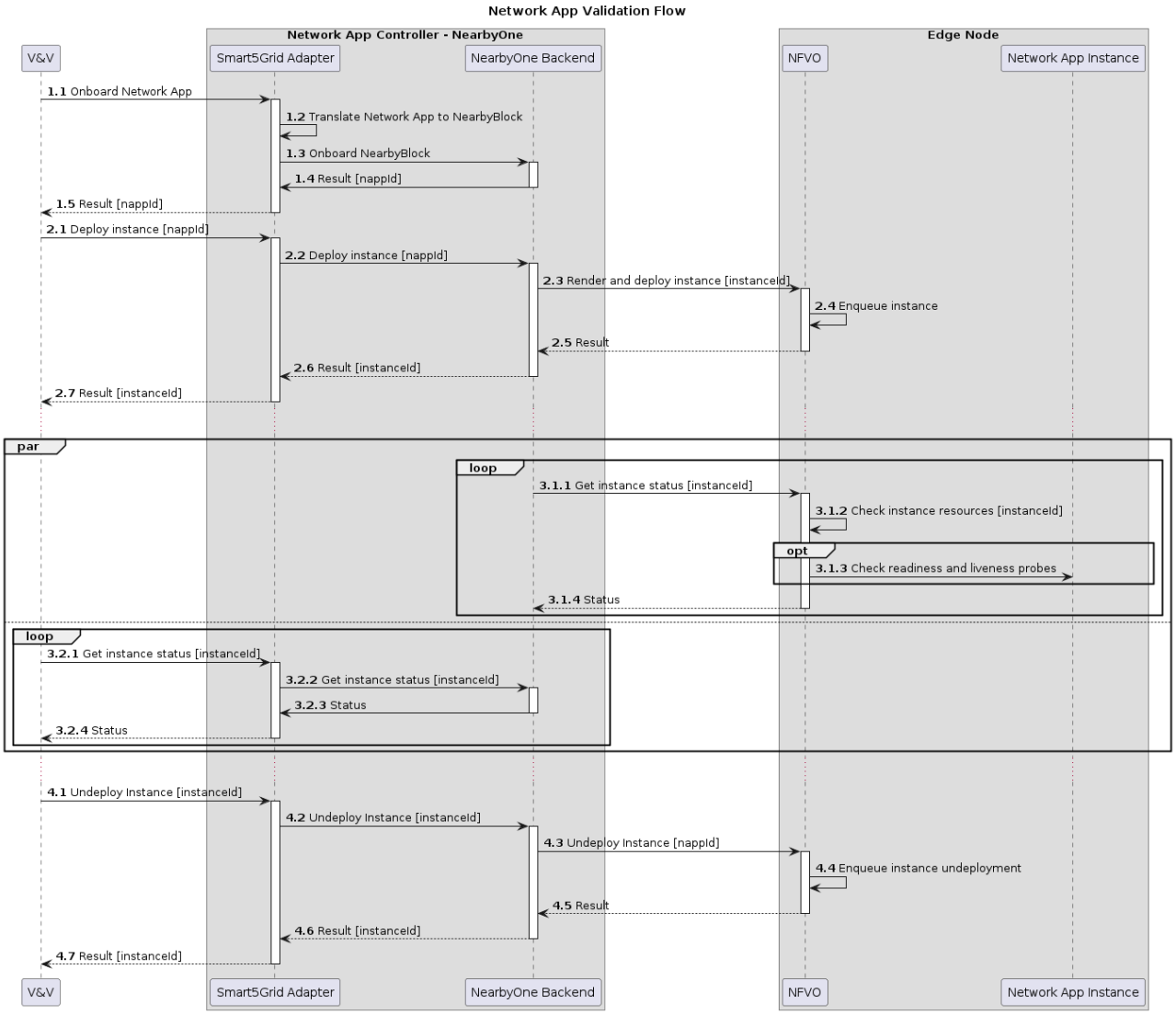

Figure 9: Network App Validation Flow with NearbyOne

3.5. Results Manager

The Result Manager is a crucial component in the Verification and Validation (V&V) process of systems and software. It plays a vital role in storing the results obtained during verification and validation phases. Leveraging a relational MongoDB database, the Result Manager efficiently handles and organizes the data related to these results, using specific parameters provided within the database.

The parameters within the MongoDB database include the following:

"_id": A unique identifier assigned to each result entry in the database.
"id": An identification number associated with the net application.
"Network AppId": A reference to the net application's unique identifier.
"done": A boolean value indicating whether the V&V process for the net application is completed.
"validate": A boolean value representing the validation status of the net application.
"verify": A boolean value indicating the verification status of the net application.
"timestamp": The date and time when the result entry was recorded.

"results": An array containing detailed information about the validation and verification processes. Within the "results" array, individual objects represent different types of results, namely "validation" and "verification". Each object includes the following key-value pairs:

"type": Indicates whether the result is from a validation or verification process.
"date": The date when the result was generated, in the format "YYYY-MM-DDT00:00:00".
"Network AppId": A reference to the net application's unique identifier.
"failed": A boolean value indicating whether the result indicates a failure (true) or success (false).
"errorCode": An error code associated with the result, if applicable.
"errorDescription": A description of the error encountered, if applicable.
"details": Additional details or data related to the result.


```
[
    {
        "_id": "6470e73c2dc064e8d446c6c5",
        "id": "1685120826.857152-your_net_app_name",
        "netAppId": "your_net_app_name",
        "done": true,
        "validate": true,
        "verify": true,
        "timestamp": "2023-05-26 17:07:08",
        "results": [
            {
                "type": "validation",
                "date": "2023-05-26T00:00:00",
                "netAppId": "your_net_app_name",
                "failed": false,
                "errorCode": 1,
                "errorDescription": "nothing",
                "details": "dummydata"
            },
            {
                "type": "verification",
                "date": "2023-05-26T00:00:00",
                "netAppId": "your_net_app_name",
                "failed": false,
                "errorCode": 1,
                "errorDescription": "nothing",
                "details": "dummydata"
            }
        ]
    }
```


Figure 10: Data Structure of the Results Manager DB

Overall, the Result Manager, in conjunction with the MongoDB database parameters, enhances the V&V process by providing a reliable and organized platform for storing, managing, and analysing verification and validation results.

4. Conclusions

This deliverable focuses on the output of Task 4.3 NFV automatic testing & validation framework through continuous integration (V&V CYCLE) of the WP4 Network Apps Technology Readiness.

The goal is to leverage DevOps principles and its benefits across the cloud to edge continuum, by automating the testing and deployment of Network Apps, in the context of Smart5Grid project. Edge computing brings more computational resources closer to field equipment, which paired with the strong coverage, reliability, and low latency of 5G make an interesting value proposition for smart grids.

The V&V framework developed in the scope of T4.3 automates testing routines by combining several components, namely the Request Handler, Verification Engine, Validation Engine and Results Manager, all of which are described in detail in this document.

The workflow starts with the Open Service Repository (OSR) triggering the V&V, via the Request Handler (assuming a Network App is available in the OSR) to start the verification and validation process. During the verification process, syntax, integrity, and topology checks are performed to the Network App, while the validation process encompasses steps such as the Network App onboarding, the deployment of a Network App instance and the un-deployment of the Network App. The results of the of the previously described steps are stored in the Results Manager, where developers can check if their Network Apps passed the tests and the respective error logs for the unsuccessful tests.

Automatisms like the V&V are key in reducing time-to-market of new software solutions, provide an easy path for continuous upgrades and improvements, under shorted release cycles.

5. References

[1] "The State of the Edge Report 2021, Linux Foundation," [Online]. Available: https://stateoftheedge.com/reports/state-of-the-edge-report-2021/.

[2] "Smart5Grid deliverable D4.1 "Development and deployment of NetApps for the energy vertical sector"," [Online]. Available: https://smart5grid.eu/dissemination-activities/deliverables/.

[3] "Smart5Grid deliverable D2.1 "Elaboration of Use Cases and System Requirements".," [Online]. Available: https://smart5grid.eu/wp-content/uploads/2021/07/Smart5Grid_D2.1_Elaboration-of-UCs-and-System-Requirements-Analysis_V1.0.pdf.

[4] "Smart5Grid deliverable D3.2 "Final report for the development of the 5G network facility"," [Online]. Available: https://smart5grid.eu/wp-content/uploads/2023/03/Smart5Grid_WP3__D3.2_PU_Final-report-development-5G-network-facility_V1.0.pdf.

[5] "Smart5Grid deliverable D2.2 "Overall architecture, design, technical specifications, and technology enablers"," [Online]. Available: https://smart5grid.eu/wp-content/uploads/2021/11/Smart5Grid_WP2_D2.2_V1.0.pdf.

[6] "NearbyOne's Smart5Gird NAC_FE REST API definition in Swagger 2.0.," [Online]. Available: https://bitbucket.org/AntonelloCorsi/NearbyOne/src/master/.

[7] "From DevOps to EdgeOps: A Vision for Edge Computing, White Paper by Eclipse Foundation," [Online]. Available: https://outreach.eclipse.foundation/edge-computing-edgeops-white-paper.

[8] "Smart5Grid deliverable D3.1 "Interim Report for the development of the 5G network facilities"," [Online]. Available: https://smart5grid.eu/wp-content/uploads/2022/11/Smart5Grid_WP3__D3.1_PU_Interim-Report-for-the-development-of-the-5G-network-facilities_V1.0.pdf.