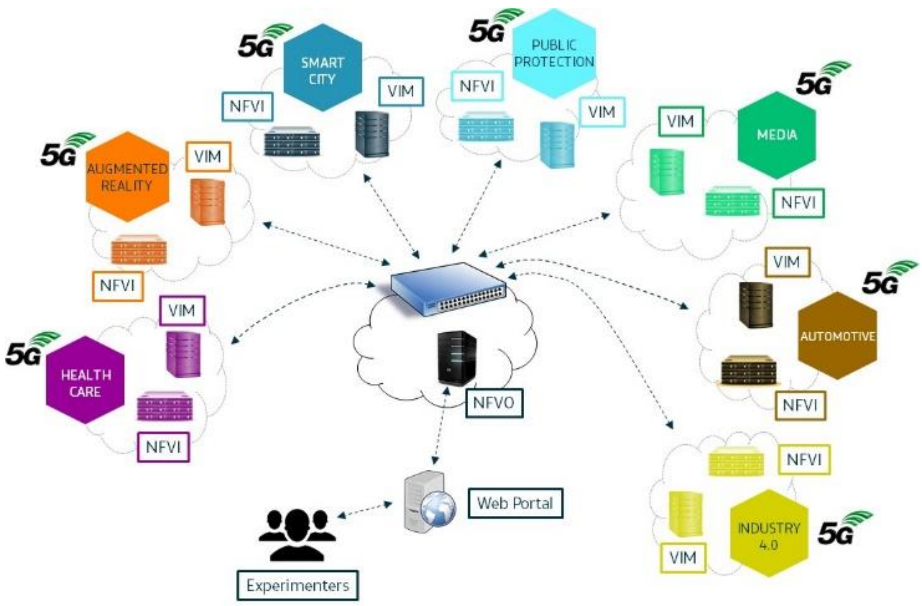
EXPERIMENTATION OVER DISTRIBUTED 5G NFV-BASED ENVIRONMENTS



Abstract

5GinFIRE is a Research and Innovation Action project under the EU programme Horizon 2020 [1]. It offers an open and extensible 5G NFV-based ecosystem of experimental facilities for third parties, selected through open calls. 25+ successful experiments, focused on a wide set of heterogeneous topics, have been executed so far over the 5GinFIRE infrastructure.

The purpose of this whitepaper is to share some of the mechanisms that have permitted to conduct such an important amount of experiments over 5GinFIRE. On the one side, **testing workflows** were defined so that participating actors, roles, capabilities and offered services were clear for the incoming experimenters. On the other, schemes to **develop, on-board and deploy experiments** were designed to improve the availability (and repeatability) of 5G tests.

https://doi.org/10.5281/zenodo.3568721 December 2019

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732497. This publication reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Impressum

Full project title: Evolving FIRE into a 5G-Oriented Experimental Playground for Vertical Industries

Short project name: 5GINFIRE

List of authors

Juan Rodriguez Martinez, Telefonica

Diego R. Lopez, Telefonica

Christos Tranoris, University of Patras

Iván Vidal Fernández, University Carlos III de Madrid

Anastasius Gavras, Eurescom GmbH

The content of this white paper has been edited using material from several 5GinFIRE project deliverables. The editors wish to acknowledge all contributions of the 5GinFIRE consortium experts that are listed in the project deliverables. More information about 5GinFIRE deliverables is available at https://5ginfire.eu/deliverables/.

1 Motivation

Affirming that the Cloud's usage has grown enormously over the past couple of decades is by no means an overstatement. What quickly became a very popular solution for remote storage has just as quickly evolved in many different directions, for example the hosting of applications which are massively consumed by users independently of their location. Well known cloud service providers (like Amazon or Google) have since developed very successful services permitting a seamless integration of cloud computing in our daily routine.

In the past few years, most telco providers have also tried to incorporate, at least to some degree, cloud concepts and technologies. Beyond telecom operators offering cloud services, numerous initiatives have been started to: i) leverage on commoditized homogeneous hardware, ii) decouple the applications from the hardware where they will be executed, and iii) orchestrate resources so that they are more efficiently consumed. Network Functions Virtualization (NFV) precisely builds on top of these characteristics, deploying the required Virtual Network Functions (VNFs) on top of industry-standard homogeneous hardware.

One of the big differences between cloud and NFV technologies is that cloud applications are typically deployed in centralized datacentres, while VNFs on many occasions must be executed at very specific network locations. Even more importantly, centralized cloud datacentres are not ideal for a big set of applications which demand very strict requirements, mainly latency-related. These two facts open new opportunities for operators, which may take advantage of distributed datacentres to host new cloud services with very stringent latency requirements.

In that sense, the appearance of 5G has only fostered concepts like Distributed Cloud or Edge Computing, both of which imply distributing computing and storage resources closer to the end users. In fact, the number of potential use cases is so incredibly high and varied that this heterogeneity has become one of the biggest challenges for external experimentation [2].

In 5GinFIRE, we have dealt with heterogeneity since day 1 of experimentation; the results of the different open calls were even unknown to the 5GinFIRE consortium, since it was a group of external experts who decided the awarded proposals. The project tasks were, therefore:

- To align the experiment objectives with the offered services,
- To accommodate the VxFs[1] subject to be tested in the available infrastructure,
- To provide useful experimentation tools, and
- To guide and help experimenters who, in any case, were encouraged to execute their tests as autonomously as possible.

The *Test, Measurement, and KPIs Validation* Working Group of 5G PPP has recently produced a whitepaper focused on "Assessing 5G architecture and Application Scenarios" [3]. While 5GinFIRE has not dealt with all the topics that the whitepaper addresses, we believe that sharing our experienced vision in the definition of test workflows and in the

[1] We define a VxF as a potentially complex constellation of virtual functions in which we do not want to distinguish between network-centric functions (or VNFs) and application-centric functions.

description of experiments will provide valuable input, and ideas for consideration, to anyone willing to expand on the assessment of 5G services.

2 The 5GinFIRE Test Platform

The long-term objective of the 5GinFIRE project was a very specific one: to design and jointly operate flexible 5G NFV-enabled experimental facilities for the exploration of software-based architectures and services. As a result, the main assets offered by 5GinFIRE (via its multiple open calls) were two: compute resources for NFV, plus testing scenarios meaningful for 5G, like smart Cities, automotive applications, eHealth, etc.

The initial layout of the 5GinFIRE testing facility included four sites:

- The Automotive Testbed, at the *Instituto de Telecomunicações* of Aveiro (Portugal)
- The Smart City Testbed, at the *University of Bristol* (UK)
- The Future Internet Testbed, at the *Universidade Federal de Uberlândia* (Brazil)
- And the 5G Telefonica Open Network Innovation Centre (5TONIC), a multipurpose testing environment located in Leganés, Madrid (Spain)

Through two open calls, five more facilities were included, focused on heterogeneous topics like 5G Media, eHealth, Virtual/Augmented Reality, Reconfigurable Radio, Slicing, Public Protection, Disaster Relief or Wireless Home Entertainment.

During the lifetime of the project, the designed orchestration framework was maintained, monitored and continuously updated. The final deployment is depicted below in Figure 1. The 5TONIC lab is hosting the single NFV Orchestrator (NFVO), implemented by successive versions of Open Source MANO (OSM). Around 5TONIC, which is also the hub of a “hub & spoke” VPN-based topology for connectivity (both in the data and in the control planes), we deployed the aforementioned multi-VIM (Virtualized Infrastructure Manager) setup of testbeds, on occasions with even more than VIM per site, separating different environments.

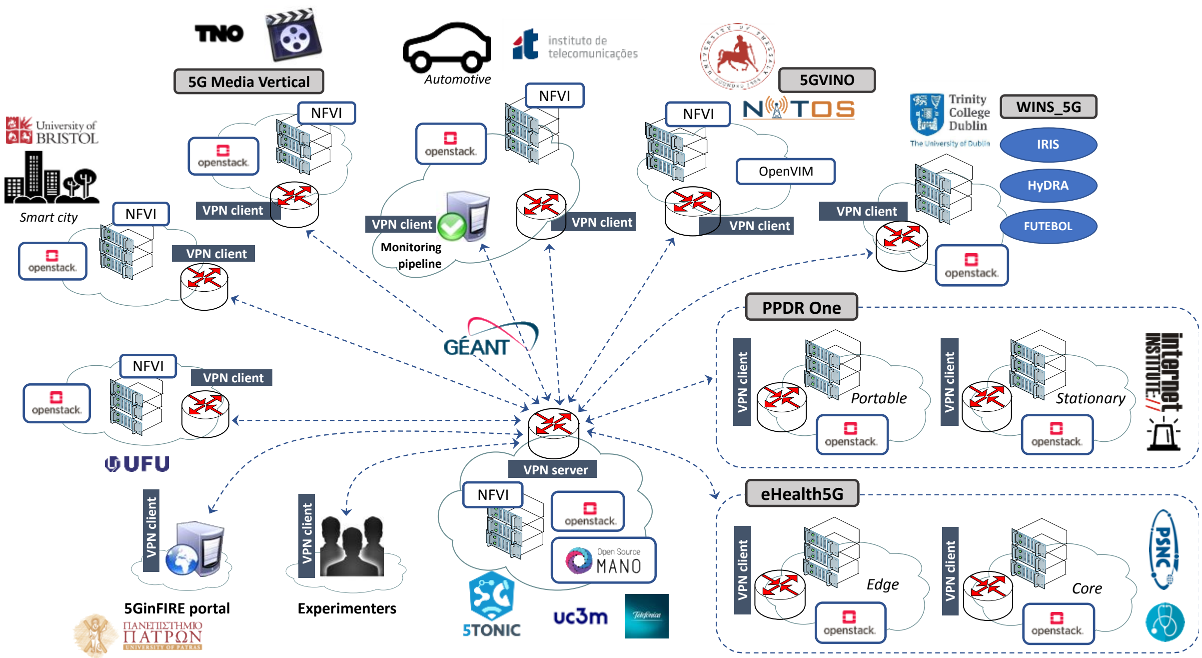


Figure 1: 5GinFIRE Orchestration Framework

Regarding the "hub & spoke" VPN topology, it is important to emphasize that it supports the following types of control and data-plane communications:

- Communications between OSM and the VIMs. This type of communication allows the orchestrator to coordinate the allocation and configuration of computing, storage and network resources at the diverse 5GinFIRE sites.
- Communications between OSM and the VNFs deployed at each site, to support the day-1 configuration of VNFs after their deployment.
- Inter-site communications, to enable the exchange of data among VNFs located at different sites.

These types of communications have been enabled at every site using different mechanisms and technologies (within 5TONIC they are provided through different VLANs, for example). Additional information on the 5GinFIRE orchestration platform can be found in [4].

Finally, the front end for experimenters, and only entry point towards the platform, is the 5GinFIRE portal. The portal implements multiple functions, among others:

- It interacts with concurrent users (multi-tenancy), based on different user roles assigned with different operational permissions
- It stores both VxF images and VxF/NS descriptors, implementing a friendly catalogue interface
- It provides user access to the various available services, in particular the issue management system (based on tickets) and the descriptor validation service, which checks the format of descriptors, to name just a few
- It triggers the experiments' deployment, based on user request and after acceptance by the site owners, using OSM's NorthBound Interface (NBI)
- It controls the lifecycle of the different experiments being executed

The 5GinFIRE portal has been developed[2] by the *University of Patras* (Greece), and is hosted in its premises. It interacts with the 5GinFIRE orchestrator via the same control plane VPN as the rest of the sites. Beyond the user-destined features listed above, it also permits the operations team of 5GinFIRE to have access to the infrastructure Health-Check Service (HCS).

Some numbers related to the Portal include:

- ✓ 100+ users
- ✓ 150+ onboarded VxFs (50+ public)
- ✓ 90+ onboarded NSDs (30+ public)
- ✓ 500+ performed deployment requests
- ✓ 3200+ tickets, including not only bugs or reported issues by experimenters, but also platform notifications, infrastructure related events, deployment requests, etc.

[2] http://wiki.5ginfire.eu/portalarchitecturedesign

3 Testing Services and User Roles

To execute more than 25 heterogeneous experiments with numbers as those from above, a very strict (yet flexible) methodology for testing had to be designed. It is a very good practice to define this architecture early in the design process, since it helps to correctly identify which are the participating **actors**, which are the required **testing services** and which are the **capabilities** or **tools** that will support those testing services. These are key elements in any testing framework designed to succeed, and must therefore be consistently revisited to ensure the house is not being built from the roof down. Once the foundations are finished, then the roof, or **value-added services**, can be handled and included in the design[3].

One very important feature of any testing facility is its level of automation. A very controversial idea that must be assumed is that “full automation is not possible”, and in fact should not be the final goal for any type of experimentation facility involving realistic service use cases. Either test cases are too heterogeneous, or they require very specialized equipment, or they involve different sets of KPIs, or they prioritize different testing services. Today, there exist too many technical and operational challenges as to realistically believe everything will be plug and play, and triggered by a single “RUN” button. Even if it was possible, any search for improvement (e.g., to optimize the usage of resources), or any new service made available, would most probably limit, at least temporarily, the full automation capabilities of our facility.

Therefore, we must assume that there will be processes which will not be automated, and will consume a good amount of time and effort. During the descriptions included in the following sections, we will thoroughly provide the reasoning behind the 5GinFIRE automation proposal, with the hope it may be of use for future Testing-as-a-Service (TaaS) providers.

3.1 Main Offered Services

Before defining the list of potential actors, this is the definition of “Experiment” in the scope of 5GinFIRE:

- **Experiment**: a set of testing activities that will be conducted during an allocated time-slot (probably spanning for several days) in which VxFs can be automatically deployed on top of the available testing infrastructure. The experiment may or may not involve multiple sites, and may or may not require the utilization of multiple network services

This definition is important because it basically summarizes what the main offered services for testing are[4].

[3] Of course, which are “basic” and which are “value-added” services will depend on many aspects, from the type of expected users and use cases, to the testing expertise, the available resources, the goals of the testing facility (technology benchmarking versus service testing), etc. What is critical is to define your “basic” services for your desired objective, and to focus on ensuring those are correctly provided.

[4] The 5GinFIRE Platform offers other types of services as well, some of which have even been outlined in section 2. Those are the considered value-added services, not critical for the project’s long-term objective.

The first one is obvious: **a multi-site testing infrastructure on top of which virtual functions can be deployed for an agreed amount of time**. The main difference with other potential infrastructures is that each of the 5GinFIRE sites offers different capabilities related to different 5G scenarios, so as a result there are different very specialized environments complementing each other (which obviously saves costs).

Secondly, it must be noted that the definition above only includes the concept of “testing”, which is in this case decoupled from the concept of “analysis”. In 5GinFIRE we have not developed this type of results validation (or even monitoring) service. As a project, we were not focused in technology benchmarking, but on application testing. Each experiment has its own test procedures and KPIs of interest, and 5GinFIRE was already a very heterogeneous environment. Furthermore, we were not even aware of the experiments that we would need to host, since these came from open calls externally evaluated.

Considering all these facts, the trade-off that we took was to provide **external management access to the deployed VxFs** as a basic test service. This way, experimenters can trigger their test procedures and collect their own results remotely, but replicating the mechanisms they would employ locally.

Another important concept derived from the definition is that experimenters may make use of any public VxF in the 5GinFIRE repository, not only their own. The idea that “no lab fits all” has been very much generally accepted, so “reusable VxFs” is our response for this challenge. Any incoming experimenter may in fact base its experiments in a combination of its own components with some others already in the repository, improving the efficiency and reducing costs. This, however, demanded **a friendly catalogue service with advanced information about the hosted VxFs**, yet another basic service to be provided by 5GinFIRE.

The final key concept is about the **automatic deployment of experiments**. We will not extend here about the obvious advantages of this service; instead, suffice to say that it is built on top of the orchestration architecture outlined in section 2 (which is the “tool”, or “capability”, supporting this service), and that the process will be fully described in section 5.

3.2 Test Methodology Actors

All users in 5GinFIRE can be accommodated in the following list of specific roles, all of which can be authenticated at the portal:

- **Experimenter**: it is the user that consumes the 5GinFIRE testing services to deploy and execute an experiment
- **VxF developer**: it is the responsible of uploading VxF images and VxF/NS descriptors into the repository
- **Testbed provider**: this role includes internal users in control of the testbed administration, configuration, integration, support, etc.
- **Experiment mentor**: this is a very special role, normally a testbed provider in charge of monitoring the progress of an experiment, providing guidance and support, and also allowing or denying deployments
- **Service administrator**: it includes internal users operating and maintaining the 5GinFIRE services and tools

In sections to come, describing some of the procedures associated with testing in 5GinFIRE, not all roles will participate in all processes. In particular, it is interesting to stress the difference between experimenter and VxF developer.

In 5GinFIRE, anybody can be just an experimenter willing to conduct an experiment, for example with the available public VxFs. Of course, specific permissions are required for this, but the platform is open to anyone interested. The platform is also open to anybody who wants to become just a VxF developer, willing to deliver a new virtual function to the catalogue; a different set of permissions controls this type of access.

Both are possible at the same time as well, like for the awarded participants in the open calls: they provide their own virtual functions, and execute their desired tests over them. But the two roles are not tightly coupled, and that is why the focus in the *Experiment Development and Onboarding* section (4) is placed in the VxF developer, and the key player in the *Testing Workflow* section (6) is the experimenter.

3.3 Feasibility Analysis

In section 1, it was commented that one of the tasks of the members of 5GinFIRE with regards to the new partners resulting from open calls was "to align the experiment objectives with the offered services".

Another big challenge of experimentation in many vertical application environments is the interaction itself between the testing service providers and the verticals themselves. We, testing service providers, tend to use complex terms like "reference architecture", "network topology", "connectors" … while vertical experimenters tend to focus on how to install their applications, and how to immediately access them to start testing. Even the KPIs of interest diverge many times: we concentrate on KPIs we are used to measure, like bandwidth or delay, while a healthcare service provider may be interested in comparing what is the average duration of remotely executed diagnosis when using video calls over 4G or 5G.

It is not that we are not sharing a common language; simply our objectives, vision, previous expertise and expectations are sometimes billions of miles away.

When approaching vertical sectors for 5G experimentation, one should expect the most varied proposals[5]. In the above example, a network engineer would for sure measure the video rate of the application, determine that 4G is already capable of handling that capacity, and declare that there should be no significant deviation in the average duration of the calls, without making any test over a 5G network. The healthcare provider could instantly argue that 5G permits "unlimited" capacity (expectations), so the video quality could be increased, which would permit closer looks at patients, thus reducing the duration of the call.

The network engineer could then be tempted to reduce the use case to a bandwidth problem, and indeed the capacity is a key component of the problem. However, the vertical service provider is not always capable of determining the relationship between a specific network parameter and a specific gain in which they are interested (reduce the duration of the calls).

[5] The expectations created by 5G play a significant role here, since it is still a very much unknown technology for most of the vertical ecosystems.

In 5GinFIRE, we had a big advantage to align offered capabilities with experimentation proposals. Since the experiments resulting from the open calls were going to be funded, we made available detailed information about services, tools, capabilities, processes, infrastructure, etc., and expected proponents to present experiments already aligned with our architecture and procedures. Independently of the quality of the available material, which can be argued but which was improved at successive open calls, we were eventually forced to execute a feasibility analysis of all the proposals. We can say that this was a value-added service, in which experimenters could very briefly describe their objectives (one or two pages), and we provided guidance to align them with our offer (before they presented their final proposals).

The most obvious lesson learnt is that simple mechanisms have to be made available for vertical service providers to clearly state their objectives, in their own terms, if we (service providers) want them to approach us and benefit together. Filling them with a huge amount of information will just not work. Providing them with a huge amount of 5G measurements that are somehow related with their applications will just not work. Expecting them to employ a huge amount of effort to adapt to our testing conditions, today, when business models are still not clear, will just not work.

Time has to be spent in understanding each other’s positions, ensuring an appropriate experiment/facility interaction. The final goal should be that vertical applications are tested in our facilities in a way as similar as possible as they are in the own vertical facilities. This is how we will improve efficiency. The good news: there is a lot of room for innovation with regards to automation in this kind of processes. In fact, as big as the room can possibly be.

4 Experiment Development and Onboarding

The 100+ users of the 5GinFIRE Platform, in their role as experimenters, have access to more than 50 public VxFs and 30 public NSDs. What is more, the number of total VxFs and NSDs triples the public numbers. That success is thanks to the efforts of VxF developers, those allowed to onboard VxF and NSDs (experiment descriptions in general) through the portal.

One of the biggest lessons learnt in this experimentation arena is that such task is by no means simple for many of the incoming players, vertical application providers in particular.

To begin with, prior to the development of VxFs/NSDs, the developer must understand the target facility, its specificities and must plan for any specific requirements (e.g., SR-IOV or VNF Internet access). This is of particular importance when dealing with multi-site experiments. It is very common, for example, that experimenters are used to having everything connected in the same local network (LAN), and encounter difficulties when facing a L3 connectivity environment. In the end, their expertise is on building applications, not on networking. The importance of a good feasibility analysis, and a good posterior interaction with the developer, is huge at this stage.

Also, creating VxF packages or NS descriptors is a very strict task, dependent on numerous factors, and very prone to errors. In 5GinFIRE, we published very detailed tutorials for both items on the public Wiki [5], to try and help developers in their task. Still, they required a lot of troubleshooting on the occasions when their requirements differed from those shown on the Wiki.

For the creation of VxFs, the tutorial included the following topics:

1. How to build the VM image
2. How to implement a VNF descriptor
3. How to create a JuJu Charm for initial configuration
4. How to package everything

Similarly, for NS descriptors, the tutorial showed:

1. How to create the folder structure
2. How to complete the NSD YAML file, with the NS metadata, constituent VxFDs and Virtual Link Descriptors
3. How to package the NS

Developers must create their descriptors according to the ETSI YANG model and the target orchestrator version (OSM in 5GinFIRE). Of course, as already said, they have the option to reuse existing publicly available VxFs, already uploaded to the 5GinFIRE repository, when building their network service descriptors.

Another important topic in the preparation of the experiment is that whenever developers decide to build their own artefacts, (e.g., when building their own VM images), these must be made available through the portal or any other public repository, for the end facility to download it and store it in its image repository.

A final consideration for the developer is the need of remote access to the VNFs while executing experiments; this implies that there is a need to incorporate an interface in the management plane to the VMs. The access to the instantiated VxFs and the running NS is performed through VPN, with credentials that are available from 5GinFIRE operations.

To cope with some of the difficulties of experimenting in 5GinFIRE, some validation mechanisms have been developed as value-added services.

4.1 Pre-validation Services

For those cases in which the VxF developer has no access to an instance of OSM, the consortium deployed the so called 5GinFIRE **mirror platform**. In that platform, parallel to the one in production, developers can validate that their implementations of VxFs and descriptors may actually be onboarded.

When this pre-validation goes well, the developer can proceed with the submission of the created artefact(s). Otherwise, the developer should fix any error regarding the VxF packages or descriptors, until the onboarding pre-validation is successful.

Also, for NSDs, 5GinFIRE has developed a format pre-validator, which developers can use to detect format errors in their descriptors. Using this mechanism, a pre-validated NSD will not necessarily comply with the desires of the developer (it may be a wrong NSD, not valid for the requirements), but will always be accepted by OSM.

This automatic pre-validation of descriptors brings to the table more ideas for innovation: if descriptors can be automatically validated, at least from a format point of view, how much would developers (i.e., different vertical sectors) appreciate an automatic descriptor development service?

Anyhow, in 5GinFIRE developers can always request support from partners to address any issues regarding VxFs, e.g., through the mailing list, Bugzilla, or the Slack channel. The request will be visible to all partners working on infrastructures and services, such that anyone can provide support to the questions.

4.2 Validation and Onboarding

When developers log in to the portal, they can manage their own already onboarded VxFs and NSDs. Figure 2 displays an example list of onboarded VxFs.

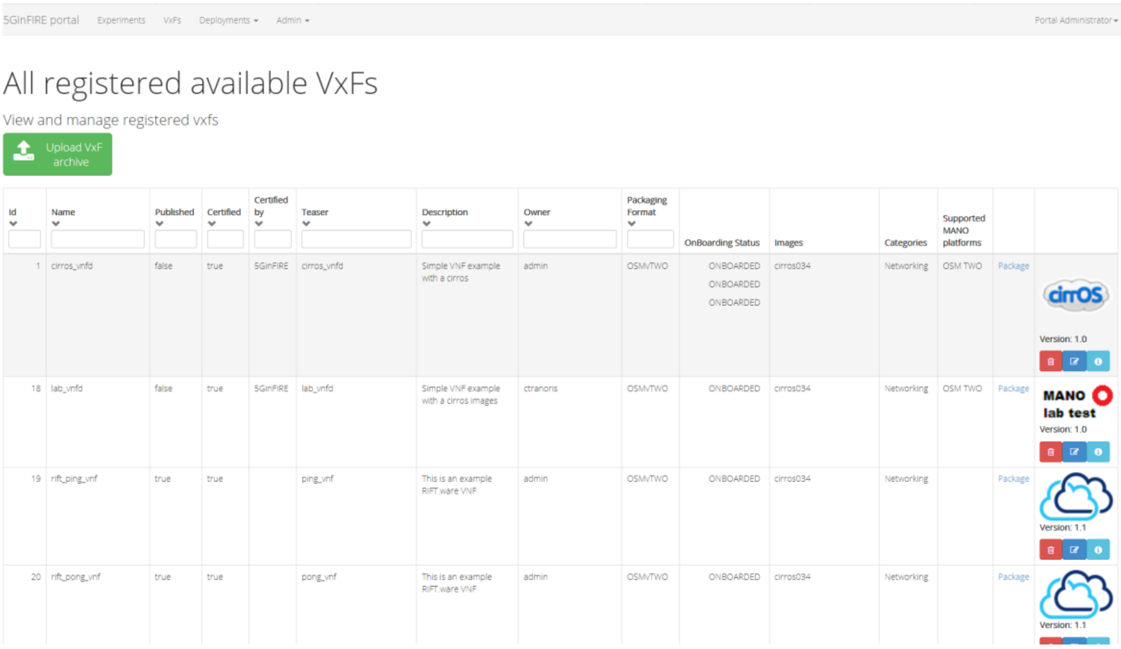


Figure 2: Available user artefacts (VxFs/NSDs) at the 5GinFIRE Portal

When the archives associated with these artefacts are ready, they can be uploaded into the Portal, the developer being able to choose target categories (e.g., “public”). Figure 3 presents the internal workflow for a VxF archive:

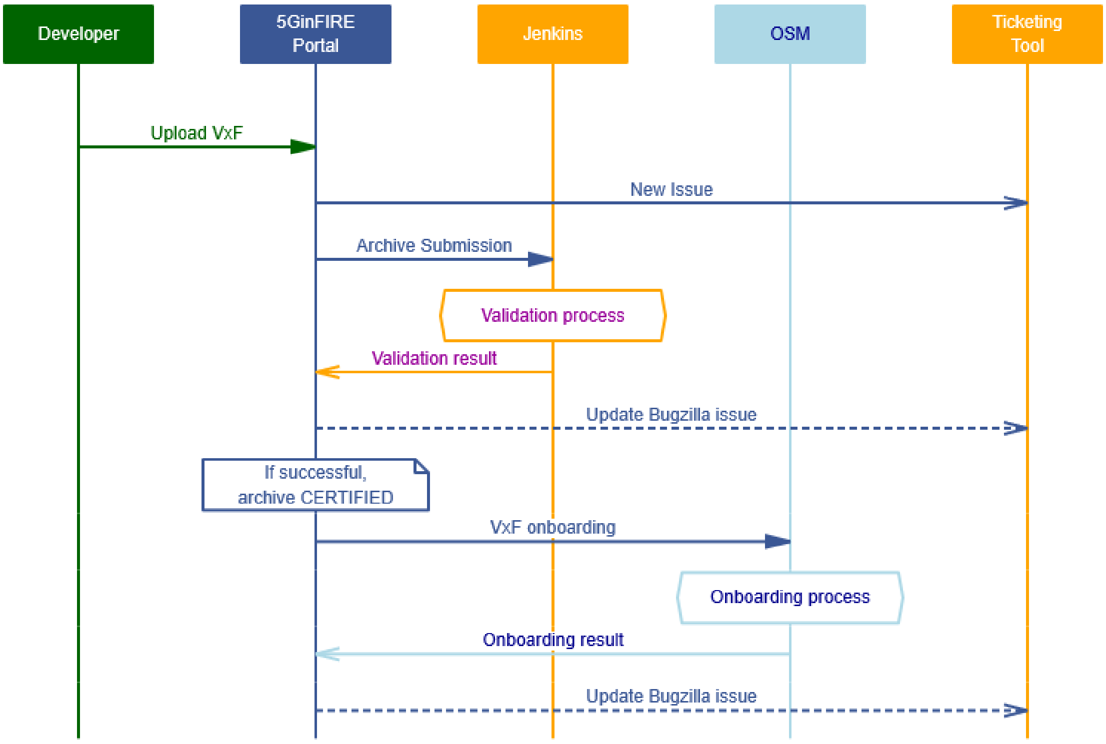


Figure 3: Validation and onboarding to OSM

After the upload, a ticket is created to continuously notify the developer and the platform operators about the onboarding progress. The rest of the automated process is as follows:

- The archive is submitted to the 5GinFIRE continuous integration validation service, based on Jenkins.
- Once this validation is finished, the portal gets the validation result and notifies both the developer and the platform operators by updating the ticket.
- If validation is successful, the archive is automatically onboarded to OSM.
- If onboarding is successful, the VxF is automatically marked as “Validated” in the portal repository and the ticket is closed as COMPLETED.
- Any error in the process makes the ticket to be closed as FAILED.

5 Experiment Deployment

Once an experimenter has developed and successfully onboarded the components[6] for the desired experiment (or has decided which of the publicly available experiments wants to execute), next step is the deployment of said experiment on top of the vacant infrastructure.

[6] These components are the experiment, described as an NSD (following the ETSI YANG model employed by OSM), plus all the VxFs that constitute the NS.

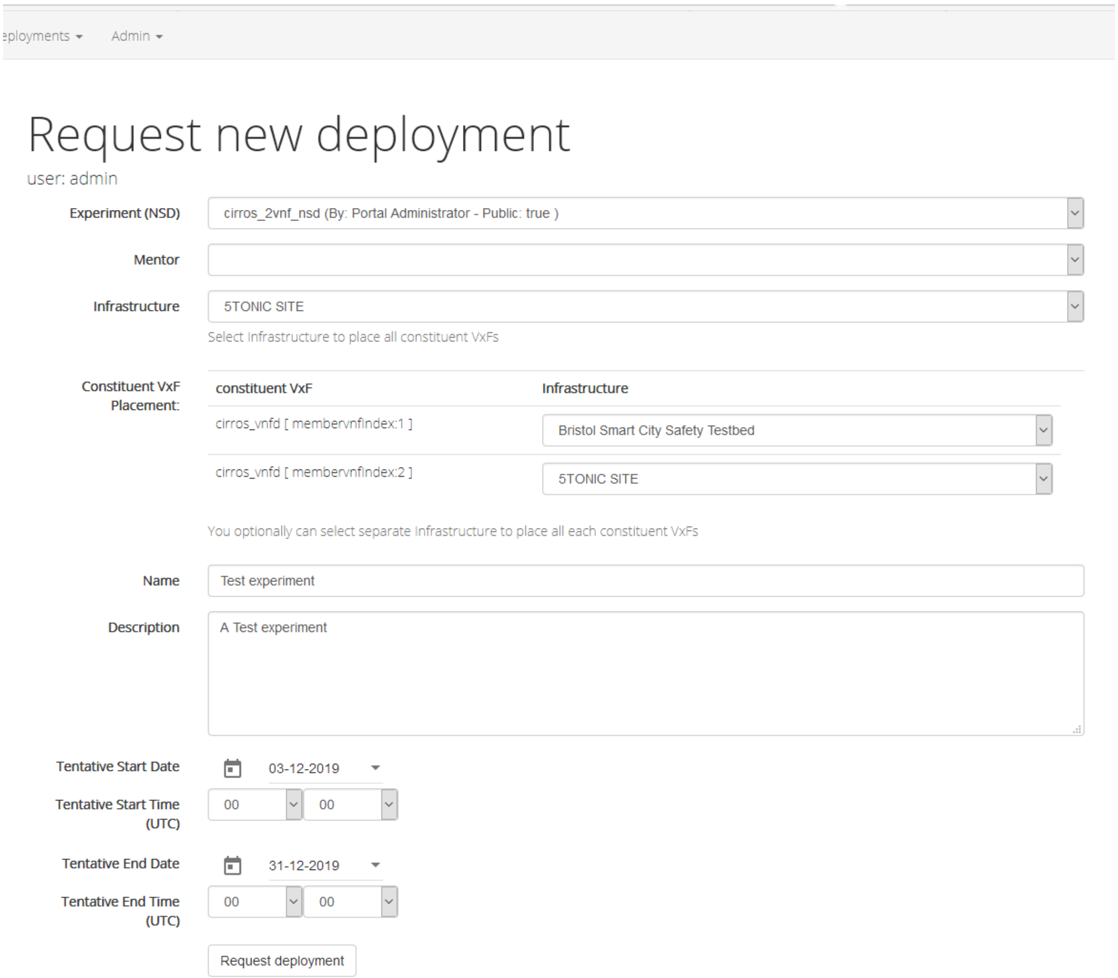


Figure 4: Experiment deployment request at the 5GinFIRE Portal

The experimenter can trigger the deployment using the 5GinFIRE portal, issuing a request as depicted in Figure 4. The request includes the desired NSD; the destination facilities (globally for the whole experiment, or on a per-VxF basis); the experiment mentor (which has been assigned by the project in advance); the tentative dates; and a name and description of the experiment, for logging purposes.

Once the experimenter submits the request, two actions are triggered: the creation of a new issue in the 5GinFIRE ticketing tool[7], which keeps track of the experiment throughout its whole lifetime, and a notification is sent to the experiment mentor (both by e-mail and by adding the mentor to the new ticket). This workflow can be followed in Figure 5. The initial state of a new experiment is always "UNDER_REVIEW".

[7] We implemented it using Bugzilla: https://www.bugzilla.org/

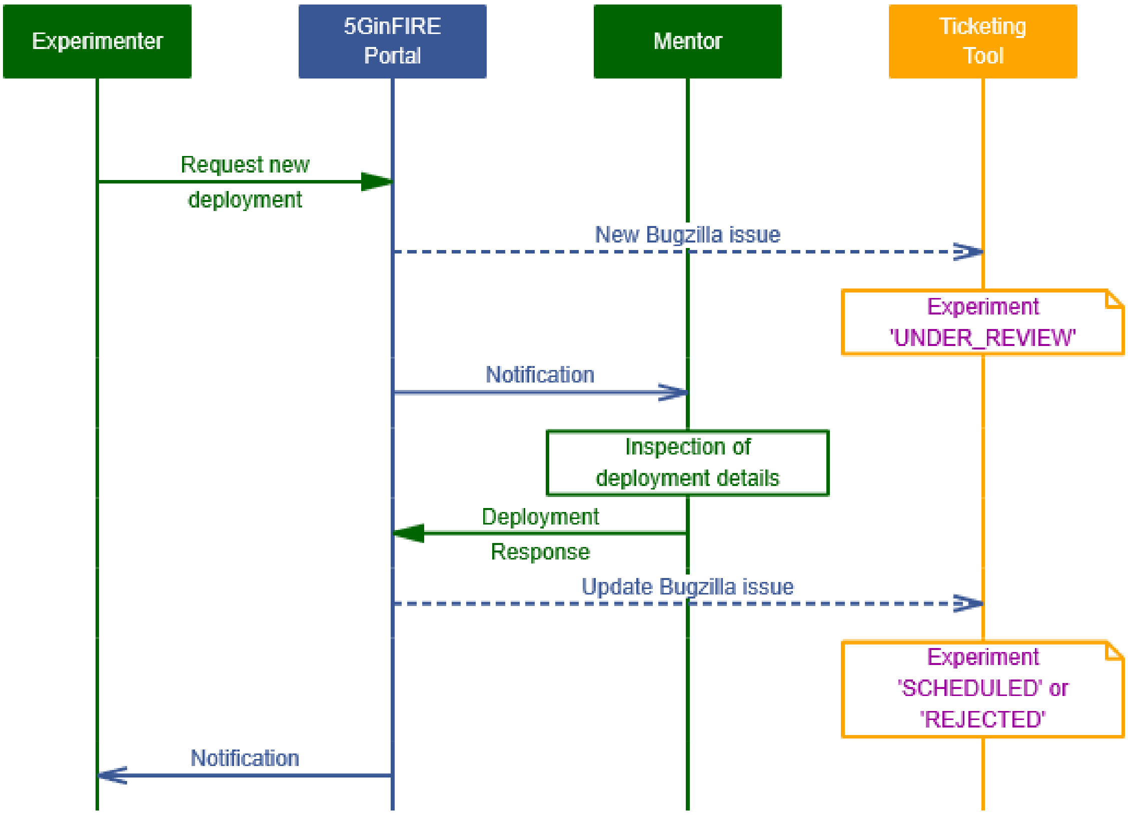


Figure 5: Experiment deployment request workflow

The mentor can then log in to the portal, and inspect the deployment details. Mentors deciding on a deployment request must make sure that:

- The requested experiment is **deployable**, i.e., that all the required components are already onboarded, and the sites to host the VxFs have the capacity to do so.
- The requested experiment **makes sense** in the selected locations, i.e., that the sites have the tools and capabilities to support the execution of the experiment. For example, an automotive experiment makes more sense in the automotive testbed than in the eHealth testbed.
- The requested **time slot** is valid for all the testbed providers involved in the experiment.

The mentor will be able to provide a response directly from the portal, either admitting the experiment ("SCHEDULED") or denying it ("REJECTED"). Independently of the response, the experiment status is updated by the ticketing tool. Mentors can also include in the generated ticket the reason for rejection, so this gives information to the experimenter on what needs to be modified if a resubmission is desired.

Figure 6 below depicts the full workflow of an experiment during its lifecycle.

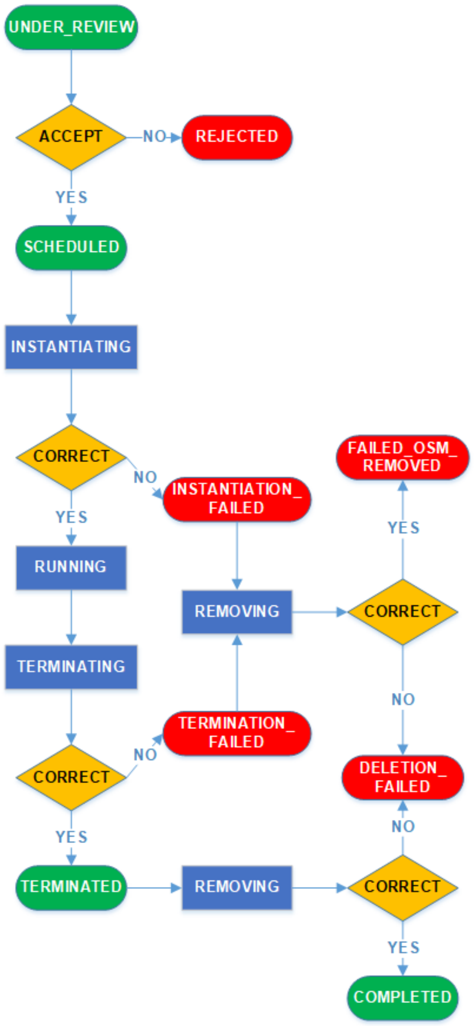


Figure 6: Full experiment deployment workflow

When a mentor approves a deployment, and the request reaches the status of "SCHEDULED", at the date/time the experiment is set to start the status automatically changes to "INSTANTIATING", and a NS instantiation request is sent by the portal to OSM.

If the instantiation succeeds, the status of the experiment changes to "RUNNING", which is when experimenters can execute their tests. If the instantiation, instead, fails, the status of the experiment goes into "INSTANTIATION_FAILED".

Once the experiment finishes, its status is changed to "TERMINATING", and a tear down of the NS is triggered. If successful, the experiment is then "TERMINATED", although its deployment container still remains in OSM. If failed, the status becomes "TERMINATION_FAILED".

In any of these correctly terminated or failed states, once the deployment end time is reached, the removal of the deployment from OSM is automatically triggered by the portal.

This leads to the three final states, which are: "TERMINATED" if the experiment and the removal were successful; "FAILED_OSM_REMOVED" if the experiment failed, but it was correctly removed from OSM; and "DELETION FAILED", if the removal from OSM failed.

All these states are of course reflected in the Bugzilla ticket associated with the experiment.

6 Full Testing Workflow

Having already described the experiment deployment workflow in the previous section, the workflow describing the full testing procedure (depicted in Figure 7, in the next page) is very straightforward. To reduce the complexity, in the workflow below only the positive deployment and tear down operations are considered.

As seen, the workflow is focused on the role of the experimenter, and does not consider the role of VxF developer. An experimenter willing to execute some test over the 5GinFIRE infrastructure would first create an account in the platform, request which would be evaluated and authorized by the system administrators (in this case, at the University of Patras). The only difference in case a VxF developer also wanted to execute an experiment would be that the VxF developer should already have an account, so this step could be skipped.

The first process (the one in the loop) corresponds to the experiment deployment. Once it is accepted by the mentor, then the experiment goes into “SCHEDULED” state, as we know from section 5, and waits until the start time and date.

Next operation corresponds to the “INSTANTIATING” phase of Figure 6. The portal automatically sends a request to the OSM (via NBI) to instantiate the NS, operation which is done in collaboration with the local VIMs from the selected sites. When (positively) finished, the portal updates the status of the experiment via Bugzilla to “RUNNING”, so it is therefore automatically advertised to both experimenter, mentor and system administrators.

As already said, test execution or results validation is not done automatically in 5GinFIRE. The alternative is admitting experimenters to connect (via VPN, and then SSH) to their deployed VxFs, for which we reuse the project management network. Experimenters request VPN credentials using the ticketing tool, credentials which are then provided by system administrators, in this case at 5TONIC. The VPN permits experimenters connecting to a jump machine at 5TONIC which allows connectivity to the VxFs management interfaces, but which drops all packets towards the rest of the infrastructure connected to this network.

At this stage, experimenters can execute their tests, with the support from their mentors (and of course testbed providers) and the supervision/monitoring of the infrastructure by the system administrators (and the 5GinFIRE Operations group).

It must be noted that the ticketing tool is always open for any issue experimenters may want to share with any of these actors.

Finally, the last part included in the workflow of Figure 7 shows the “TERMINATING” and “REMOVING” processes from Figure 6. As it can be seen, both processes are also automatic, upon the arrival of the end time.

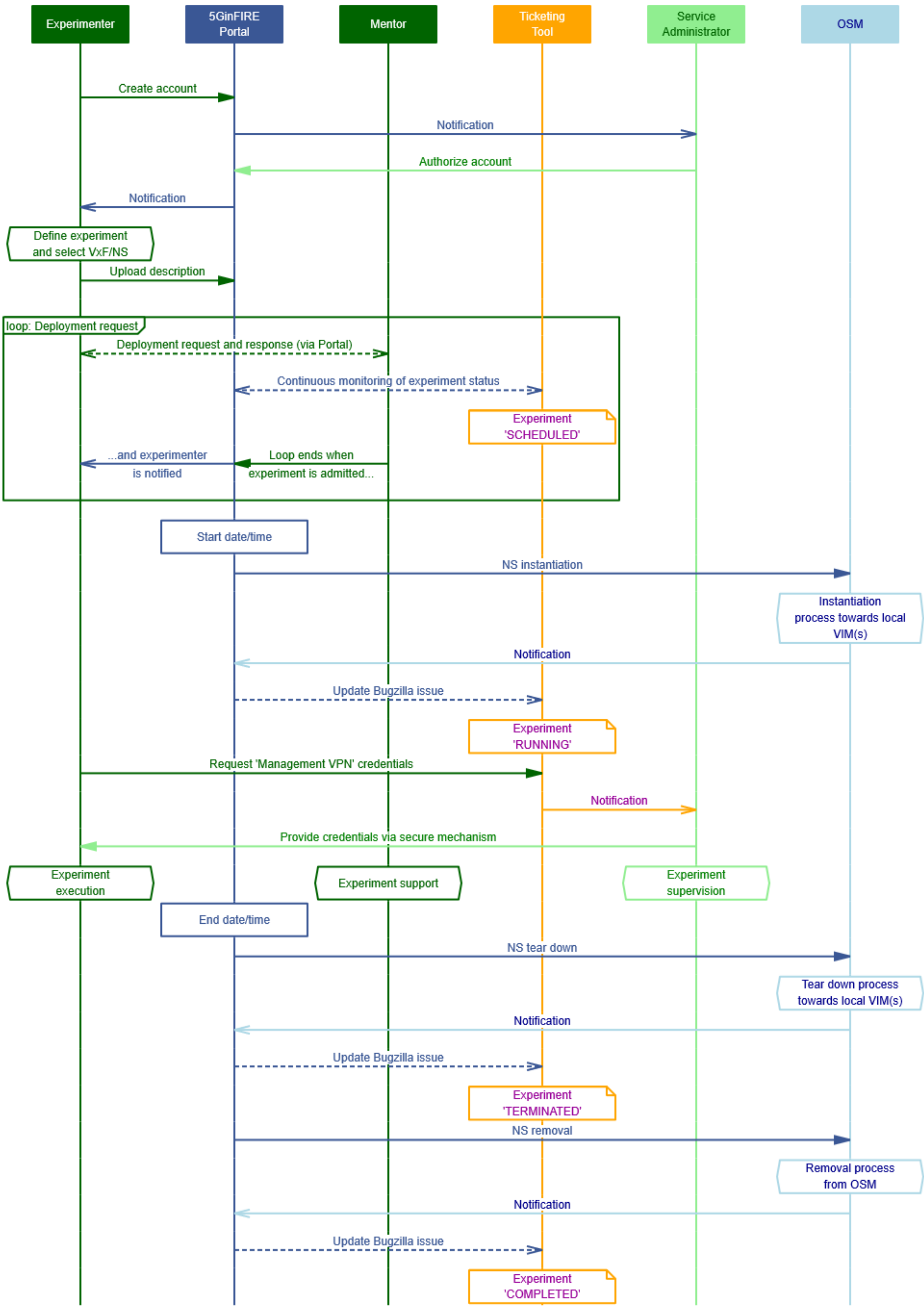


Figure 7: Full testing workflow

7 Conclusions

By all means, 5GinFIRE's original goal to provide an orchestrated, multi-site NFV-based platform for third-party experimentation on 5G applications can be considered a success story. The number of granted experiments in the open calls (25+), and the huge implications that this number has had in the project development and operational teams, put 5GinFIRE partners in an excellent position to share a good amount of experience and knowledge; this has been the objective of this whitepaper.

While it is impossible to include all the valuable insight in such a small document, hopefully readers can find and pick up interesting ideas here and there, and can work on improving those topics in which they believe 5GinFIRE may have not reached the standards they would require.

One of the biggest lessons learnt is that, generally, application developers and network providers do not speak the same language. Operators have a great opportunity to attract new 5G-based businesses, as they will be able to offer infrastructure compliant with the strict latency requirements that are being demanded. But with business cases that are still not clear for vertical service providers, network operators cannot expect them to invest a huge amount of effort in adapting their testing procedures. This has to be a joint endeavour, and all the work destined to simplify what is demanded from verticals will revert into good relationships, efficient testing and, ultimately, costs savings.

It is also important to mention that each testing facility may have different objectives. Tools and procedures for technology benchmarking may look totally different than those for testing of services and applications. In that sense, each facility must define very clearly what **testing services** are being offered, what **tools** are available for experimenters, in support of those services, and of course what are the **procedures** that will enable the execution of tests on top of the offered infrastructure. Prioritizing core testing services from value-added services will enable the facility to host experiments earlier, which is a key demand nowadays.

Another key demand is automation in the testing procedures. Benefits of automation are numerous and very obvious, but it must be accepted that the idea of full automation at all stages is just not feasible. It may happen that some processes or services can be designed to be automated from the very beginning, but generally, automation must be incorporated gradually. Indeed, the logic to automate some processes may be so complex that it may not make sense to delay certain functionality just to make it automatic. In general, *let the feature bring the desired added value quickly, and then generate even more value by automating it*.

Continuing with automation, it is not always a good approach to start from those processes that consume more time. It may take several weeks to fully understand an experiment, but it is done only once; you may save 3 or 4 hours by implementing an automated deployment, but each experimenter may deploy one experiment 20 or 30 times, even before they are ready to start testing. For example, in 5GinFIRE we have had 25+ experimenters and 500+ deployment requests. Thus, analysing the correct approaches for incorporating automation may pay off very quickly. Again, in 5GinFIRE, where the support of the operational team was provided at a best-effort level, automating all those tasks that depended from this team proved to be a wise move (e.g., the automatic onboarding of VxFs/NSDs in OSM after the validation process, not waiting for an explicit OK from the operational team).

To conclude, this whitepaper provides detailed workflows (or appropriate references) for procedures that we believe may be of interest for all those willing to offer similar or related testing infrastructure to third parties. These include the development and onboarding of VxFs and NSDs, the deployment of experiments, and finally the testing workflow. The whitepaper has tried to stress in all the descriptions the important role of the 5GinFIRE portal, as single interface towards experimenters, in this case, our customers.

It is obvious that readers may not find all the included concepts applicable, and we have also made an effort to point out potential improvements. However, hopefully, we have generated a varied enough whitepaper as to permit a wise cherry-picking that enhances testing facilities to come.

References

[1] http://5GINFIRE-5g.eu/ © 5GINFIRE consortium 2019

[2] Juan Rodriguez. "How can OSM accelerate 5G Vertical Services" – *OSM & 5G Verticals* session's keynote, at *7th OSM Hackfest* – Patras (Greece), September 2019

Slides available at: http://osm-download.etsi.org/ftp/osm-6.0-six/7th-hackfest/5G%20Verticals/5G%20Verticals%20Keynote%20-%20Juan%20Rodriguez%20Telefonica.pdf

[3] 5G PPP *Test, Measurement, and KPIs Validation* Working Group Whitepaper: "Validating 5G Technology Performance – Assessing 5G architecture and Application Scenarios" – June 2019

Available at: https://5g-ppp.eu/wp-content/uploads/2019/06/TMV-White-Paper-V1.0-03062019.pdf

[4] Borja Nogales; Ivan Vidal; Diego R. Lopez; Juan Rodriguez; Jaime Garcia-Reinoso, Arturo Azcorra. "Design and Deployment of an Open Management and Orchestration Platform for Multi-Site NFV Experimentation", IEEE Communications Magazine (Volume: 57, Issue: 1, January 2019)

DOI: https://doi.org/10.1109/MCOM.2018.1800084

[5] http://wiki.5ginfire.eu/

Abbreviations

KPI	Key Performance Indicator	**SSH**	Secure Shell
MANO	Management & Orchestration	**VIM**	Virtualized Infrastructure Manager
NBI	North-Bound Interface	**VLAN**	Virtual Local Area Network
NFV	Network Functions Virtualization	**VM**	Virtual Machine
NS	Network Service	**VNF**	Virtual Network Function
NSD	Network Service Descriptor	**VPN**	Virtual Private Network
OSM	Open Source MANO	**VxF**	Virtual Function (in general)
SR-IOV	Single Root I/O Virtualization	**YAML**	Yet Another Markup Language