EXPERIMENTATION AND VALIDATION OPENNESS FOR LONGTERM EVOLUTION OF VERTICAL INDUSTRIES IN 5G ERA AND BEYOND

[H2020 - Grant Agreement No.101016608]

Deliverable D4.4

Network Apps for Interactions of Employees and Machines

Editor C.Bailly (IMM)

Contributors (INF), (GMI), (IMM), (NCSRD), (UMA)

Version 1.0

Date August 31st, 2023

Distribution PUBLIC (PU)

DISCLAIMER

This document contains information, which is proprietary to the EVOLVED-5G ("Experimentation and Validation Openness for Longterm evolution of VErtical inDustries in 5G era and beyond) Consortium that is subject to the rights and obligations and to the terms and conditions applicable to the Grant Agreement number: 101016608. The action of the EVOLVED-5G Consortium is funded by the European Commission.

Neither this document nor the information contained herein shall be used, copied, duplicated, reproduced, modified, or communicated by any means to any third party, in whole or in parts, except with prior written consent of the EVOLVED-5G Consortium. In such a case, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced. In the event of infringement, the consortium reserves the right to take any legal action it deems appropriate.

This document reflects only the authors' view and does not necessarily reflect the view of the European Commission. Neither the EVOLVED-5G Consortium as a whole, nor a certain party of the EVOLVED-5G Consortium warrant that the information contained in this document is suitable for use, nor that the use of the information is accurate or free from risk and accepts no liability for loss or damage suffered by any person using this information.

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

REVISION HISTORY

Revision	Date	Responsible	Comment
0.1	*June, 23rd , 2023*	*G.Makropoulos*	*TOC*
0.2	*July, 12th , 2023*	*C.Bailly*	*Initial contribution*
0.3	*July, 31st , 2023*	*G.Makropoulos*	*First internal review*
0.4	*August, 18th , 2023*	*J.Garcia*	*Internal review*
0.5	*August, 24th , 2023*	*C.Bailly*	*Comments addressed*
0.6	*September 21st, 2023*	*C.Bailly*	*TSN content added*
0.7	*September 25th, 2023*	*F. Luque*	*TSN content reviewed*

LIST OF AUTHORS

Partner ACRONYM	Partner FULL NAME	Name & Surname
GMI	GMI	Marc-Olivier Sauer
GMI	GMI	George Kanterakis
INF	Infolysis	Konstantinos Fragkos
INF	Infolysis	Vaios Koumaras
INF	Infolysis	Christos Sakkas
INF	Infolysis	G. Theodoropoulos
INF	Infolysis	Antonios Varkas
INF	Infolysis	Georgios Koumaras
IMM	Immersion	Charles Bailly
IMM	Immersion	Julien Castet
UMA	University of Malaga	Francisco Luque

GLOSSARY

Abbreviations/Acronym	*Description*
5GS	*5G System*
AR	*Augmented Reality*
CAPIF	*Common API Framework*
CPE	*Customer Premises Equipment*
FSM	*Finite State Machine*
GUI	*Graphical User Interface*
IEM	*Interaction between Employees and Machines*
MNO	*Mobile Network Operator*
MRO	*Maintenance, Repair and Overhaul*
NEF	*Network Exposure Function*
SDK	*Software Development Kit*
SME	*Small and Medium Enterprises*
TSN	*Time-Sensitive Networking*
UE	*User Equipment*
vAPP	*Vertical Application*

EXECUTIVE SUMMARY

The objective of this deliverable is to present in detail the **final prototypes** and the **two integration rounds for testing and validating the use cases** that have been followed for each of the three Network Applications by the three SMEs participating in task 4.2.

- Initially, the deliverable describes in detail the **final prototypes of the Network Apps** developed within the Interaction between Employees and Machines (IEM) in the EVOLVED-5G context, driven by Task 4.2:
  - The Remote Assistance in AR Network App by IMM.
  - The Chatbot Assistant Network App by INF.
  - The Digital/Physical twin Network App by GMI.
- Next, the **two cycles of integration activities and use case testing** that have been followed for each of the three Network Applications, are described.
  - The first round of integrations has been carried out with the aim at ensuring seamless and reliable communication between various components within the system, including Network Apps, Vertical Apps (vApp), NEF, CAPIF and 5G network connectivity, on top of the cloud infrastructure provided by the Athens and Malaga platforms. **The connectivity of 5G with the cloud infrastructure has been verified, specifically the connection between vApps and the 5G network**.
  - The purpose of the second integration round was to validate the use-cases utilizing the final components developed in the context of the EVOLVED-5G project. On the one hand, NEF, CAPIF and the SDK were enriched with additional features, as described in D3.3 and D3.4. On the other hand, SMEs finalized their Network Apps by enhancing the 3.0 version and using the last versions of NEF, CAPIF and SDK, until the latest version of the Network Apps was finally developed: 4.1. This version 4.1 of the Network Apps also exploited the validation pipeline before the integration tests took place. The Networks Apps were deployed in Kubernetes clusters in Athens and Malaga premises instead of using Docker containers running locally. Finally, IMM tested their Network App with UMA's TSN setup. With the second round of integration tests, the Networks Apps of the pillar have reached their final stage, interacting with the last versions of NEF and CAPIF through the SDK and communicating with their respective vApp(s). **The three SME use-cases have also been validated** and such results highlight the fact that the Network Apps reached a mature enough state to be used by other SMEs through the EVOLVED-5G Marketplace.

As a final point, in the context of EVOLVED-5G, it is essential to highlight that a terminology update has been implemented. Specifically, the term "Network App" is now being used instead of "NetApp," as initially selected in the first period of the project. This update reflects the shortened form of "Network Application" and has been applied consistently across all project's documents and materials.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The current report describes the final prototypes of the three Network Applications, that have been developed within the Employees and Machines (IEM) pillar to support the innovative interaction of employees and machines and is driven by Task 4.2. The report provides details on the development of the final prototype (version 4.1) in terms of technical architecture, features and dependencies, while also utilizing the final version of the tools (SDK, NEF, CAPIF) developed within the EVOLVED-5G framework. Moreover, the report contributes to the testing and evaluation of the use cases, described in the previous deliverable of WP4 (D4.2), through the integration activities that took place both in Malaga and Athens infrastructure focusing on the iterative validation of 5G connectivity and communication between components (5G network <–> Network Applications <-> Vertical Applications).

1.2 STRUCTURE OF THE DOCUMENT

The document is divided into 3 main sections:

- Section 2 "**CONTEXT OF THE PILAR**" echoes D4.2 and presents a summary of the IEM pilar goals, challenges and specificities.
- Section 3 "**FINAL PROTOTYPE OF NETWORK APPLICATIONS**" describes the finalized version of the IEM Network Apps (version 4.1). After a reminder of the NetworkApp use-case(s), it describes the technical architecture, features and dependencies of each Network App.
- Section 4 "**INTEGRATION ACTIVITIES AND USE-CASE TESTING**" presents the two integration rounds for use-case testing performed during the project. The first reported test was performed with intermediate version of Network Apps and components (NEF, CAPIF and SDK) while the second test was performed with final versions of Network Apps and all EVOLVED-5G components.

1.3 TARGET AUDIENCE

The release of the deliverable is public, intending to expose the overall EVOLVED-5G ecosystem and Network Apps progress to a wide variety of research individuals and communities.

From specific to broader, different target audiences for D4.4 are identified as detailed below:

- **Project Consortium**: To validate the fact that all IEM pilar Network Apps have reached their final state. One of the main goals is to document the technical evolution of these Network Apps with respect to the initial vision and use-case.
- **Industry 4.0 and FoF (factories of the future) vertical groups:** To crystallise a common understanding of technologies, and tools that were used for the development of the Network Apps. Besides, it also demonstrates the final architecture and features a Network App can reach. A non-exhaustive list of Industry 4.0-related groups is as follows:

  - Manufacturing industries (including both large and SMEs) and IIoT (Industrial Internet of Things) technology providers.
  - European, national, and regional manufacturing initiatives, including funding programs, 5G-related research projects, public bodies and policy makers.
  - Technology transfer organizations and market-uptake experts, researchers, and individuals.
  - Standardisation Bodies and Open-Source Communities.
  - Industry 4.0 professionals and researchers with technical knowledge and expertise, who have an industrial professional background and work on industry 4.0-related areas. This includes in particular professionals and researchers working on domains related to the Human-Machines cooperation, such as industrial chatbots, remote assistance and digital/physical twins for manufacturing and repair.
  - Industry 4.0 Investors and business angels.
- **Other vertical industries and groups**: To seek impact on other 5G-enabled vertical industries and groups in the long run. Indeed, all the architectural components of the facility are designed to secure interoperability beyond vendor specific implementation and across multiple domains. The same categorization as the above but beyond Industry 4.0 can be of application.
- **The scientific audience, general public and the funding EC Organisation:** To document the work performed and justify the effort reported for the relevant activities. The scientific audience can also get an insight of finalized Network Apps' processes, tools and features.

2 CONTEXT OF THE PILLAR

The emergence of new equipment and technologies such as AI, Digital Twins and Augmented Reality offers promising potential to boost production performance for factories. Nonetheless, Industry 4.0 requires workers to collaborate efficiently with each other, with machines and with digital systems. Human-Computer Interaction (HCI) is thus at the heart of Industry 4.0. The Interaction of Employees and Machine (IEM) pillar reflects this need for factories and manufacturers to facilitate such collaborations.

Within the scope of EVOLVED-5G project, 3 challenges have been identified and each of these challenges is effectively addressed by the three SMEs within Task 4.2:

- Allowing efficient collaboration between remote workers (IMM).
- Supporting the work of employees via autonomous chatbot-driven systems (INF).
- Facilitating verification and certification phases (GMI).

For the IEM pillar, the EVOLVED-5G project represents the opportunity to address the current limitations related to network capacities. On the one hand, traditional network infrastructures and technologies are well-known but often not suitable within factories from a performance or security viewpoint. On the other hand, 5G networking is often more obscure for manufacturers but it offers the performance level and the features required to support Industry 4.0 use-cases. The three Network Apps developed during the project's lifetime, aim at addressing the 3 IEM challenges mentioned above and facilitate the adoption of 5G for other stakeholders interested in Industry 4.0.

The main specificity shared by the three IEM Network Apps is that they are directly linked to the activity of end users, i.e., factory workers. As detailed in Section 3, each partner's use-case is built around an already existing vertical application (vApp) used by workers. Being coupled to such applications means that IEM Network Apps can directly affect in a positive way how human workers will work in future industrial environments.

3 FINAL PROTOTYPE OF NETWORK APPLICATIONS

3.1 Remote Assistance in AR Network App

3.1.1 Use case description

The Network App developed by IMM focuses on the Quality of Service (QoS) of the network during a remote assistance call in Augmented Reality. The first goal of the Network App is to monitor the 5G network Quality of Service (QoS) and notify IMM's vApp -- in real time -- when this QoS changes. This way, both the local worker within the factory and the remote expert helping the worker can be informed of the different degradations of the network and adapt their behavior during the remote assistance call, as illustrated in Figure 1. Letting end-users be aware of the current state of the network is a first step towards limiting confusion and errors during the call, which is crucial to ensure fluid cooperation during the procedure.

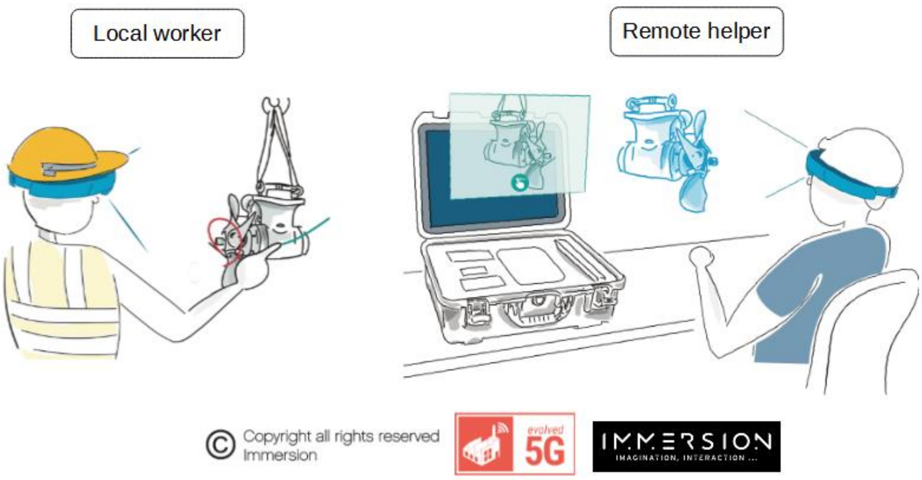


Figure 1: The IMM base scenario. A local worker within the factory is helped by a remote expert thanks to an Augmented Reality based vApp.

However, the main and most promising feature of the IMM Network App goes beyond simple QoS notifications. In fact, the heart of its added value relies on the autonomous service adaptations the Network App can suggest to a vApp. Depending on the current QoS, the IMM Network App sends recommendations about which features of the vApp should be enabled or temporarily disabled. For instance, in case of network congestion due to a high number of users entering a given location, the Network App can autonomously suggest disabling heavy data transfers such as the sending of 3D models or their real-time synchronization between users.

To achieve this goal, several challenges had to be faced during the EVOLVED-5G project:

- Contrary to an increasing number of mobile phones and tablets, autonomous Augmented Reality (AR) headsets are not necessarily compatible with 5G networks. We thus had to make sure the selected AR device (the Hololens 2 headset from Microsoft) could communicate with the Network App through 5G.
- Being able to monitor in real time the QoS of the 5G network. Time-sensitive networking was thus necessary to match the needs of remote assistance in an industrial context.

- Add some “intelligence” into the Network App to suggest the best service adaptations that could be useful for a significant number of Augmented Reality and/or video streaming vertical applications.

3.1.2 Detailed Architecture

While the Network App has the same architectural foundations that its early versions presented in D4.2. It still has its basis built on the EVOLVED-5G-5G SDK, a dedicated Flask server to communicate with the NEF emulator and separated entities to handle QoS and location requests, dynamically generate endpoints, etc.

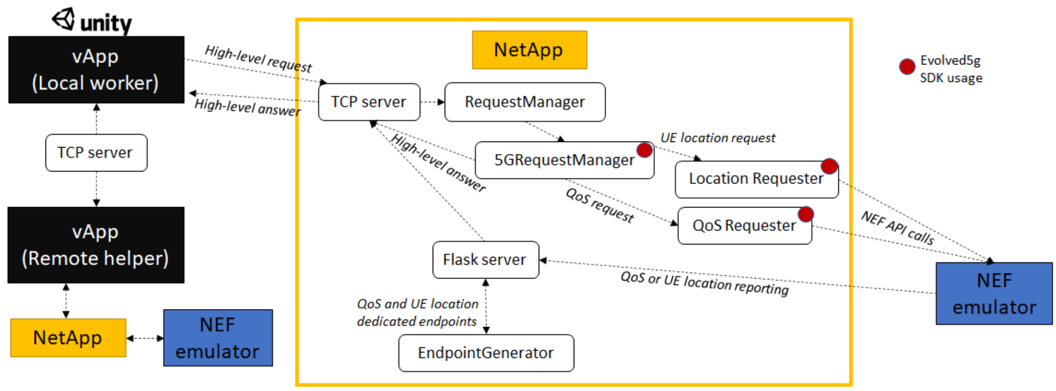


Figure 2: Architecture of the early versions of the IMM Network App from D4.2.

Nonetheless, significant improvements have been made until reaching the current state. First, the Network App now includes a module dedicated to the autonomous generation of service adaptations for the vApp. The *RequestManager* class, which still acts as the central controller of the application, now monitors the User Equipment (UEs) through a specialized *UEController*. This instance handles and stores the data of monitored UEs, including their IP address, location and if the expected QoS can be guaranteed for them or not. Upon receiving a notification from the NEF emulator, the *UEController* updates the monitored UEs data. A Finite State Machine (FSM) for the current general QoS (called *QosFSM*) is also updated.

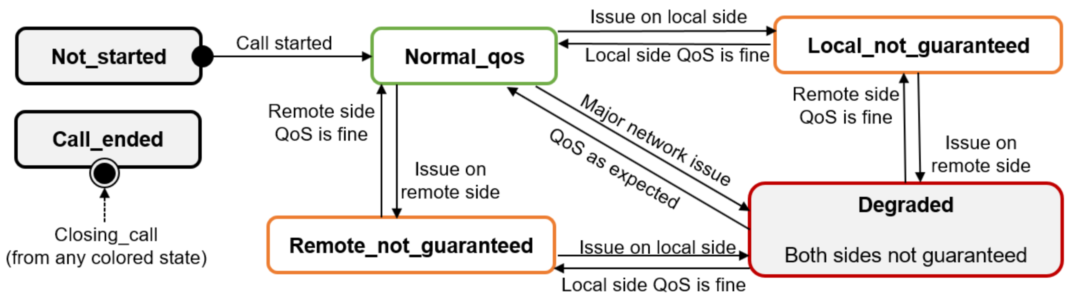


Figure 3: Overview of the QosFSM states and transitions.

Each state transition within the *QosFSM* is associated with one or several service adaptations. These adaptations belong to one of the two possible categories:

1. *Suggested* adaptations are optional propositions to slightly improve the experience for end-users. Vertical applications can choose to safely ignore such adaptations if they prefer not to apply them.

2. *Mandatory* adaptations are decisive and impactful service adaptations that should not be ignored. Of course, the Network App cannot force a vApp to change its behavior. Nonetheless it is still heavily advised to consider the Network App recommendation to adapt available features to the current network QoS.

It is worth noticing that in any case, the service adaptations generated by the Network App are generic enough to be suitable to many contexts and potential vertical applications. An overview of possible adaptations is available in Table 1.

Category	Adaptation	Example of trigger
Video	Increase/Decrease video quality Start/Stop sending video Start/Stop receiving video	Minor QoS degradation Major QoS degradation (ex: too much UEs on both user locations)
Audio	Start/Stop sending audio Start/Stop receiving audio	Critical network failure (preserving audio is top priority for many use-cases)
AR features	Start/Stop all AR Start/Stop AR object synchronization	Major QoS degradation Minor QoS degradation
Base	Reset features to normal usage	QoS reached again the required level

Table 1: List of possible service adaptations that the Network App can provide to a vApp.

Secondly, version 4.1 of the Network App includes a Graphical User Interface (GUI) instead of being only terminal-based. Developed with web technologies, the GUI takes the shape of an interactive web application to monitor a remote assistance call QoS (see Figure 4). Upon starting to monitor a call, the Network App is waiting for a vApp to connect and send requests. The data of monitored UEs is then displayed and updated in real time on the web page. Additional UEs to monitor can also be added or removed manually through the interface. Besides, it is also possible to simulate a vApp connection and basic requests for quick tests and debugging purposes.

From a technical point of view, the web application runs on a dedicated Flask server and can be accessed locally through a common browser (tested on recent versions of Firefox and Google Chrome). The *WebRequestHandler* instance handles the user input action effects, queries the *RequestManager* when needed (for instance, to trigger request to NEF), then updates the current webpage through Javascript functions.

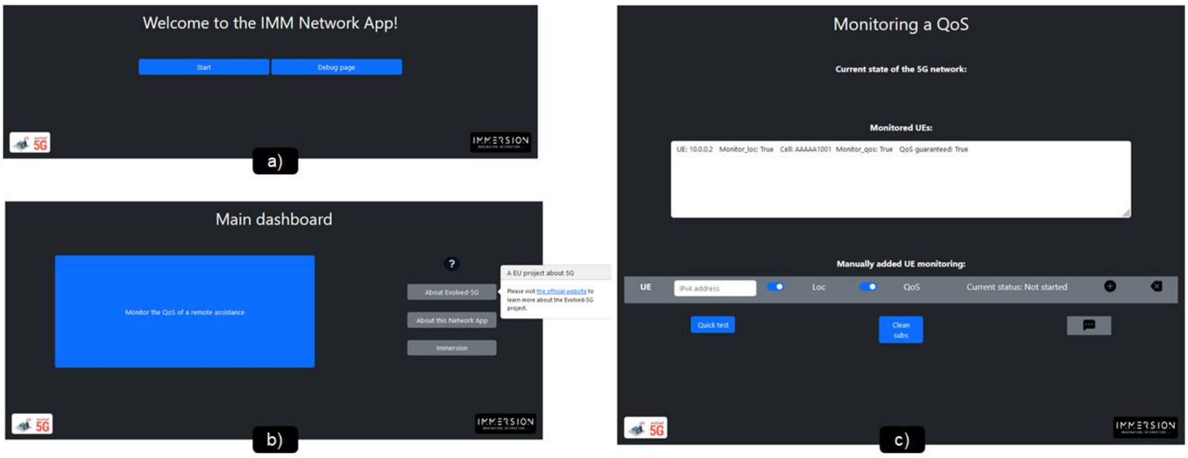


Figure 4: Overview of the web GUI of the IMM Network App.

Finally, the Network App now registers to CAPIF to access NEF. The whole Network App is containerized through Docker and was deployed on UMA's premises, as detailed in Section 4. Figure 5 shows the architecture of the last version of the Network App.

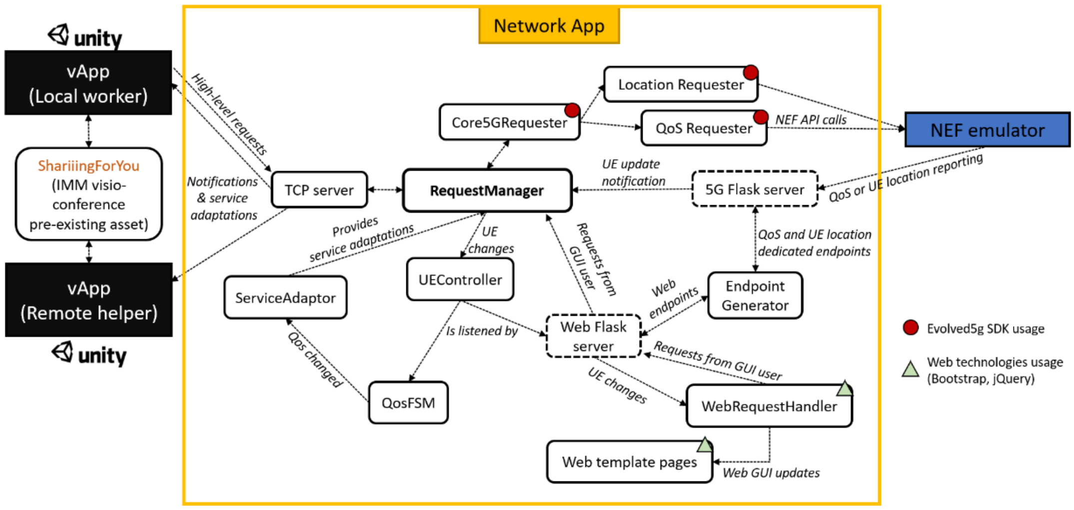


Figure 5: Final architecture of the IMM Network App.

3.1.3 Additional dependencies

This QosFSM was implemented using the *statemachine* Python package. The Network App GUI was developed using Bootstrap and JQuery.

3.2 CHATBOT ASSISTANT NETWORK APP

3.2.1 Use case description

The main goal of the target use case for the Chatbot Assistance Network App, within the scope of EVOLVED-5G Project, is to establish a dedicated series of actions that will take place in a factory environment to facilitate the handling of maintenance scenarios via a chatbot platform. The use case scenario is described in detail in the following template. Additionally, the use case schema shown in Figure 6 explains the sequence of actions that has been identified, while also demonstrating the interaction between the 5GS, the vApp, and the Network App.

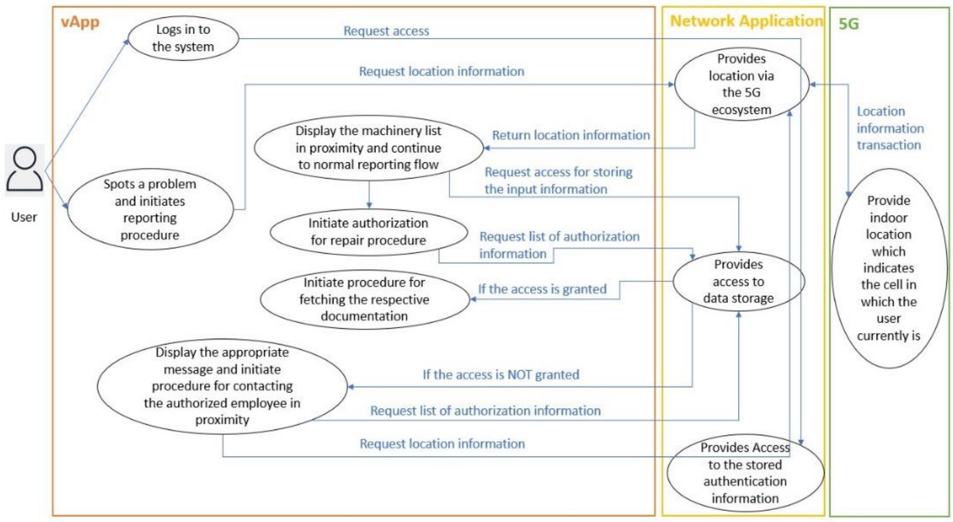


Figure 6: INF use-case schema.

This use case focuses on maintenance scenarios within a factory environment. The goal is to introduce a chatbot application for reporting malfunctions and facilitating the necessary actions to address and fix the reported issues. The Network App developed to support this Use Case utilises location services, providing helpful information about the specific area to the user, user authentication restrict access to unauthorized personnel to specific areas, and offers relative material or manuals about the equipment in those. Also, notifies available specialized technicians to solve a pending issue. Additionally, the INF web-based chatbot platform has been modified to support and utilize the features provided from the described Network App.

Having a deployment of a 5G non-public network within the factory environment, a Network App will enable the use of a chatbot to help identify and solve possible malfunctions in a shorter time frame using a more user-friendly solution. Also, the 5G network can provide an ID for all the connected workers of the factory and their relative location. Workers can be connected from any device. The Network Application can match the location to a predefined factory area (block or cell). It will also have a list with ranks and access of the employees so it can give a warning if a person is in a dangerous area (e.g., toxic exposure in the area) regardless the access status or notification for restricted access or even provide material relative with the area if the person has work to do there.

All workers have a mobile phone, or a tablet connected to the 5G network of the factory. If a worker encounters faulty equipment, they can use the vApp to report the issue. Firstly, the Network App retrieves the worker's location, such as Area A. The first step is to check if area A is safe for any worker to be present. If there is an alert or warning for the area, a notification will be issued so an evacuation can take place in a quick and safe way always with health care in mind. If the area is safe, a second check is conducted to verify the worker's access status of Area A. In case the worker should not be in the specific area he/she is prompted to leave. If the worker is cleared, a request for the manuals of the area can be made and further instructions to fix the damaged machinery. Finally, if the problem remains unresolved, the closest specialized technician will be notified to aid. Controlling accessibility to the areas and providing easy assistance in a friendly automated way are some of the benefits of this use case.

The Network Application's reusability makes it an attractive proposition for SMEs. The presented use-case demonstrates its versatility across different business scenarios, offering a comprehensive understanding of its implementation. This not only simplifies integration into existing setups but also empowers SMEs to leverage the Network Application for various purposes beyond its initial deployment. The adaptability and flexibility of the Network Application enables other SMEs to tailor it to their specific requirements and chatbot systems in order to explore new opportunities for their business growth. More precisely, an SME that requires a location Network Application in order to advance the provided functionalities of its services, could reuse the specific location Network Application in order to couple it with its vApp and advance the provided services.

In terms of benefits, the Industry 4.0 App developer can benefit by the location Network Application for the evolution of the vApp in order to support additional features and make it 5G-enabled. The Integrator can benefit from this use case by understanding the proposed solution and using it to create a new product that can be licensed and sold to another stakeholder. An equipment vendor, such as a private 5G vendor and/or a mobile UE vendor for industrial spaces, could be benefited from the enhanced chatbot service for Industry 4.0 environments, and include the provided chatbot service and Network Application as a pre-installed feature. The MNOs could benefit by including the location Network Application in their marketplaces to support the development of other vertical Apps that require location info for their operation. The testbed operator could perform performance experimentation on the Network Application and vApp coupling by assessing the performance resilience and location accuracy under various reception conditions.

Regarding the current status of the situation, the web-based platforms usually used within factory environment to support the maintenance services, do not have access to location information within an indoor facility. This limitation makes to impossible to implement the described use case for the following reasons:

- Chatbot is unable to provide a detailed list of the machinery in close proximity.
- The Chatbot cannot notify the appropriate technicians who are closer to the reported problem in order to effectively resolve any evolving issue.

By utilizing the EVOLVED-5G Innovations, a Network Application with access to location and verification services can be developed solving not only the forementioned issues but also addressing the smooth operation of the factory workflow and the workers' safety as described in the Detailed Description section of the Use Case.
In terms of evaluation, the criteria include accurate location provision, the authentication process, and location classification, along with the identification of authorized workers in close proximity. One key challenge for the Network Application used in this Use Case is ensuring the accuracy of the location provided by the 5GS, as it is essential for creating the factory grid. Additionally, a comprehensive list of necessary information is required to facilitate granting

access and providing the respective materials to address any pending issues. Overcoming these considerations and burdens was a crucial factor towards the successful implementation and functionality of the Chatbot Assistance Network App within the EVOLVED-5G project.

3.2.2 Detailed Architecture

The general architecture reflecting the interaction between the components of the identified chatbot and their mapping with the use case, can be seen in Figure 7.

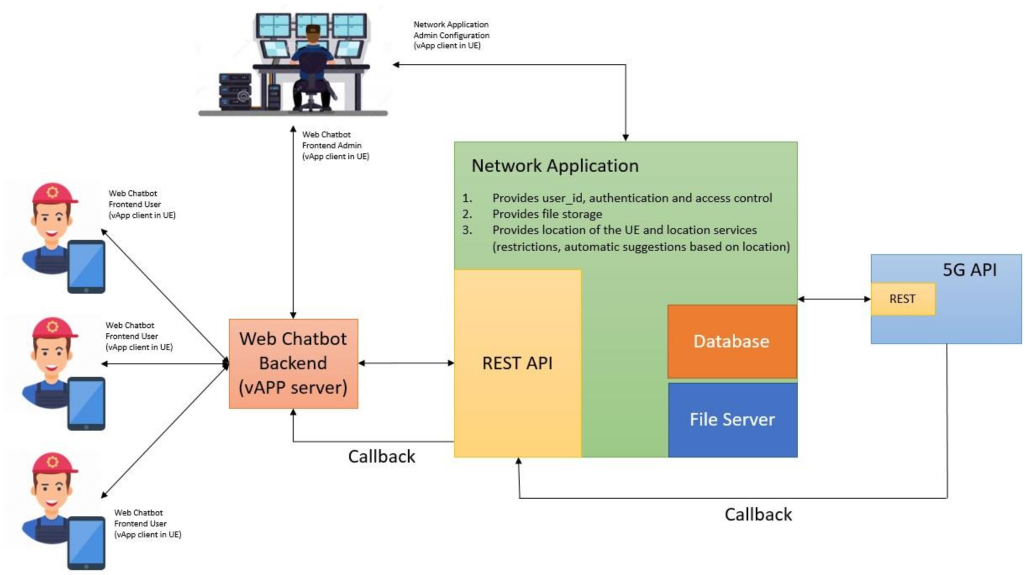


Figure 7: INF Architecture schema.

In a more detailed explanation, the architecture of the Network App includes the following elements:

- The FastAPI Server, responsible for exposing the necessary APIs used for interaction between the vApp and Network Application. Additionally, it can return user location or required machine files based on the API call.
- The Evolved5G SDK, facilitating communication with the NEF.
- The Database and File Server, which stores administrative information crucial for the realization of the use case scenario.
- The Web Admin GUI at Figure 8, serves as the user interface for managing administrative information and interacting with the database and file server.

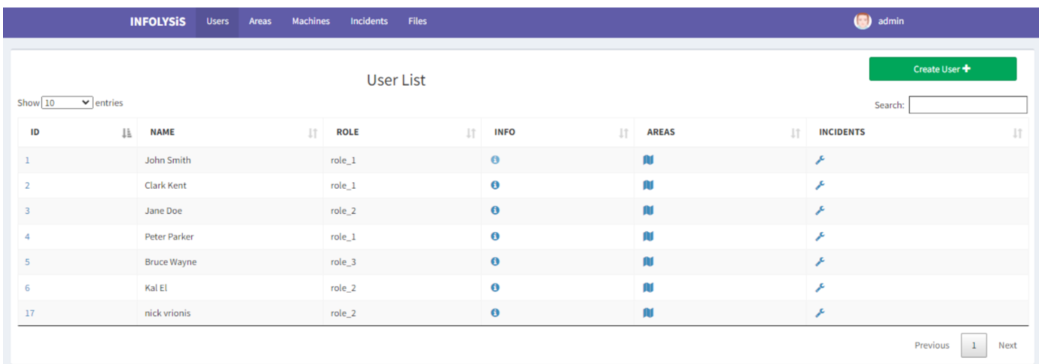
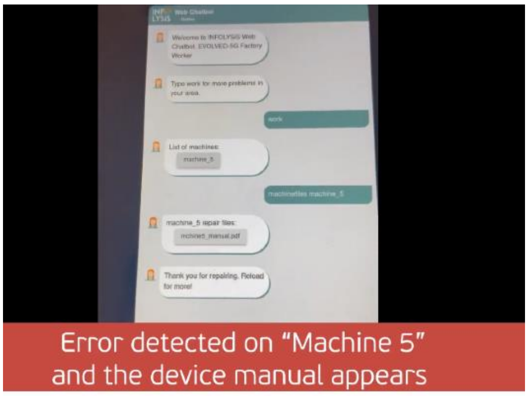


Figure 8: Administrative Platform.

When the user is logged in, the process is initiated by entering the word "work" in the chatbot. The following sequence of steps is then executed:

- The command is sent to the Web Chatbot Backend.
- The Web Chatbot Backend calls the appropriate endpoint of NetApp (http://localhost:8000/AreaMachines/) while sharing the relevant worker ID.
- Network Application (since it is registered with CAPIF), through the mentioned endpoint, communicates with the NEF Emulator using the Evolved5G SDK and retrieves the cell ID corresponding to the worker's indoor location.
- Utilizing the obtained cell ID, Network Application accesses the database to retrieve a list of machines located in that cell, flagged as malfunctioning, and returns it to the vApp. The list is displayed on the chatbot screen as buttons for the user to select from.
- The user chooses a desired machine by pressing the associated button, transmitting the respective machine ID to the Web Chatbot Backend.
- The Web Chatbot Backend then calls the appropriate endpoint of NetApp (http://localhost:8000/MachineFiles/) while sharing the relevant machine ID.
- Based on the machine ID, Network App accesses the database, retrieves the files associated with the specific machine, and returns them as a list to the vApp. The list is displayed on the chatbot screen as buttons corresponding to the machine list mentioned earlier.
- Finally, the user can view each file by selecting the respective linked button. An example of the results after the completion of the final step can be seen in Figure 9.



Machine 5 Repair Manual

1. Power off the machine and disconnect all power sources.
2. Remove all external components and check for any visible damage.
3. Thoroughly inspect all internal components and note any abnormalities.
4. Replace any broken or damaged parts as required.
5. Reassemble the machine and reconnect all power sources.
6. Test the machine to ensure all components are functioning correctly.
7. Adjust the machine settings to meet production requirements.
8. Run a diagnostic test to check for any further issues.
9. Perform a final inspection of the machine before reinserting it into the production line.
10. Monitor the machine's output to ensure it is running correctly.

Figure 9: Web chatbot

3.2.3 Additional dependencies

The successful operation of the Network Application relies on hardware and software dependencies as follows:

Hardware dependencies:

1. **Server or Hosting Environment**: Network Application is typically deployed on servers or hosting environments capable of running Python applications. The minimum recommended specifications for the server or hosting environment are as follows:
    - Processor: Intel Core i3 or equivalent
    - RAM: 4 GB or higher
    - Storage: 20 GB or higher
2. **Network Connectivity**: The server or hosting environment should have a stable and reliable internet connection. Network Application handle HTTP requests and responses, so network connectivity is essential for the proper functioning of the project. Furthermore, it would be necessary to have CAPIF and NEF in place for the operation to be successful.

Software dependencies:

1. **Database:** MySQL MariaDB
2. **Apache web server:** For the hosting and the file server
3. **Python:** Also, our Network Application relies on several essential Python libraries like SQLAlchemy and Pydantic to deliver powerful functionality and ensure smooth operations.
4. **PHP:** For the Web Admin GUI

3.3 Digital/Physical twin Network Application

3.3.1 Use case description

When a bonded composite repair is performed, all repair data (temperature, humidity, vacuum level etc.) are recorded, in order to certify that the overall process has been performed according to specifications and confirm the physical and mechanical properties of the repaired part (especially the composite patch and the adhesive bond). However, several repairs take place "on-wing" and remotely (maybe even outside of hangars) at challenging environmental conditions, due to geographical location (extremely low temperature, increased humidity, very high altitude etc.). In addition, increased geometrical complexity of contemporary all-composite aircraft (e.g., A350, B787) may lead to extensive Temperature variations during curing, well beyond specified limits (usually +/-5oC), which may affect the curing degree and / or the mechanical properties of the produced repair. This may lead to ambiguities on the evaluation of the repair results and subsequently delay or even prohibit the authorization of aircraft to resume flight operations, especially when repairs on safety critical structures are performed. More specifically:

- **Physical Repair Twin:** A setup using the same materials and subject to exactly the same environmental conditions in terms of ambient temperature and humidity during layup will be prepared. The patch will be then cured simultaneously to the actual repair, using appropriately selected portion of real-time transmitted data (e.g. lagging thermocouple), to enable imminent destructive and/or non-destructive testing of produced material at the Engineering Centre laboratories after the end of the curing process. Physical Twin's curing could be performed either by applying a classical vacuum bagging setup with an ANITA EZ, or using appropriate thermal press, equipped with dielectric sensors for curing degree monitoring and / or other sensors, as deemed necessary. The latter setup could additionally provide real time DoC and other data, equally valid for the remotely performed structural repair on the aircraft.

- **Digital Repair Twin**: Using material simulation software, the Degree of Curing (DoC) of the specific (i.e. the one applied on the actual aircraft) resin and adhesive will be calculated, according to the data received on-line from the ANITA EZ performing the repair, in terms of Temperature and Vacuum conditions. Such calculations could be performed several times in parallel, according to the results received by EACH T/C, ensuring quality of the part at every point of measurement. In case of deviation of calculated DoC from expected % level, the ANITA EZ will be instructed (either on or off-line) to extend the curing process by the required duration, until the appropriate DoC values are reached. Calculation of expected porosity level, as a combination of Temperature, Vacuum and environmental Humidity data during layup, could be also enabled. In a second step, the Digital Twin routine could be included in the ANITA EZ software, so as to run in situ.

Within the frame of EVOLVED-5G the ANITA hot bonder(s) used for repair curing will be connected to the 5G network at the repair area, to transmit in real-time all related data to the Engineering Centre of aircraft manufacturer / airline / MRO. This will help GMI to perform a technological leap in the face of emerging competitors by providing innovative solutions, adapted to the specific aircraft requirements, not available yet on a global scale. It will help in optimizing the integration of systems in the airframe along with the validation of important structural advances and to make progress on the production efficiency and manufacturing of structures. Solutions will assist in avoiding part scraping during manufacturing, as well as in MROs, airlines and composite plants, by increasing the range of application of bonded composite repairs.

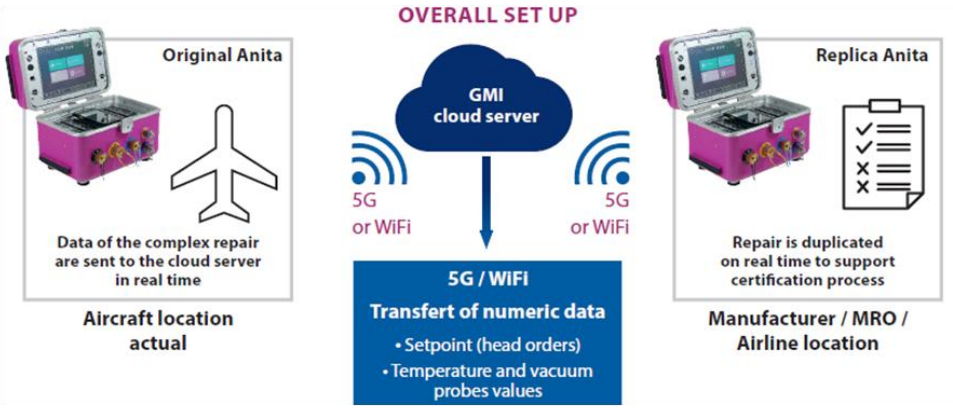


Figure 10: Overall GMI use-case setup.

In addition, application of Physical / Digital Repair Twins is expected to directly reduce structural composite repair certification costs and increase aircraft availability, while assisting in continuous airworthiness of commercial aircraft fleet.

Check communications between end-users' devices - GMI use case 1:

The first use-case for GMI involves a scenario where both end-users (one on the primary Anita, the second on the secondary Anita) need to initialize secured communication necessary for the realization of the twin repair. Two major outcomes emerge from this use case: 1) checking that a satisfying QoS can be reached and 2) checking that authentication procedures are respected to ensure the security of communications. Concerning the Current Status, the Anita HMI should be adapted to the authentication process and the communication settings. Additionally, Anita cloud system and database are already deployed and in the context of EVOLVED-5G Innovations, the goal is to enable sufficient QoS to be achieved through 5G infrastructure, and to also benefit from authentication services that may be offered. As far as the Evaluation criteria are concerned, the KPI: Quality of connection is of crucial importance. The expected outcomes include the check whether the Network Application can allow both vApp to communicate. Additional prerequisites, constraints, restrictions involve the Anita computer actually uses a Wi-Fi antenna for Internet access, or it may be necessary to use a dongle to reach 5G. The target location has been selected to be the 5GENESIS Malaga (Indoor only).

Retrieve UE location information - GMI use case 2:

Concerning the second contribution of GMI entitled:" Retrieve UE location information", the objective is to retrieve information about the geographical location of both devices: primary and secondary Anita. Moreover, for any type of composite repair, it is important to document the process as much as possible. In the case of the twin repair, geographic location information will be essential and should be included in the curing report. This information, gathered through the location service, can be used to obtain other data regarding climatic and environmental conditions, always with the goal of documenting the repair as much as possible. The user can also enter other information manually in the location and environment category. Finally, the expected outcomes are to obtain and store the geographic location information for both devices involved in the twin repair. Concerning the current status, currently there is no notion of location on the Anita device. On the other hand, the curing cycle report exists, and will have to be enriched with these last data. Concerning the EVOLVED-5G Innovations, GMI is interested in accurate location information that will provide a real benefit to repair documentation.

Start repair on primary Anita, retrieve and process data on secondary Anita – GMI use case 3:

The third contribution of GMI as far as the use case is concerned consists in ensuring that the communication will be maintained during the process. The name of the use case is "Start repair on primary Anita, retrieve and process data on secondary Anita. This third and final step is to initiate the repair cycle. It is essential that the data be transmitted continuously to ensure the twin repair process. On the primary Anita side, the probe data and the setpoint instruction must be sent. On the other hand, on the secondary Anita, the data has to be read and then evaluated. Either to apply the same setpoint on the sample repair in real time, or to store the data for a study by a thermal study software. The setpoint instruction must be refreshed, received and processed by the secondary Anita, every 10 to 20 seconds maximum to ensure proper twin repair. The transfer time of the data packet must respect this refreshment constraint. Concerning the expected outcomes, points of interest are both the verification if a satisfying

QoS can be reached during the entire process, as well as the data integrity. Currently the status is that we have the possibility to transfer the baking data to our cloud database. This data is evaluated by web applications. It will be necessary to develop a reading module that will be used in the case of a slave Anita.

3.3.2 Detailed Architecture

The Network Application uses the Python language and more precisely the FastAPI framework. This application receives requests from the Anita HMI application (vApp), and then proceeds to the interactions with NEF services, to return desired results. These results are passed to the vApp using JSON format.

The Network App and the NEF emulator run on a specific machine, while the vApp runs on an embedded computer inside the Anita machine. This computer offers a 5G connection possibility by using a specific add-on card. The final prototype of the Network App (v4.1) effectively uses the latest versions of the SDK and NEF emulator. During the deployment process, the application is also able to register and integrate with the latest release of CAPIF services.

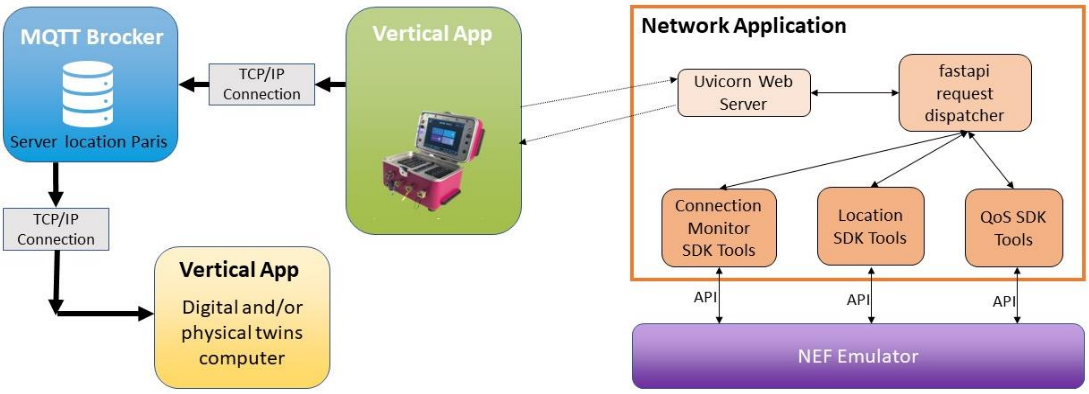


Figure 11: Interaction between the different software units

From an operational point of view, there are two main functions with the Network App, which enable the vApp to improve its repair process:

1) User Equipment (UE) location information. The objective is to retrieve information about the geographical location primary Anita. For any type of composite repair, it is important to document the process as much as possible. In the case of the twin repair, geographic location information will be essential and should be included in the curing report. This information, gathered through the location service, can be used to obtain other data regarding climatic and environmental conditions, always with the goal of documenting the repair as much as possible. The expected outcomes are to obtain and store the geographic location information for both devices involved in the twin repair.
For the moment, the information returned is displayed on the device's main menu, as well as in the final cure report:

Figure 12: Start menu display.

2) Then, the "QoS for transfers between end-users' devices" function. The purpose of the function is to adjust the amount of data transmitted between the two components of the digital twin based on the received QoS information from the 5G core. The goal is to optimize data transmission by adapting to the network conditions. If the connection is degraded and the quality of service (QoS) is insufficient, the function will apply a minimum data packet size to be transmitted, ensuring the availability of at least one minimum packet. This ensures some continuity in the exchange of information, even under poor network conditions. On the other hand, if the QoS is deemed sufficient, the function allows for the transmission of all the data, ensuring smooth and complete communication between the components of the digital twin.

QoS Type	Content of the data package
Below standard = Minimum requirements	Time Coldest thermocouple value
Satisfactory = Full data package	Time Theorical setpoint 24 Thermocouple lines values Hottest line & coldest line 2 vacuum lines values 2 Heating % values

For the time being, the QoS information is displayed on the main page of the vApp application and on a "Debug" page. In addition, the data transmitted on the MQTT broker is adapted as shown in the table above.

Figure 13: QoS display on main menu.

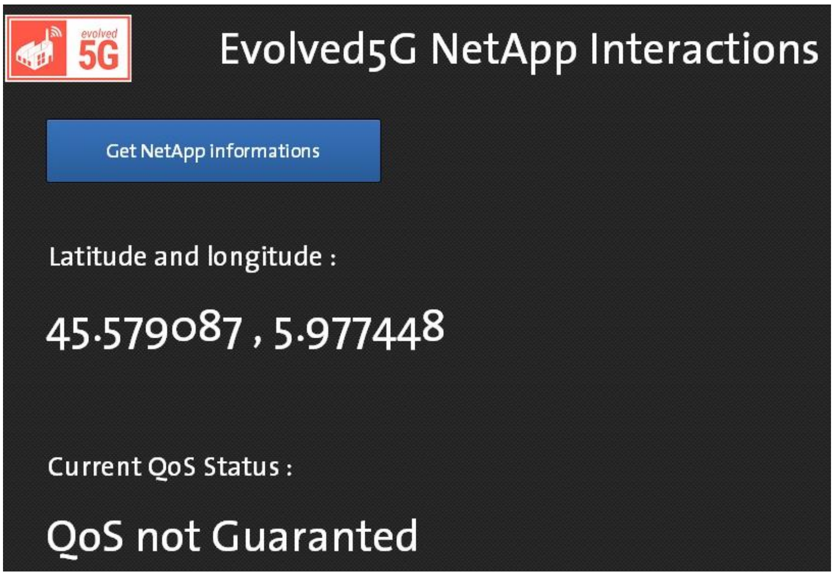


Figure 14: Debug Network page communication.

Moreover, a web page was designed to simulate the vApp and make calls to the Network App for testing purposes when it is not feasible to use the actual device. This web-based simulation allows users to emulate the behaviour and functionality of the vertical application, replicating the interactions and data exchanges that would occur between the application and the Network App in a real-world scenario. By using the web page, developers and testers can thoroughly evaluate the performance, compatibility, and functionality of the application without relying on a physical device. It provides a convenient and efficient way to validate the application's behaviour and ensure its proper integration with the Network App, even when direct device access is not possible.

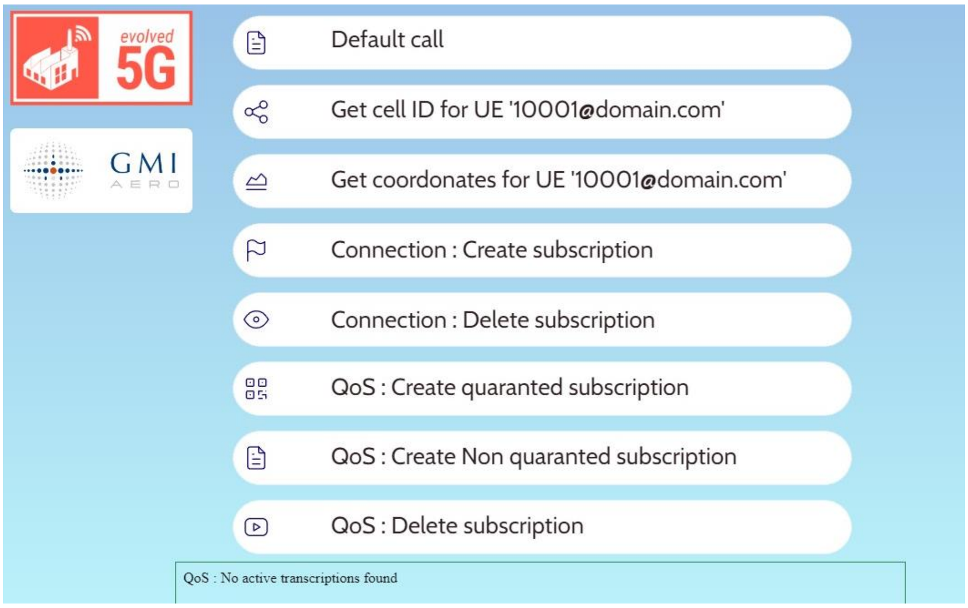


Figure 15: Overview of the web page simulating vApp interactions.

4 INTEGRATION ACTIVITIES AND USE CASE TESTING

4.1 Purpose Of the Integration Tests (1st Round)

The primary objective of these integrations was threefold:

- To verify the connectivity of 5G, specifically the connection between vApps and either Demokritos' or UMA's 5G network.
- To assess the overall communication among CAPIF, NEF, Network Applications, and vApp, as well as the 5G connectivity with the cloud infrastructure.
- To enable conducting real-world testing of the EVOLVED-5G use cases and validating the communication across all components.

Among the Network Apps within the IEM pillar, the Network App Remote Assistance AR Network App has been tested in Malaga's infrastructure. Meanwhile, the Digital/Physical twin Network App and the Chatbot Assistance Network App have been tested in Athen's infrastructure.

In the initial phase of the integration testing, the following components were utilized:

- Network Applications v3.0
- NEF v1.6.2
- CAPIF v2
- SDK v0.8.7.

4.2 Topology and Setup

4.2.1 Remote assistance in AR Network App

The first step to overcome in the IMM use case was to ensure the 5G connectivity of the Augmented Reality device. Most existing headsets are compatible with Wi-Fi 5 or even 6, but having 5G connectivity on such devices is much less frequent. IMM selected the Hololens 2 headset from Microsoft for two reasons. First, IMM has extensive experience with this device and has existing software applications and assets targeting it. Secondly, a recent update (2021) on that device added 5G connectivity as an experimental feature. The Hololens2 headset was successfully connected to UMA's 5G network by using a 5G enabled Android smartphone acting as hotspot, as shown in Figure 16.

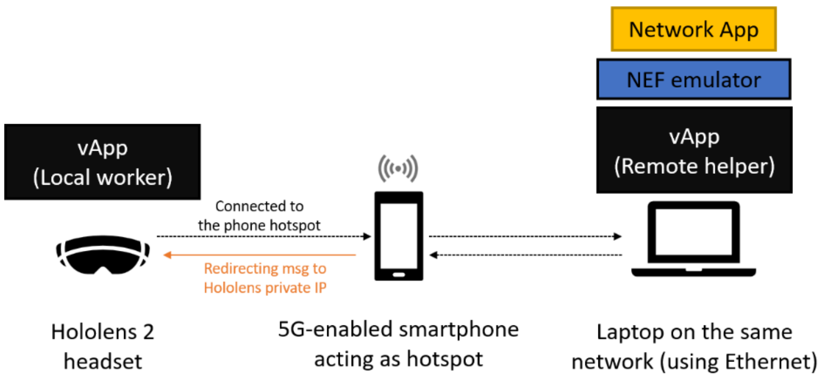


Figure 16: Setup for the 1st integration round of IMM.

The scenario used to test the connectivity between the IMM Network App and the vApp instances reflected a remote assistance scenario. A specific scenario was designed within the NEF emulator web interface. The first UE in a given cell represented the local technician while a second UE in another location (i.e., in another cell) represented the remote expert. On each side, an additional "perturbator" UE was added. The goal of perturbators was to trigger occasional QoS events by entering and exiting the cells of the other UEs (Technician and Expert UEs).

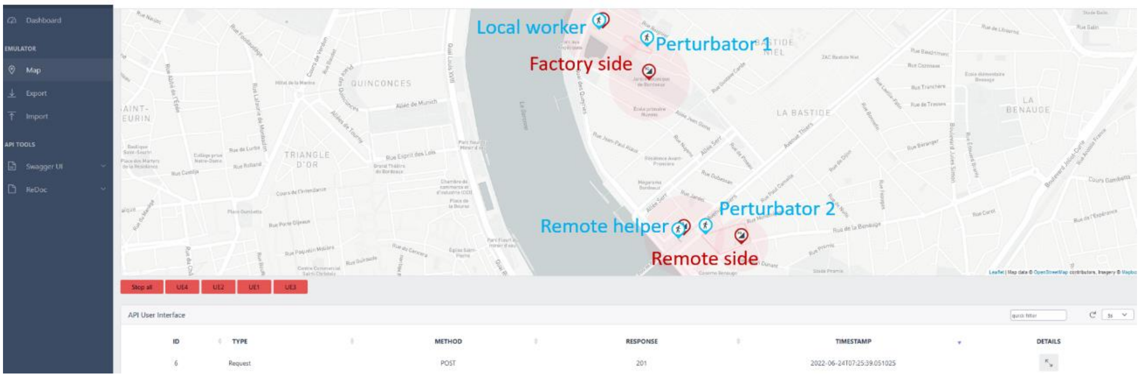


Figure 17: The IMM scenario within the NEF emulator web interface.

Both NEF and CAPIF instances were running locally through Docker Desktop on the laptop. Similarly, the Network App was running locally within its own Docker container.

4.2.2 Chatbot assistant Network App

The first round of integration was conducted at the facilities of Demokritos campus in order to check the 5G connectivity between Network Application and vApp. For the testing, Demokritos Openstack Cloud Infrastructure along with 5G Amarisoft to evaluate use case, has been utilised. The topology infrastructure is depicted in Figure 18, where CAPIF, NEF and INFOLYSiS Network Application are deployed on the OpenStack Cloud Infrastructure and User with the Chatbot (vApp) on a mobile phone with a 5G SIM-card can establish connection to the 5G network via 5G Amarisoft, facilitating the transfer of information within OpenStack environment.

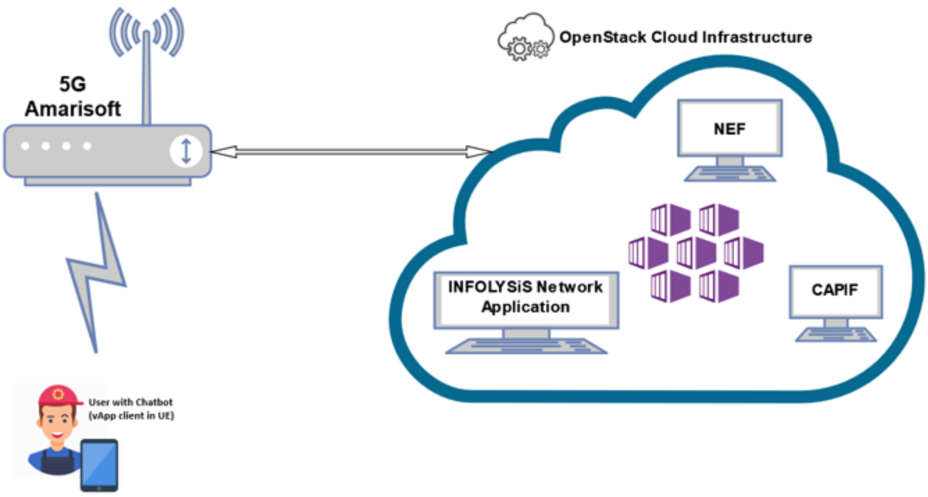


Figure 18: Infrastructure Topology.

The scenario that has been used to test the connectivity between vApp and INFOLYSiS Network App, as well as their functionality, is illustrated in Figure 19. Within the given scenario, there were four cell areas, and among them, one area had an issue with a malfunctioning machine. Additionally, there were three UE with varying permissions. In conclusion, it was verified that the UE without proper permission was unable to detect any issues within the specified cell area. Conversely, UEs with the necessary permissions were able to identify problems in the area and request manual assistance to rectify the malfunctioning machine.

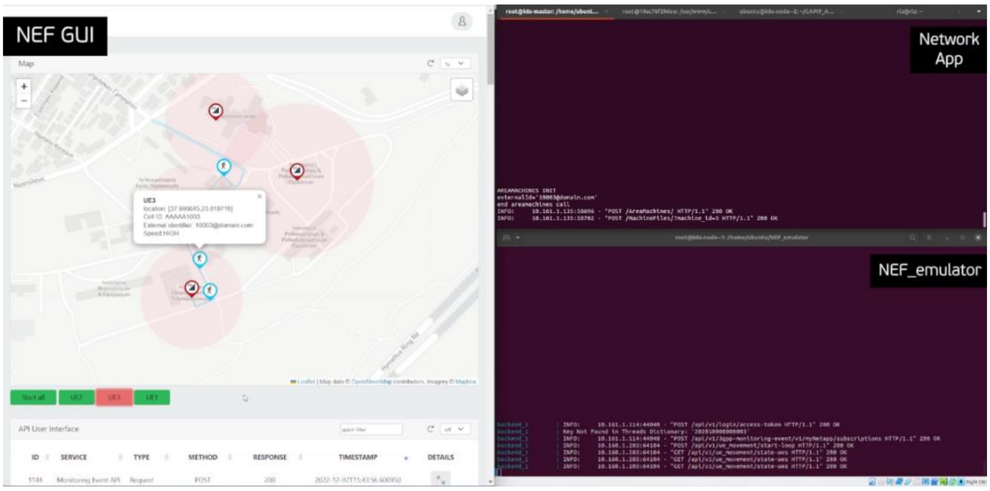


Figure 19: The INF scenario within the NEF emulator.

4.2.3 Digital/Physical twin Network App

The aim of the first integration session was to test the first version of the containerized network application, including communication with the NEF emulator and CAPIF services. The trial was conducted on the premises of Demokritos University in Athens. The UE, the Anita bonding console, was brought to the university's premises to test communication between the vApp, the console's user interface, and the network application using a 5G interface. The Anita console is natively equipped to connect via Wi-Fi. It was therefore easy to use the 5G CPE equipment proposed by Demokritos, which consists of converting 5G connectivity to Wi-Fi. The 5G connectivity was provided by the Amarisoft equipment.

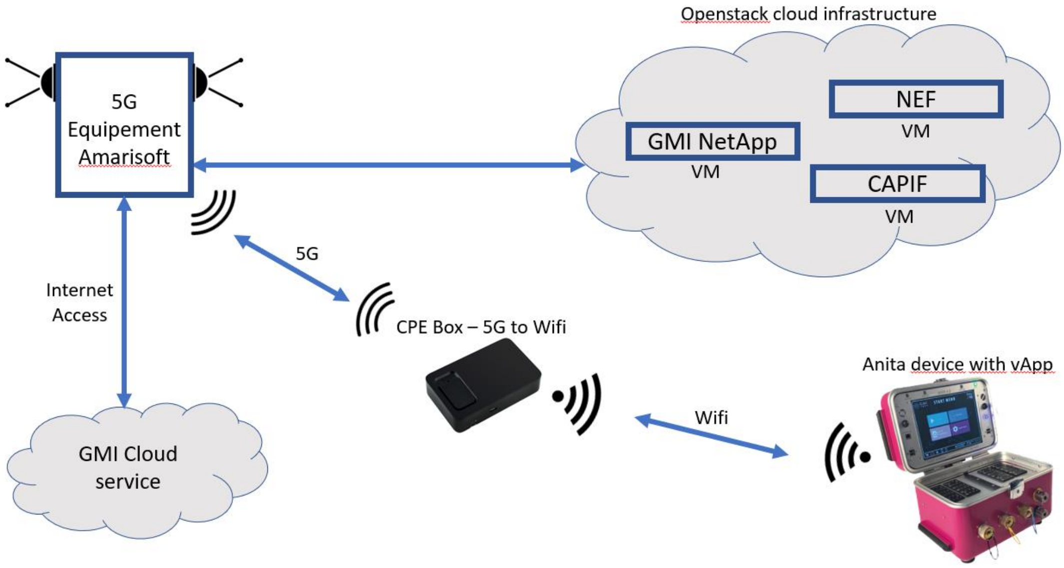


Figure 20: The GMI setup for first integration test round.

Regarding the use case, the objective of this integration was to demonstrate a good communication between vApp and Network application, the ability to request 5G services to retrieve quality of service information, as well as the geographical coordinates of the cell in which the UE is located in order to document the composite repair report.

In this first integration context, the quality of service is simply displayed on the main page of the user application, and there was no adaptation of the size of the data packet at this stage.

For the geographical coordinates, this information was also relayed on the main page, but also integrated into the repair report, as shown in Figure 22.

Figure 21: Main menu.

Page 1/ 4	REPAIR BONDING CYCLE REPORT ANITA 4.0 - Zone 1				
ID Cycle	60°C-20min-12/07/22-10h53			Operator	
ID Recipe	60 °C - 20 min			Report Creation	Printed on 08 déc
Start At	07 déc. 2022 10:53	End At	07 déc. 2022 11:30	Duration	37 mn
Console S/N	A4N003	Calibration Date	29 mars 2022		
5G Localization	Lat : 45.579087 , Lon : 5.977448	Software Version	3.29 - 0.5		
T/C Channel	1-3-5			Vacuum Line	NU

Figure 22: Report of curing cycle.

Figure 23: Data package visualization on distant server.

4.3 Results and Takeways

4.3.1 Remote assistance in AR Network App

The first IMM integration test was successful as the 5G connectivity of all components was validated. The Hololens device was successfully connected to the 5G network through the mobile phone hotspot. This way, the vApp instance for the AR user was able to communicate with the Network App and receive UE location and QoS notifications. These high-level messages were displayed on a virtual panel in Augmented Reality, as shown in Figure 24. This way, end users were notified of both 1) the movements of perturbator UEs and 2) if the QoS was guaranteed or not for the other user or themselves.

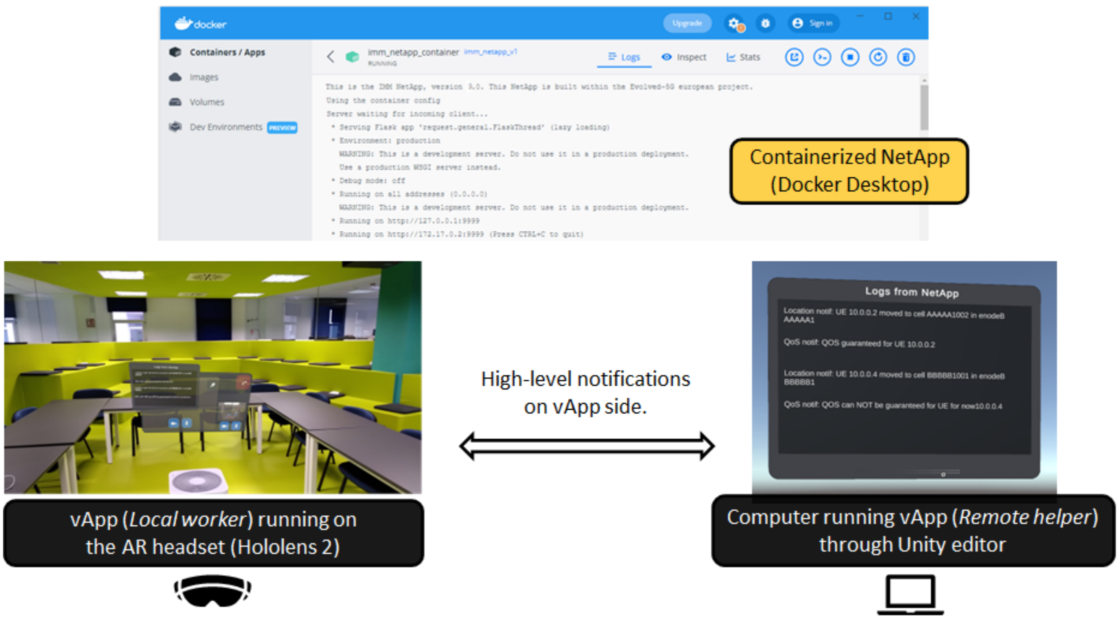


Figure 24: Overview of the first round of the IMM integration test at Malaga. Both vApp users receive notifications from the Network App.

4.3.2 Chatbot assistant Network App

Firstly, the connection between vApp and the 5G network was successfully established. Subsequently, comprehensive testing of the use case was conducted at the Demokritos campus, which confirmed its functionality in a real-world environment and also ensured effective communication between the Vertical Application (Chatbot), Network Application, NEF and CAPIF. Figure 25 displays instances taken during the tests conducted at the Athens Platform, showcasing the use case in action.

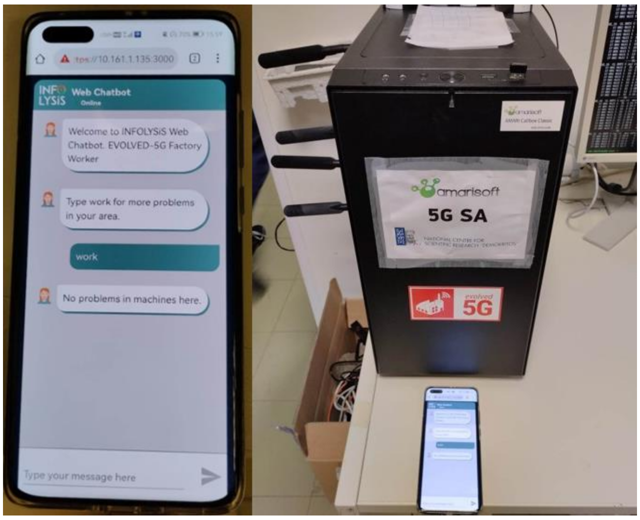


Figure 25: Use case test at Athens Platform (1st round).

4.3.3 Digital/Physical twin Network App

This first round validated the fact that the console can connect via Wi-Fi to 5G equipment and communicate with a service provider such as the Network App. The first changes to the Anita console user application show how well the services offered by the 5G network are being integrated.

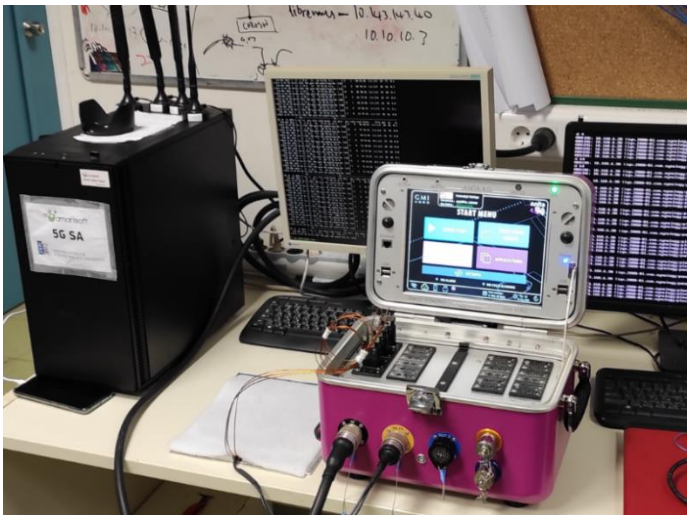


Figure 26: Test architecture for GMI 1st round.

Two of our use cases were correctly validated at the end of this first round of integration:

- First of all, the QoS information is correctly retrieved and displayed (on the main page in particular). In a second stage, the behaviour of the vApp will need to be adapted depending on the QoS.
- To continue, the geographical position of the equipment has been correctly received and displayed on the main page and at the start of the repair. In addition, this information now appears in the cure report, providing essential information that was previously unavailable.

The second round of integration tests will involve improving the integration and the usage of the results obtained via these services, to obtain all the information required for repair in a digital twin situation, and as mentioned above, adapt certain vApp operations according to the results obtained.

4.4 Purpose Of the Integration Tests (2ND Round)

The purpose of the second integration round was to validate the use-cases with the final version of components of EVOLVED-5G. On the one hand, NEF, CAPIF, TSN (Time-Sensitive Networking) and the SDK were enriched with additional features. On the other hand, SMEs also finalized their Network Apps by expanding the 3.0 version and using the last versions of NEF, CAPIF, TSN and SDK. The final prototype (v4.1) of the Network Apps was also completed and used the validation pipeline before the integration tests took place. Finally, the Networks Apps were deployed in Kubernetes clusters in either Athens or UMA’s premises instead of using Docker containers running locally.

It's worth noting that until the end of WP3, it was deemed necessary for the SDK to undergo some minor improvements, primarily aimed at enhancing functionality and addressing specific bugs. During this second integration round, the final version of the EVOLVED-5G components was utilized:

- Network Applications v4.1
- NEF v2.2.2
- CAPIF v3.1.2
- SDK v1.0.8
- TSN 1.2.1

4.5 Topology and Setup

4.5.1 Remote assistance in AR Network App

The second IMM setup was pretty similar to the first one. Once again, the Hololens 2 headset was connected to the hotspot of a 5G-enabled smartphone. The vApp of the *Remote Helper* was still running on a nearby laptop on the same 5G network, (both Wi-Fi and Ethernet connection were tested and working properly). Besides, the IMM Network App, NEF and CAPIF were both deployed in Kubernetes clusters on UMA’s premises, as shown in Figure 27.

More precisely, the Network App was deployed as a *LoadBalancer* service to make sure some given ports were accessible from outside the cluster. This was used 1) to access the Network App web interface from a browser (on the laptop) and 2) to make sure the vApp could establish a connection with the NetworkApp through its TCP server. This way, the external IP of the NetworkApp was the only element which required to be entered inside the vApp to make it work.

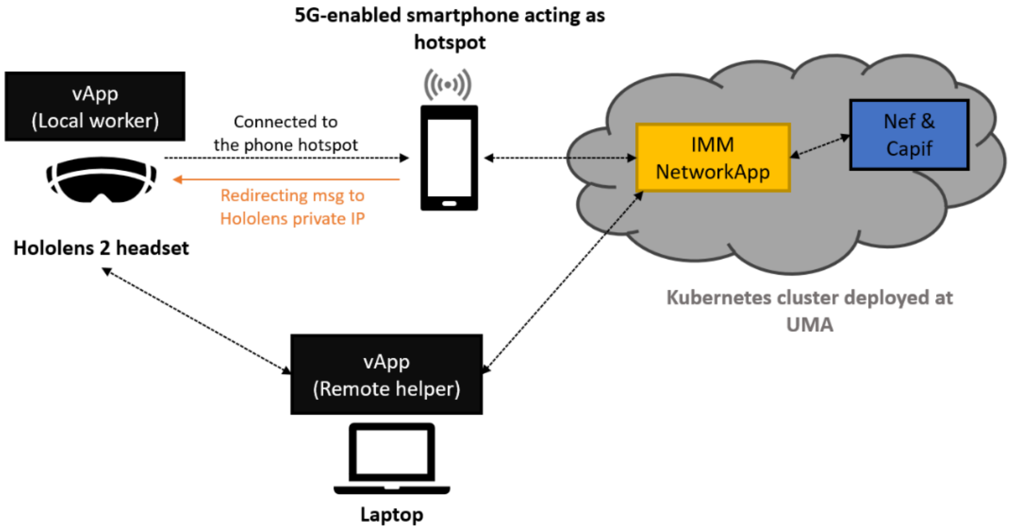


Figure 27: The IMM setup for the second integration round, without TSN

IMM was also interested in Time-Sensitive Networking (TSN) capabilities for their use-case. The second integration round was thus also used to test the IMM use-case and devices with UMA's TSN setup. The initially envisioned setup with complete TSN endpoints is shown in Figure 28.

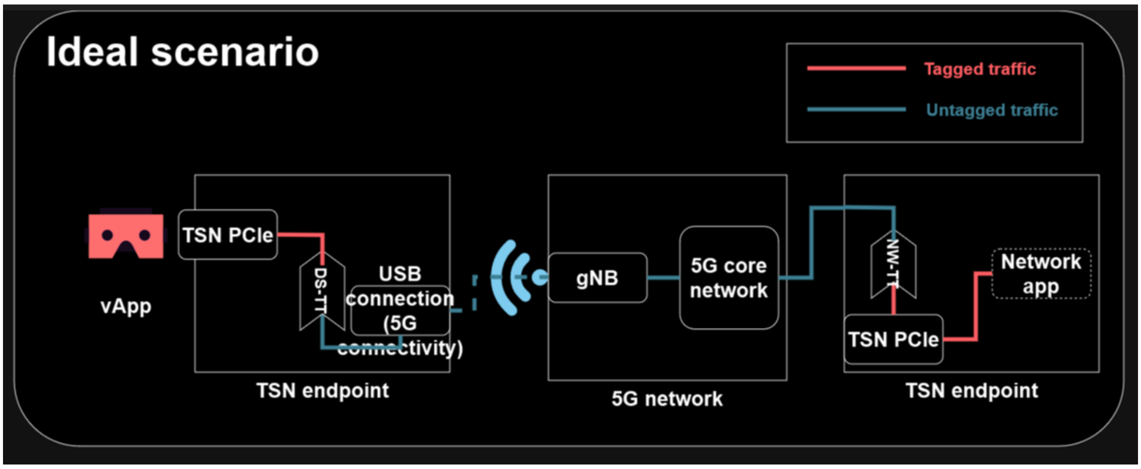


Figure 28: The initially envisioned TSN setup at UMA's premises for the IMM use-case

However, specific adaptations had to be performed in order to test TSN with the IMM use-case. First, the Hololens 2 AR headset does not have an Ethernet port. **More generally, currently available AR headsets do not include any Ethernet port.** The Hololens 2 is one of the only headsets which can nonetheless still be connected to a 5G network thanks to the hotspot of a compatible 5G-enabled smartphone**.** The initial setup at Malaga was not compatible with such a configuration since the used 5G modem (Telit fn980m) in the TSN endpoint could not create directly such Wi-Fi hotspot. The UMA's 5G setup has thus been updated to also work with Wi-Fi.

Besides, another limitation was that some of the network traffic between the two vApp instances was going through an external network. To create a video-conference lobby for users, IMM was using its own service hosted in France by OVH. Both audio and video communications were going through this service called ShariingForYou. The synchronization of virtual objects in AR was directly done between devices (local worker's Hololens and remote expert's laptop). One of the main benefits of this approach is to offload heavy computations from the Hololens, especially encoding and decoding video and audio.

To fully take advantage of TSN and to be able to capture the whole application traffic, IMM adapted the vApp instances. Audio and video streams were directly shared between the two devices, removing the need for their ShariingForYou service. Thanks to this change, a full end-to-end connection in Malaga was achieved for TSN test purposes. The final setup is shown in Figure 29, which includes the Wi-Fi hotspot next to the 5G modem. In this case, the TSN translators do not use tagged traffic in order to be compatible with the IMM application. In addition, traffic beyond the TSN endpoint is marked as best-effort traffic, as it cannot be controlled by the TSN system over 5G.

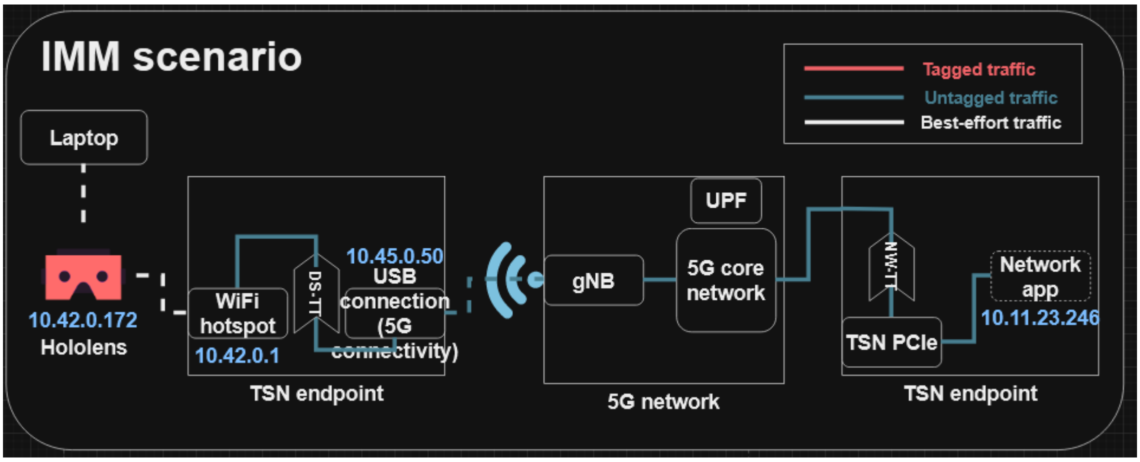


Figure 29: The final TSN-like setup at UMA's premises for the IMM use-case.

The final setup does not fully reflect a complete TSN ideal scenario. However, as the latter was not achievable with the Hololens, it represents a compromise to include some TSN features and endpoints while working with the real vApp instances and devices.

4.5.2 Chatbot assistant Network App

The second round of integration testing was conducted again at the facilities of Demokritos campus in order to check the 5G connectivity between Network Application and vApp and all the other components of EVOLVED-5G in their final versions. For the testing, we utilized Demokritos Kubernetes Cluster Infrastructure along with 5G Amarisoft to evaluate our use case. The topology infrastructure is depicted in Figure 30, where CAPIF, NEF and INFOLYSiS Network Application are deployed on the Kubernetes Cluster Infrastructure and the Chatbot (vApp) on a mobile phone with a 5G SIM-card in order to establish connection to the 5G network via 5G Amarisoft, facilitating the transfer of information within the Kubernetes environment.

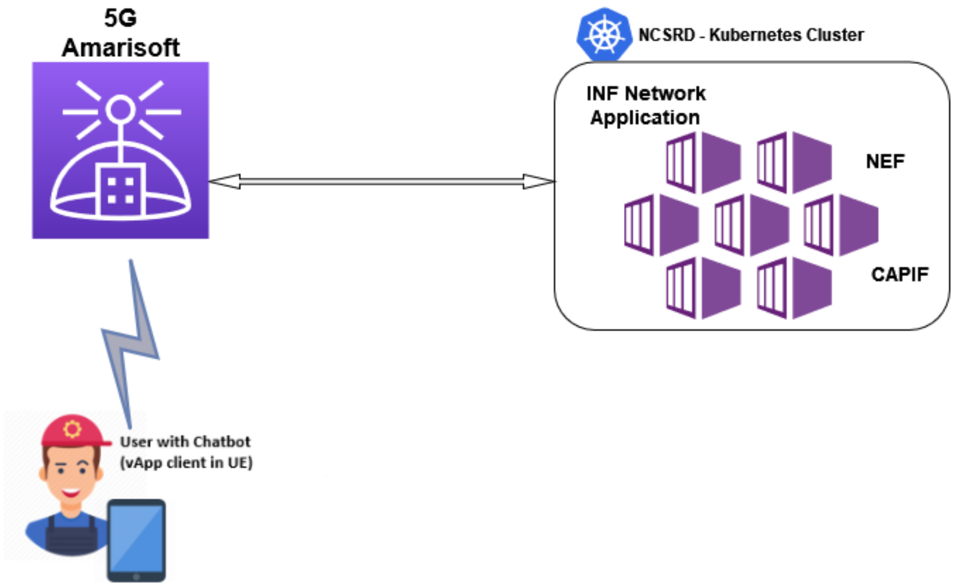


Figure 30: The INF setup for the second integration round.

The second phase of integration activities encompassed the subsequent steps:

1. Upload the image of INFOLYSIS Network Application in Dockerhub:

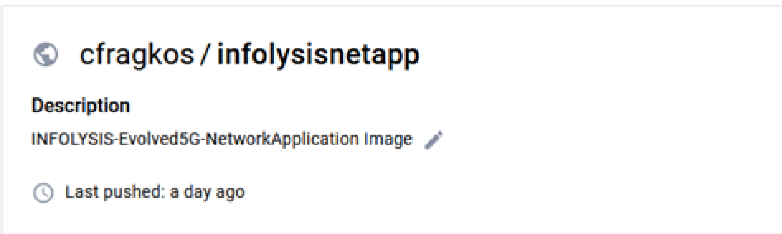


Figure 31: Infolysis Network Application Image.

2. Generating two YAML files that are essential for enabling the INFOLYSIS Network Application to operate effectively on the Kubernetes platform.

   a. deployment.yaml: This YAML file consists of all the essential configurations required to create a pod for the container of the Network Application

K8s > deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: infolysisnetapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: infolysisnetapp
  template:
    metadata:
      labels:
        app: infolysisnetapp
    spec:
      containers:
      - name: infolysisnetapp
        image: cfragkos/infolysisnetapp
        imagePullPolicy: Always
        ports:
        - containerPort: 80
        - containerPort: 8000
```

Figure 32: Infolysis deployment.yaml file.

b. service.yaml: Defines all network ports that need to be exposed in each pod

K8s > service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: infolysisnetapp
spec:
  selector:
    app: infolysisnetapp
  ports:
    - name: apache
      protocol: TCP
      port: 80
      targetPort: 80
    - name: api
      protocol: TCP
      port: 8091
      targetPort: 8000
```

Figure 33: Infolysis service.yaml file.

3. Testing the deployment of all the components with vApp

4.5.3 Digital/Physical twin Network Application

This second round of integration took place at Cosmote's premises in Athens with the aim to utilise the other site of the Athens platform (NSRD, Cosmote). The network setup used was quite similar to the one used for the first round, but the aim here was to demonstrate integration with the latest progress on the project. The various software components were deployed on the kubernetes platform this time, with the latest versions of NEF 2.2.2, CAPIF 3.1.2 and the corresponding network application. Since the setup and the configurations for NCSR Demokritos and Cosmote K8s cluster are almost similar, an initial deployment test had been carried out remotely using the kubernetes platform present at NCSR Demokritos over the previous days, to check that the Network app was working properly.

On the hardware side, the Anita console was this time connected by Wi-Fi to another model of CPE device supplied by Cosmote, which was itself linked to an external 5G access point in standalone configuration.

As in the previous round, the Anita had to be capable both of communicating with the nginx web server within the Kubernetes containers on the internal network and also with the GMI server via external Internet access.

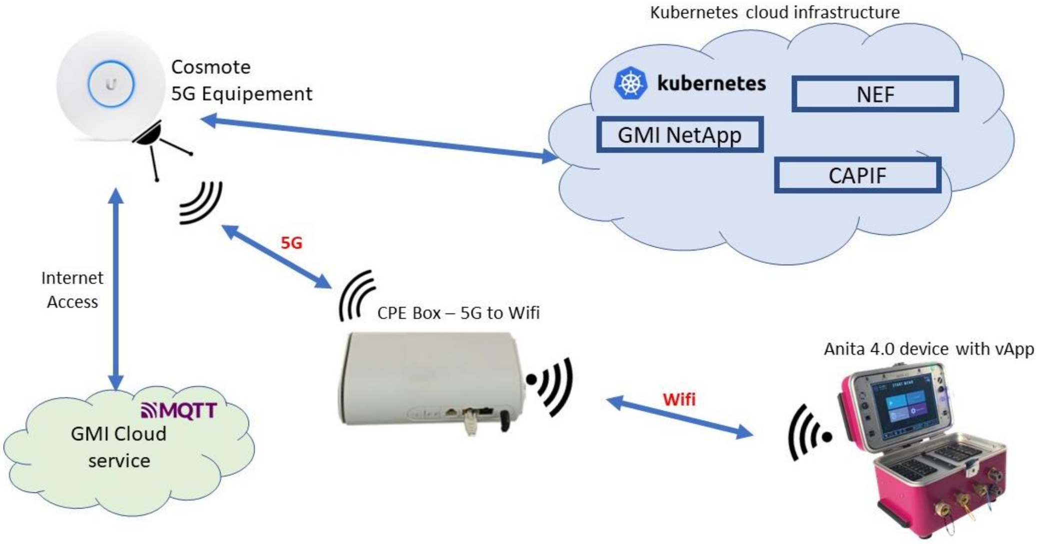


Figure 34 : Network diagram

```
cosmote@evolvemaster:~/k8s-validation-templates/nef$ kubectl get pods
NAME                                        READY   STATUS    RESTARTS      AGE
api-invocation-logs-6b96f499c-jxclv         1/1     Running   0             111s
api-invoker-management-5888597bb-hmwb6      1/1     Running   3 (81s ago)   111s
api-provider-management-55f99cff9f-p6hg4    1/1     Running   2 (95s ago)   111s
backend-55498549c6-7l7td                    1/1     Running   0             24s
capif-events-8476b4875d-rc8m7               1/1     Running   0             111s
capif-mongo-7d89c9f6f8-6jccd                1/1     Running   0             110s
capif-mongo-express-7d547fd679-z9p8g        1/1     Running   3 (46s ago)   109s
capif-routing-info-5d8fd667b-6jnj6          1/1     Running   0             110s
capif-security-758c9bcdb8-5nr5m             1/1     Running   3 (87s ago)   110s
db-65465448bd-g6smt                         1/1     Running   0             24s
easy-rsa-88f8cd5b5-cnn9h                    1/1     Running   0             110s
jwtauth-5cdfb84b5c-7c2k9                    1/1     Running   3 (84s ago)   110s
logs-5d499587f4-j5bgj                       1/1     Running   0             111s
mongo-express-6c9cc9f746-zcbmx              1/1     Running   2 (20s ago)   24s
nef-mongo-9b85fcd44-vxrnk                   1/1     Running   0             24s
nginx-578d7d64f8-86hv7                      1/1     Running   0             109s
published-apis-5ff8f5f8dd-98xpl             1/1     Running   0             110s
redis-7c8976bb95-n8qxp                      1/1     Running   0             111s
reverse-proxy-84b8865bf6-6gnk2              1/1     Running   0             24s
service-apis-d5645c484-2czp7                1/1     Running   0             111s
```

Figure 35 : Nef & Capif services in Kubernetes

```
cosmote@evolvemaster:~/k8s-validation-templates/gminetapp$ kubectl logs -f gmiaeroapp-6959498d45-2rqbn
/usr/local/lib/python3.10/site-packages/requests/__init__.py:102: RequestsDependencyWarning: urllib3 (1.26.16) or chardet
0.12) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({})/charset_normalizer ({}) doesn't match a supported "
Your netApp has been successfully registered and onboarded to the CAPIF server.You can now start using the evolved5G SDK!
/usr/local/lib/python3.10/site-packages/requests/__init__.py:102: RequestsDependencyWarning: urllib3 (1.26.16) or chardet
0.12) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({})/charset_normalizer ({}) doesn't match a supported "
INFO:     Started server process [16]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8383 (Press CTRL+C to quit)
```

Figure 36 : Network Application in Kubernetes

4.6 RESULTS AND TAKEAWAYS

4.6.1 Remote assistance in AR Network Application

The second integration test was successful as well. First, the 5G connectivity of devices (Hololens and laptop) was ensured. The Network Application deployed on Kubernetes was able to communicate with the deployed versions of CAPIF and NEF on UMA's premises. Both the local worker and remote helper could communicate with each other in AR and collaborate in real-time around virtual 3D models (see Figure 37). By simply entering an IP address, the local worker was able to connect to the deployed Network App. This is an encouraging result because it shows that it may be easy and quick for a worker within a factory to connect to a deployed Network App, which are key conditions for adoption.

Figure 37: Overview of the setup and of both IMM vApp instances.

Besides, the fact that our vApp was adapting its behavior based on the current QoS status has been validated. Upon receiving a QoS notification with mandatory and/or suggested adaptations from the Network App, the vApp enabled or disabled some features to adjust to current network conditions. This is highlighted in Figure 38. For instance, when the QoS could not be guaranteed on one user side, the synchronization of Augmented Reality objects was paused. Features like shared virtual cursors or sending annotated screenshots were then disabled. If the QoS was degraded for both users, both cameras were also disabled to save bandwidth and prioritize audio communications.

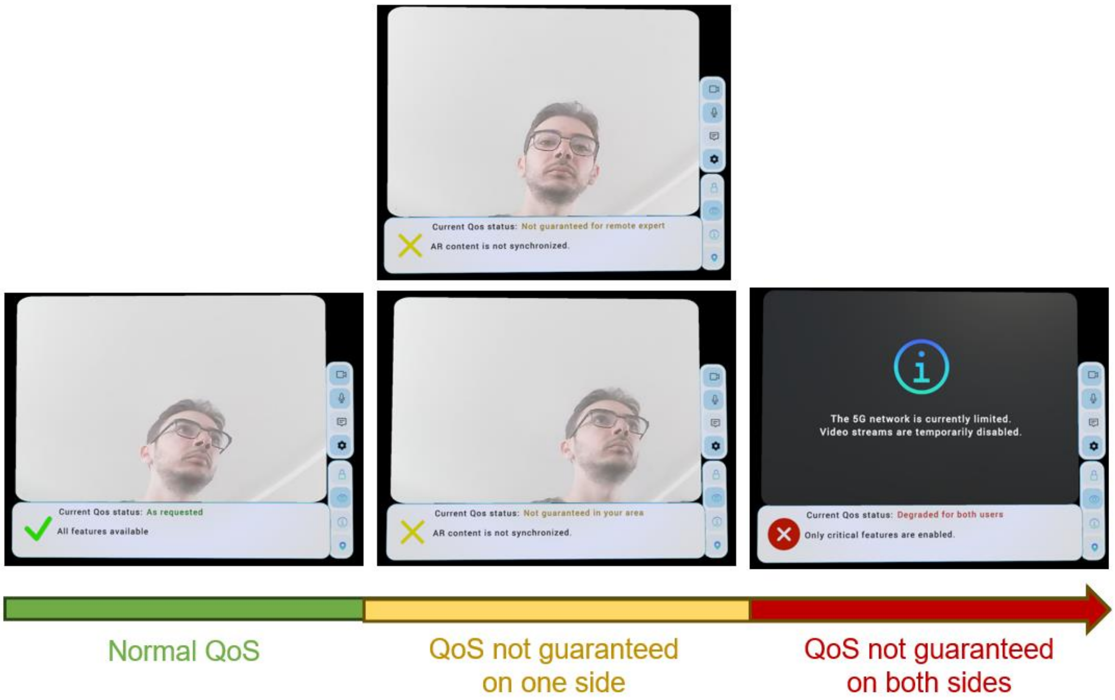


Figure 38: Adaptation mechanisms implemented in the vApp.

Finally, the TSN test was also successful. The updated versions of vApps communicated as expected with the deployed IMM Network App and NEF/CAPIF and the network traffic was captured using the Wireshark software.

As mentioned in Section 4.5.1, the used setup presents limitations compared to a complete TSN setup due to AR headset's constraints. The introduction of a Wi-Fi connection within the TSN endpoint impacted the performance by not being able to guarantee traffic quality. Besides, performing all video and audio encoding and decoding operations directly on the Hololens also significantly impacted performance aspects. Such heavy computations are quickly draining the headset battery, which is problematic as the device offers lower performance while its battery is low (usually around a 40% threshold).

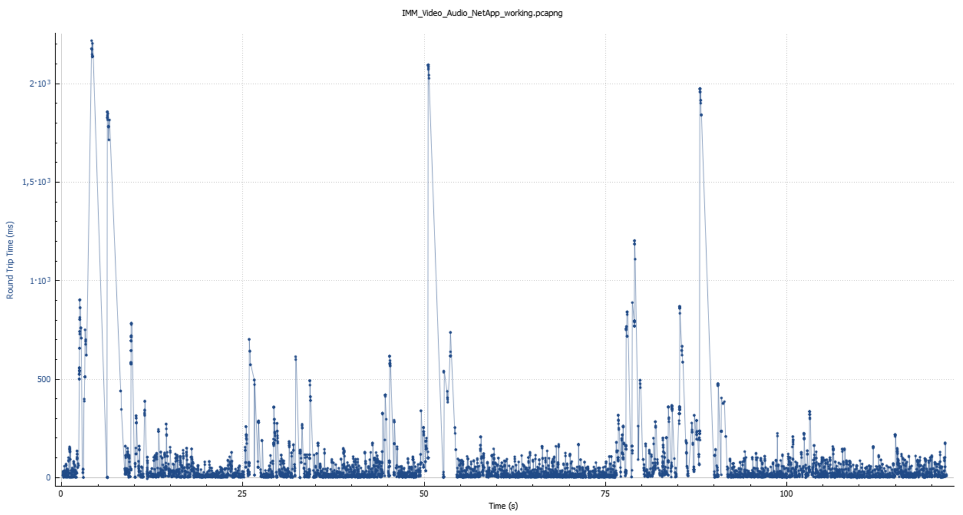


Figure 39: Round-trip time (RTT) observed during the TSN test. Overall, the mean RTT is around 20ms.

Despite these drawbacks, encouraging data was collected during the test, as shown in Figures 39 to 41. For instance, the overall round-trip time for TCP packets was around 20ms on average. This is a major KPI for IMM as perceivable latency is one of the most important factors for the manipulation of shared virtual objects in AR. Such level of latency can already start to impact users performing direct manipulations on objects but remains largely acceptable for interactions over the network and the end users targeted in the IMM scenario.

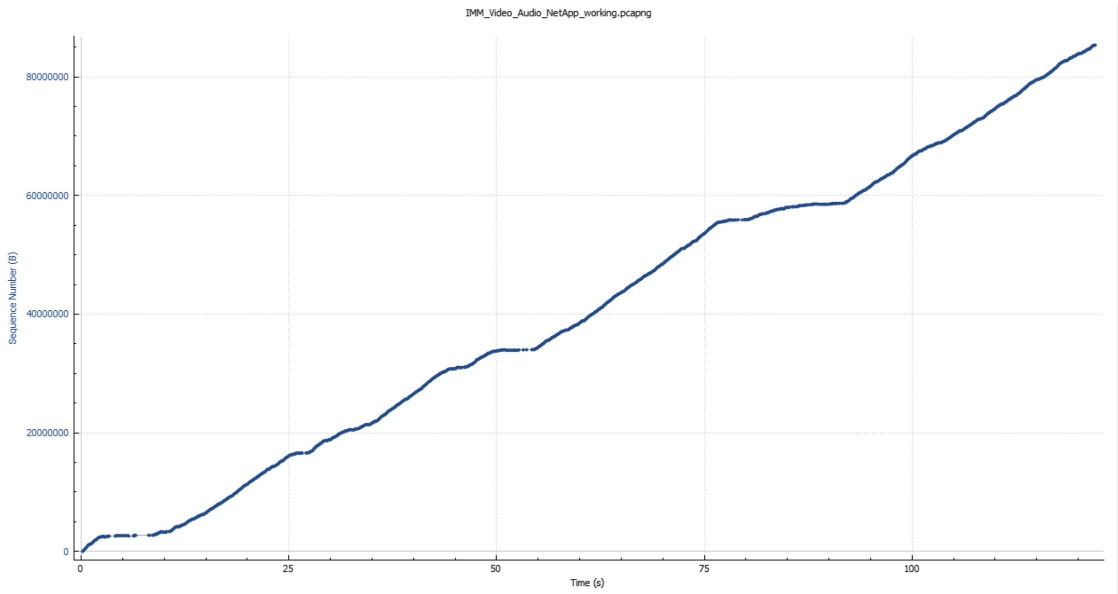


Figure 40: Sequence numbers over time for TCP packets observed during the TSN test. Results are quite close to the ideal scenario (perfect line).

One of the main limitations observed during the TSN test was a non-negligible number of retransmitted TCP packets. This phenomenon may come from the introduction of Wi-Fi in the setup, thus affecting the reliability of packet transmission compared to an Ethernet connection.

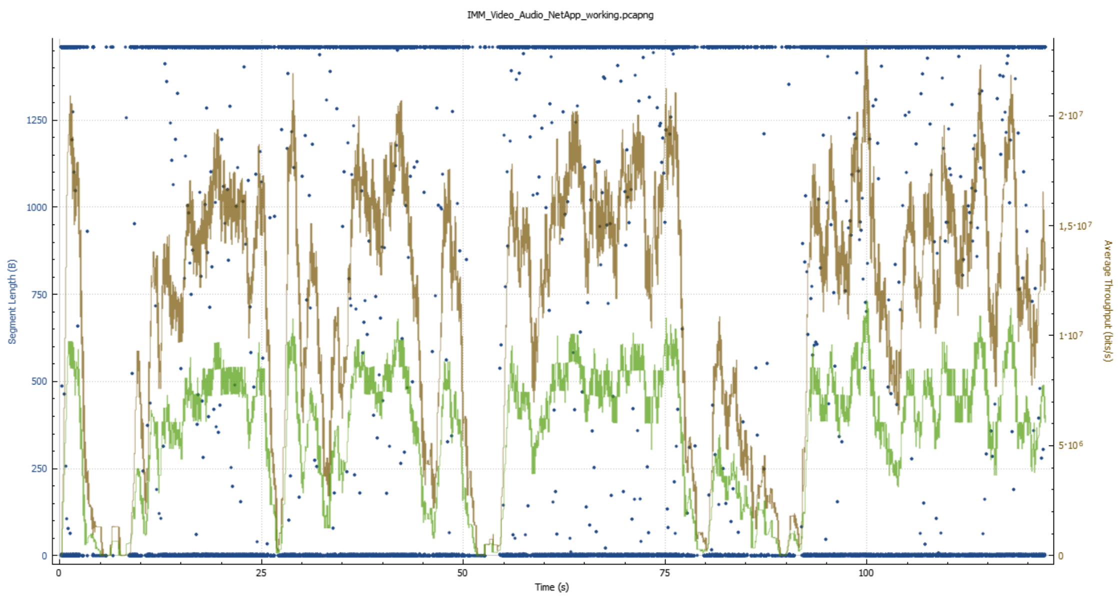


Figure 41: Comparison of throughput (orange) and goodput (green) observed during the TSN test.

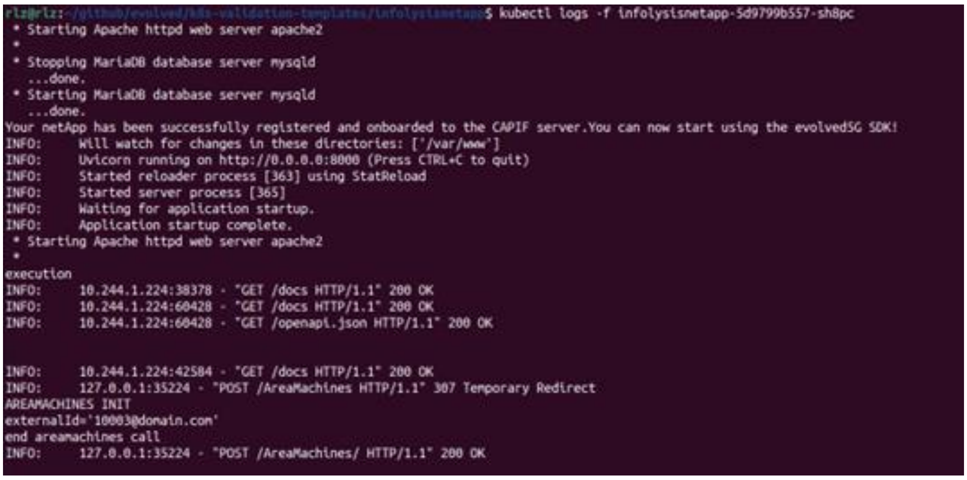
Overall, the IMM use-case was thus validated. All objectives of the integration rounds were successfully achieved.

Besides, the performed tests highlighted several key insights. First, there is a gap between the need of real-time communications for AR scenarios and the capabilities of current AR hardware. While smartphones and tablets may directly have 5G-compatible sim cards, this is not the case for existing AR headsets. Being designed for hands-free and user mobility also means they do not focus on tethered connections. This design choice complexifies the testing of TSN with such devices. Nonetheless, integration tests highlight that it is still possible to adapt TSN setups to work with devices like the Hololens 2 while preserving performance which is acceptable for AR interactions. Finally, we did not observe major effects of the device's battery level on network performances.

A promising next step could be to adapt the current setup to a more ecological environment, *i.e.* inside a factory with a 5G network and with audio and video streams handled by an on-premises architecture.

4.6.2 Chatbot assistant Network Application

After the successful deployment of INFOLYSIS Network Application in NCSRD Kubernetes platform, the functionality of the network application and the end-to-end communication with the Vertical Application (Chatbot) has been tested and validated.



```
rlz@rlz:$ kubectl logs -f infolysisnetapp-5d9799b557-sh8pc
 * Starting Apache httpd web server apache2
 *
 * Stopping MariaDB database server mysqld
   ...done.
 * Starting MariaDB database server mysqld
   ...done.
Your netApp has been successfully registered and onboarded to the CAPIF server.You can now start using the evolved5G SDK!
INFO:     Will watch for changes in these directories: ['/var/www']
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [363] using StatReload
INFO:     Started server process [365]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
 * Starting Apache httpd web server apache2
 *
execution
INFO:     10.244.1.224:38378 - "GET /docs HTTP/1.1" 200 OK
INFO:     10.244.1.224:60428 - "GET /docs HTTP/1.1" 200 OK
INFO:     10.244.1.224:60428 - "GET /openapi.json HTTP/1.1" 200 OK


INFO:     10.244.1.224:42584 - "GET /docs HTTP/1.1" 200 OK
INFO:     127.0.0.1:35224 - "POST /AreaMachines HTTP/1.1" 307 Temporary Redirect
AREAMACHINES INIT
externalId='10003@domain.com'
end areamachines call
INFO:     127.0.0.1:35224 - "POST /AreaMachines/ HTTP/1.1" 200 OK
```

Figure 42: Successful Deployment of Infolysis Network Application in NCSRD Kubernetes Platform.

First, the administrative platform of INFOLYSIS network application at the following url: "https://infonetapp.com/configui/index.php" was accessed to check that everything works fine.

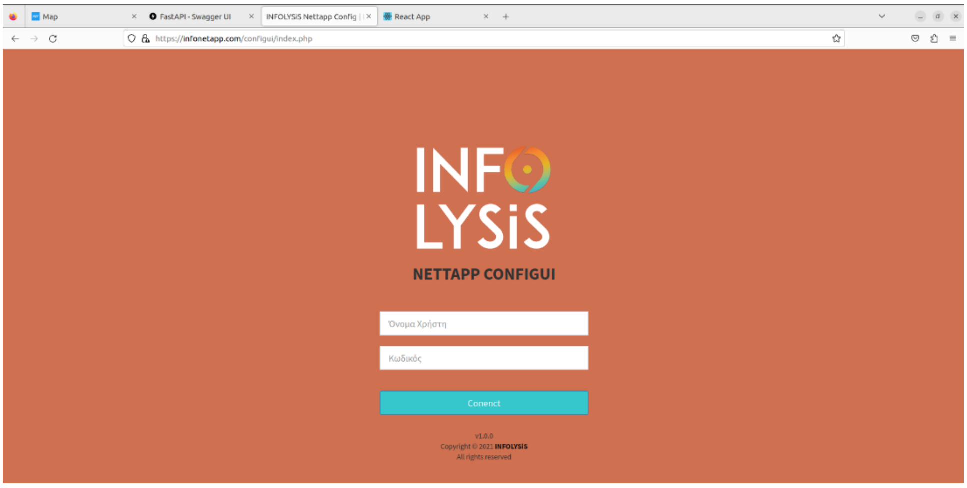


Figure 43: Administrative platform in NCSRD Kubernetes Platform.

Then, testing of the Vertical Application (Chatbot) was initiated with a phone connected to the 5G network via 5G Amarisoft. Everything was verified for proper functionality, including the validation of real 5G connectivity for the use case, and ensuring effective communication between the Vertical Application (Chatbot), Network Application, NEF, and CAPIF. The complete validation of the use case, involved the final components of Evolved-5G, evaluated using the default scenario provided by the NEF Emulator. Within this designated scenario, four cell areas were defined, and one of these areas had a malfunctioning machine issue. Furthermore, there were three UEs, each with distinct levels of permissions. In summary, the validation process confirmed that UEs lacking the appropriate permissions were unable to detect any anomalies within the specified cell area. In contrast, UEs possessing the requisite permissions successfully identified problems within the area and request manual assistance to address the malfunctioning machine.

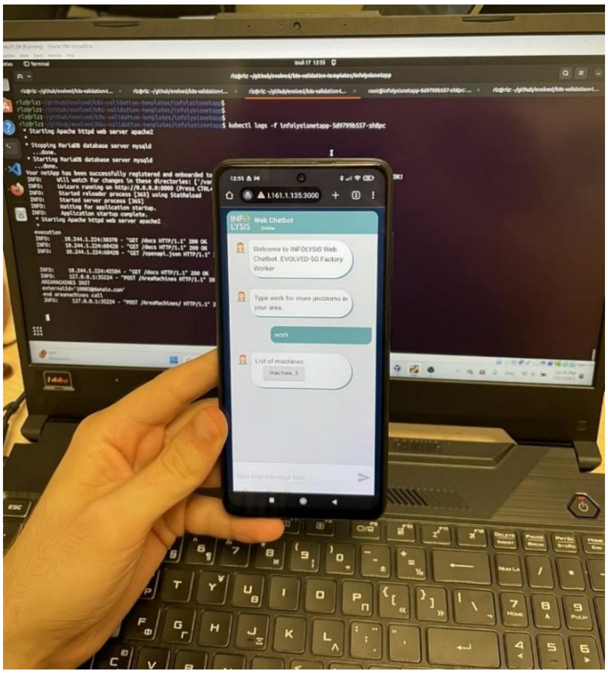


Figure 44: Chatbot in NCSRD Kubernetes Platform.

4.6.3 Digital/Physical twin Network Application

The second round of integration test was also successful for GMI and the three objectives achieved. During this second test session, the Anita console was able to connect properly to the new CPE in use, as well as to the platform hosting Kubernetes on Cosmote server, using the 5G network and last version of the different services. This last part was a challenge for the NCSRD and Cosmote teams, as it had never been tested and required a significant amount of configuration effort.

On the other side, the vApp was able to communicate securely with the Network application and retrieve the required information. Moreover, the connection with the Cloud server was effective and the curing data were successfully transmitted.

In accordance with the defined use case, the content of the data packet transmitted has been correctly adapted according to the QoS. A specific network tool was used to visualise the data transmitted to GMI's cloud server. This allowed to check that a laptop on a completely independent network (4G connection shared from a mobile phone) could exchange data with GMI's cloud server.

In that sense, all use case related objectives, namely establish secure communications between vApp and network application, proof of the ability to request 5G services and retrieve quality of service information ensuring fidelity of the digital twin development, as well as the sound presentation of the geographical coordinates of the cell in which the ANITA 4.0 is located in order to document the composite repair report have been fully achieved and validated.

Figure 45 : Anita curing cycle and external laptop

```
cosmote@evolvemaster: ~/k8s-validation-templates/capif    cosmote@evolvemaster: ~/k8s-validation-templates/nef    cosmote@ev
Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage
 warnings.warn(
GMI NETAPP

=> VERSION OF THE NETAPP : 4.2

NETAPP_PATH : /src
--- RETRIEVING INFORMATION ABOUT SUBSCRIPTION 64ca1b7033b7e4345a4fd048----
{'alt_qo_s_references': [2, 82],
 'dnn': 'province1.mnc01.mcc202.gprs',
 'ipv4_addr': '10.0.0.1',
 'ipv6_addr': '::1',
 'link': 'http://nef-backend/nef/api/v1/3gpp-as-session-with-qos/v1/None/subscriptions/64ca1b7033b7e4345a4fd048',
 'mac_addr': '22-00-00-00-00-01',
 'notification_destination': 'http://gmiaeroapp:8383/monitoring/callback',
 'qos_mon_info': {'lat_thresh_dl': None,
                  'lat_thresh_rp': None,
                  'lat_thresh_ul': 20,
                  'rep_freqs': ['PERIODIC'],
                  'rep_period': 15,
                  'req_qos_mon_params': ['UPLINK'],
                  'wait_time': None},
 'qos_reference': 1,
 'snssai': {'sd': '000001', 'sst': 1},
 'usage_threshold': {'downlink_volume': 5368709120,
                     'duration': None,
                     'total_volume': 10737418240,
                     'uplink_volume': 5368709120}}
INFO:     10.244.1.128:50600 - "GET /qos_createQuarantedSubscription HTTP/1.1" 200 OK
```

Figure 46: Network Application log and subscription on K8s

```
QoS : Deleting subscription with id: 64ca1b7033b7e4345a4fd048
INFO:     10.244.1.128:50600 - "GET /qos_deleteSub HTTP/1.1" 200 OK
New notification retrieved : callbacktype = qos
QOSOK
INFO:     10.244.1.44:45520 - "POST /monitoring/callback HTTP/1.1" 200 OK
New notification retrieved : callbacktype = qos
QOSOK
INFO:     10.244.1.44:35878 - "POST /monitoring/callback HTTP/1.1" 200 OK
New notification retrieved : callbacktype = qos
QOSOK
INFO:     10.244.1.44:56874 - "POST /monitoring/callback HTTP/1.1" 200 OK
New notification retrieved : callbacktype = qos
QOSOK
INFO:     10.244.1.44:43118 - "POST /monitoring/callback HTTP/1.1" 200 OK
New notification retrieved : callbacktype = qos
QOSOK
INFO:     10.244.1.44:38354 - "POST /monitoring/callback HTTP/1.1" 200 OK
New notification retrieved : callbacktype = qos
QOSNOK
INFO:     10.244.1.44:57264 - "POST /monitoring/callback HTTP/1.1" 200 OK
```

Figure 47 : Notification callback for QoS

```
Coordonates for UE  10001@domain.com  requested : 45.578625,5.974191
INFO:     10.244.1.128:50600 - "GET /location_getLatAndLong/10001%40domain.com HTTP/1.1" 200 OK
INFO:     10.244.1.128:50600 - "GET /qos_getStatus HTTP/1.1" 200 OK
INFO:     10.244.1.128:50600 - "GET /qos_getStatus HTTP/1.1" 200 OK
```

Figure 48: Network application log for localisation

5 CONCLUSION AND NEXT STEPS

The work presented in this deliverable describes in detail the final prototypes of the Network Apps developed within the Interaction of Employees and Machines pilar in the EVOLVED-5G context, driven by Task 4.2. Moreover, detailed descriptions of the two iterations of integration tests that the Network Apps have undergone on top of the EVOLVED-5G infrastructure, both in Athens and UMA platforms, are provided. The integration activities aimed to test the functionality of the Network apps when seamlessly connected with the vApp as well as evaluating the overall use case for each Network App.

SME	5G connectivity (vApp – Network Application)	Communication between last versions of EVOLVED-5G deployed components (NEF, CAPIF, SDK, TSN, Network Applications)	Real-world testing of use-cases	TSN
IMM	✓	✓	✓	Tested with adapted setup
INF	✓	✓	✓	Not needed
GMI	✓	✓	✓	Not needed

Table 2: Summary of the objectives achieved during the integration tests.

With the second round of integration tests, the Networks Apps of the IEM pillar have reached their final stage, interacting with the last versions of NEF and CAPIF through the SDK and communicating with their respective vApp(s). The three SME use-cases have been validated on a real 5G network provided by the two respective infrastructures. Besides, a TSN setup was adapted and tested for the IMM use-case. Such results highlight the fact that the Network Apps reached a mature state. This progress marks the end of WP4. Now that Network Applications are complete and deployed versions tested on real 5G setups, the next steps will take place within the scope of WP5. SMEs have used the validation pipeline to check their Network Applications. When they pass the final validation and certification steps, the Network Applications will be ready to be made publicly available through the EVOLVED-5G Marketplace. This way, other professionals and researchers will be able to use these applications, learn from them and build their own following the EVOLVED-5G methodology and pipelines.