TNOVA

NETWORK FUNCTIONS AS-A-SERVICE
OVER VIRTUALISED INFRASTRUCTURES

GRANT AGREEMENT NO. 619520

Deliverable D1.4

Final Report

Editor A. Kourtis (NCSRD)

Contributors G. Xilouris (NCSRD)

Version 1.0

Date December 31st, 2016

Distribution CONFIDENTIAL (CO)

Table of Contents

1. INTRODUCTION

This document is intended to be a summary of the activities performed in each of the T-NOVA work packages and tasks throughout the entire project period. It reflects the overall work done in all the project phases and activities, as well as the key achievements.

Essentially, this report consolidates the information provided in the three project periodic reports in a single, consistent report focusing on the final outcomes of the tasks rather than the intermediate steps.

The chapters to follow correspond to each one of the project work packages. It must be noted that all project milestones were appropriately met, and all the project objectives, as described in the DoW, were completed successfully.

2. WP1. PROJECT ADMINISTRATION

2.1. T1.1. Project Management

Project management tasks throughout the project can be summarized as follows:

- Management of the communication inside the project consortium and towards the European Commission
- Creation of a Consortium Agreement fulfilling the specific needs of all partners, discussions with the lawyers' departments of partners, iterations and final signature
- Edition of a set of templates for T-NOVA deliverables and presentations
- Liaison with the External Advisory Board (EAB) members
- Submission of deliverables to the EAB members (when needed) and gathering their opinions
- Reviewing the deliverables to ensure quality
- Management of the pre-financing transfers to the partners
- Preparation and revision of QMRs
- Creation and maintenance and update of the project's web site
- Attending EC events (concertation meetings, EuCNC, EU Net-Tech Future Coordination Meeting)
- Organizing T-NOVA related publicity activities and representing the project to external entities and organisations
- Organisation, preparation and management of plenary and technical meetings as well as preparation of meetings.
- Preparation of a large number of audio-conferences (3-4 per week) and preparation of the resulting agreed minutes

External Advisory Board (EAB)

According to the GA, an External Advisory Board (EAB) had to be created. Candidates were proposed by all partners and through an open selection process based on opinions of all partners, three members were selected : Gonzalo Camarillo (Standardisation Director – ERICSSON), Shahar Steiff (AVP of business operations – PCCW GLOBAL) and Steven Wright, (NFV & SDN Industry Engagement Standards & Industry Alliances - AT&T). An External Advisory Board (EAB) Agreement was created and signed between each member of the EAB and the PM.

2.2. T1.2. Technical Coordination

Throughout the project, the Technical Coordinator closely supervised the progress of the project in the Scientific/Technical sector and coordinated the effort in all tasks involving system design and development. He also reviewed and validated the technical deliverables. Design and Implementation plans were aligned to the related standards (ETSI NFV ISG, IETF-NFVRG, OPNFV, OPENSTACK TELCO WG) and were monitored in order to verify technical soundness and efficiency.

More specifically, the work in T1.2 included:

- scientific/technical coordination at the project level;
- scientific/technical coordination between WPs;
- Supervision and insurance of the scientific/technical high level of the project work, deliverables and reports;
- Supervision of the technical flow of information within the project;
- Quality Assurance: to assure that the technical process follows the quality rules for the project.
- Setting up and follow-up of tasks and objectives related to the Technical Committee.

In order to perform the above tasks, and monitor the progress of work in all WPs and Tasks the following tools and means were used :

- Creation and maintenance of the T-NOVA Wiki, based on MediaWiki
- Teleconference meetings 3-4 times per week, based on commercial web conferencing platforms (Webex initially and later Gotomeeting)
- Organization of a technical meeting for WP3/WP4
- Presentation of work progress to the EAB members, either during their participation in plenary meetings (where special time slots were allocated for EAB) or by sending them specific deliverables and elaborating on their comments.

3. WP2. System specification

3.1. T2.1. System Use Cases and Requirements

T2.1 focused on the specification of use cases and requirements, the definition of basic T-NOVA stakeholders, as well as the description of several business-oriented application scenarios and associated value chain, to illustrate how the deployment of T-NOVA is envisaged in practice.

This work followed a use case-driven approach, as recommended by requirements engineering best practices, starting with the identification of stakeholders, business models, basic use cases, and then evolving to the specification of requirements.

The overall methodology followed in T2.1 is illustrated in the figure below.

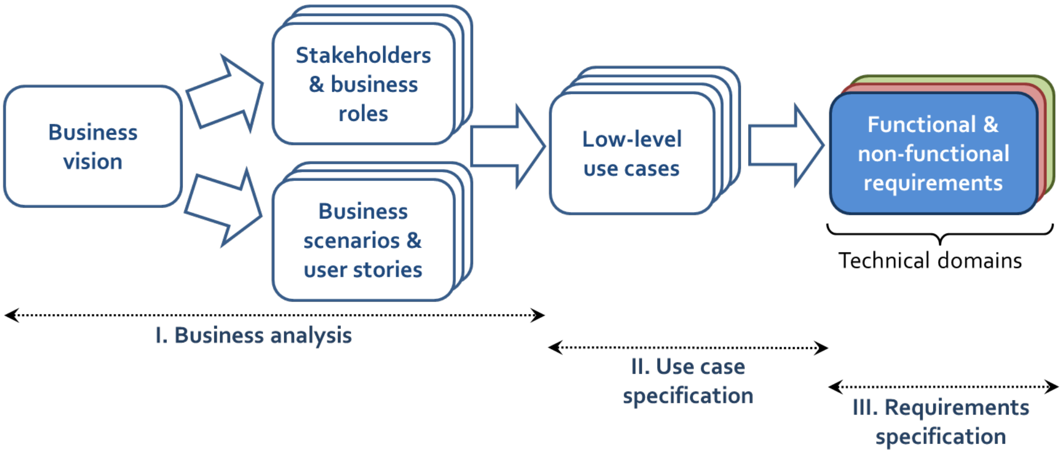


Figure 1 Overall Task 2.1 methodology

The results of T2.1 are provided in D2.1. This deliverable establishes a common ground on which the remaining WP2 tasks (T2.2 to T2.6), and later technical WPs (WP3 to WP6), are expected to build their foundations. Most significant results provided in D2.1 are:

- Description of several representative application scenarios built around the T-NOVA ecosystem. Two high level scenarios, illustrating the use of T-NOVA in two different environments – enterprise and residential – have been described.
- Definition of the Use Cases foreseen for T-NOVA system. Six use cases, supposed to describe the sequence of interactions that take place between the T-NOVA system and the involved stakeholders to achieve some outcome of value, have been defined.
- Identification of T-NOVA system business roles and value chain, considering their potential interest in the adoption of TXNOVA. Three basic stakeholders have been identified and characterized (Service Provider, Function Provider, Customer) as well as a few optional stakeholders (e.g. broker, service integrator, network infrastructure provider, Cloud infrastructure provider, end user)
- A collection of functional and non-functional system requirements considering all business actors and identification of their role in the T-NOVA ecosystem. The

requirements specified stem from the various use cases. Requirements address several thematic areas, as follows: management and orchestration, elasticity, security, resiliency, service continuity, operations and market / commercial operability.

3.2. T2.2. Overall System Architecture

The effort in WP2 Task 2.2, was focused in the following tasks:

- Survey of currently proposed NFV architectures, including industry initiatives (CloudNFV, HP OpenNFV, Intel/Tieto, ALU CloudBand), research projects (Mobile Cloud Networking, CONTENT, NetIDE, UNIFY) as well as the current specifications, drafts and trends in ETSI NFV ISG.
- Definition of T-NOVA High-level architecture, seeking compliance with the current ETSI NFV ISG vision. Assembling all the basic functional blocks, a high-level view of the T-NOVA system architecture was derived, as shown in the Figure 2.

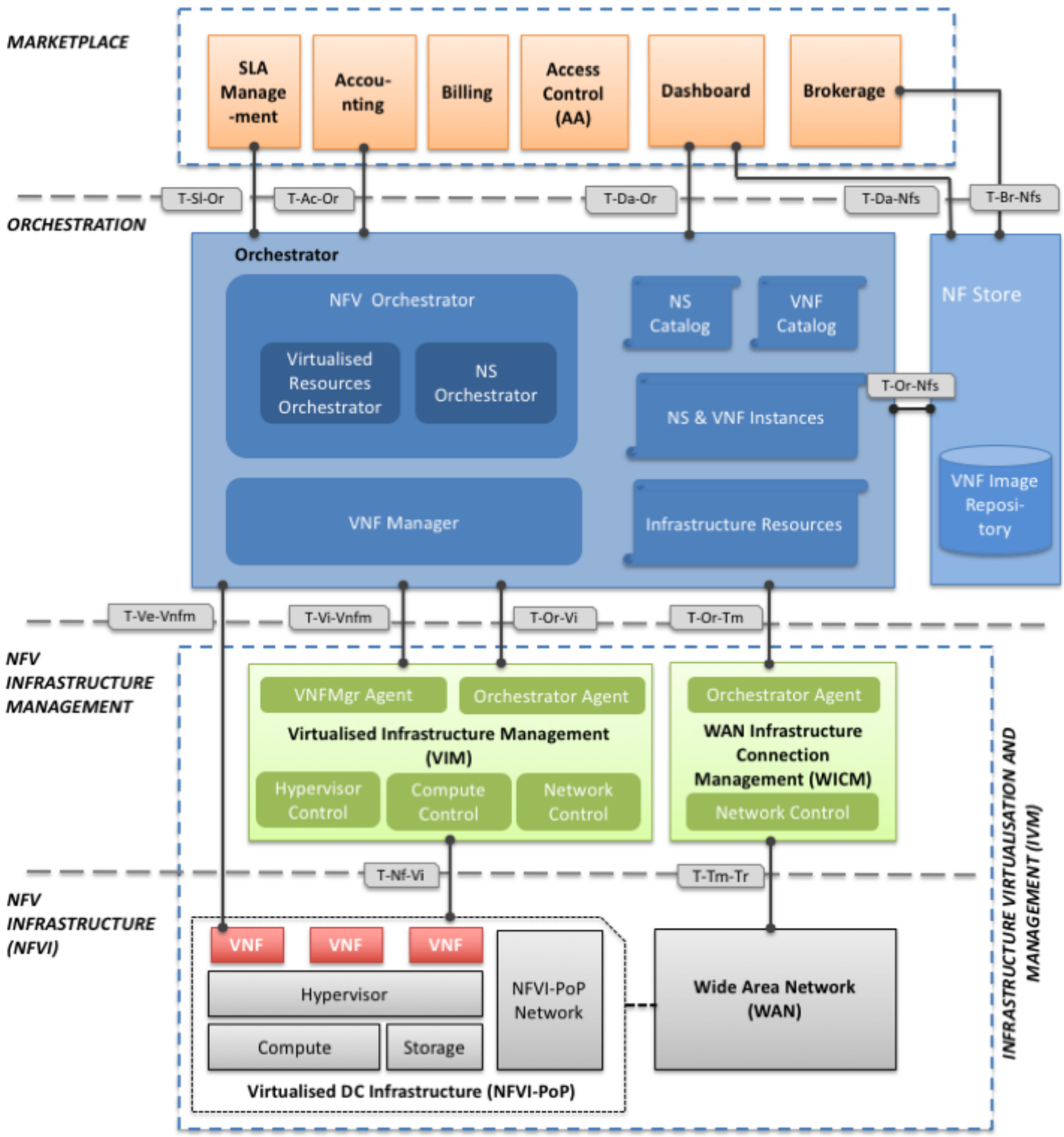


Figure 2 High-level view of overall T-NOVA System Architecture

The T-NOVA architecture can be hierarchically organised into four architectural layers:

- The NFV Infrastructure (NFVI) layer includes the physical and virtual nodes (commodity servers, VMs, storage systems, switches, routers etc.) on which the services are deployed.
- The NFVI Infrastructure (NFVI) Management layer comprises the infrastructure management entities (VIM, WICM). In the sections as well as the deliverables to follow, the NFVI and management layers are conceptually grouped under the name Infrastructure Virtualisation and Management (IVM).

- The Orchestration layer is based on the T-NOVA Orchestrator and also includes the NF Store
- Finally, the Marketplace layer contains all the customer-facing modules which facilitate multi-actor involvement and implement business-related functionalities.

Work also included:

- Sequence of interaction between subsystems for each use case (as defined in D2.1) using UML notation.
- Discussion of several architecture variations i.e. : interfacing with a BSS/OSS, multi-provider NFVI-PoPs, support of legacy non-SDN DC infrastructures, support of multiple T-NOVA SPs with a single broker, and multi-SP federation.

3.3. T2.3. Specification of the Orchestrator Platform

T2.3 was responsible for the specification of the orchestrator platform. The main activities and accomplishments by Task 2.3 were:

- Analysis on the state-of-the art of orchestrators (standardisation bodies, main vendors and open-source solutions);
- Specification of the orchestrator (internal and external) design requirements;
- Specification of the orchestrator architecture and interfaces;
- Specification of the orchestrator main workflows (on-boarding, provisioning, monitoring, scaling and termination); definition of sequence diagrams for T-NOVA use cases developed in cooperation with task 2.4;

In terms of content, the T-NOVA Orchestrator addresses the key virtualised network functions (VNFs) as well as Network Services (NSs), which must be supported by the T-NOVA architecture.

This architecture has been modelled, taking into account a working stage 1/stage 2 methodology, which departed from the elaboration of a list of Orchestrator requirements identified after a research study involving several sources, e.g. use cases defined in deliverable D2.1, ETSI ISG NFV requirements, ITU-T requirement for NV, as well as excerpts of relevant parts of the ETSI ISG MANO WG architecture and associated FEs (8). After that stage 1 step, a stage 2 methodology took place with the derivation of the stage 2 reference architectures and its Functional Entities.

The reference architecture has been interrogated and validated at a functional level through the development of NS and VNF workflow diagrams, which illustrated the key actions and interactions within the T-NOVA system during standard operational activities related to the deployment and management of NS and VNF services.

The Orchestrator reference architecture, as well as the interfaces with the external Functional Entities (FEs) is depicted in the figure below. In detail, the orchestrator interacts with the Marketplace, which is the T-NOVA domain responsible for accounting, SLA management and business functionalities. Besides the Marketplace, the Orchestrator also interfaces with the IVM, and in particular with the VIM, which in the T-NOVA scope is located in the IVM domain, for managing the data centre network/IT infrastructure resources, as well as with the Wide Area Network Interconnection Manager (WICM) for WAN connectivity management. Finally, the Orchestrator interacts with the VNF itself, which in the T-NOVA scope is also located in the IVM domain, to ensure its lifecycle management.

Internally, the T-NOVA Orchestrator consists of two main components and a set of repositories. One of the core elements is the NFVO, acting as the front-end with the Marketplace and orchestrating all the incoming requests towards the other components of the architecture. To support the NFVO operation procedures, a set of repositories is identified in order to store the description of the available VNFs and NSs (VNF Catalogue and NS Catalogue), the instantiated VNFs and NSs (NS & VNF Instances), as well as the available resources in the virtualised infrastructure (Infrastructure Resources Catalogue). Finally, the NFVO also interacts with the other core element, the VNF Manager (VNFM), responsible for the VNF-specific lifecycle management procedures.

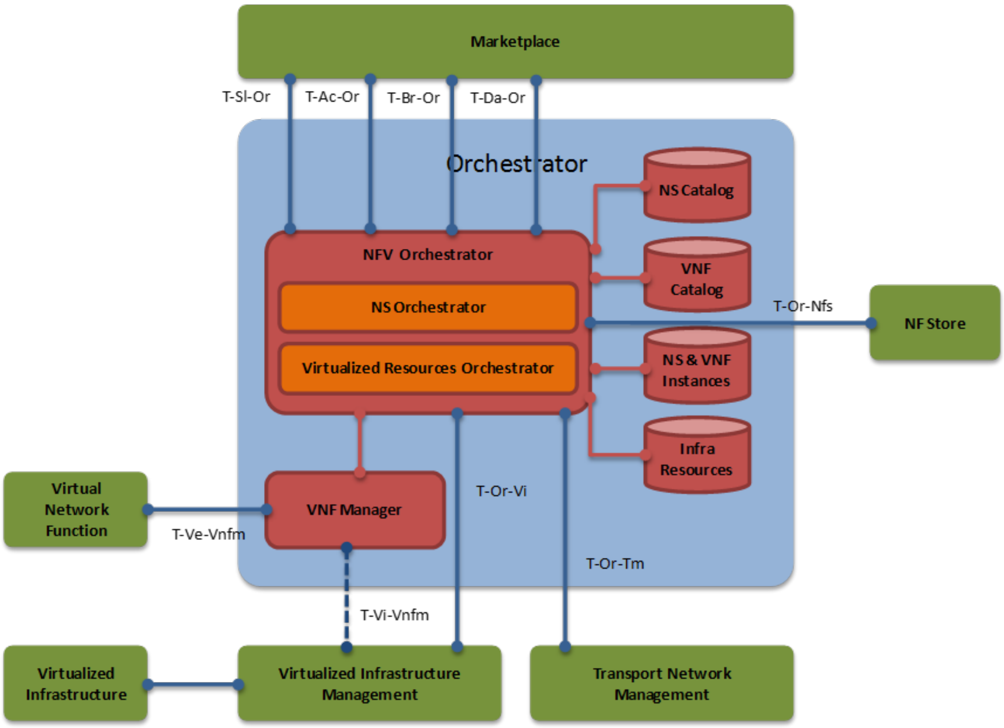


Figure 3 Detailed Orchestrator architecture

3.4. T2.4. Specification of Infrastructure Virtualisation and Management

Task 2.4 was focused on the overall integrated architecture for the Infrastructure Virtualisation Layer (IVM) together with the architecture of the various domains that comprise the IVM with their respective internal and external interfaces. Additionally, the task was focused on identifying the key goals, objectives and requirements for the IVM and constituent functional entities. The key activities and outputs of this task are as follows:

- The task 2.4 partners completed an extensive state of the art review encompassing a wide range of software and hardware technologies that were relevant to the Infrastructure virtualisation management (IVM) layer of the T-NOVA architecture and its functional entities namely the VIM and NFVI. Technologies reviewed and analysed including hypervisors, compute virtualisation technologies, virtual infrastructure management, cloud computing environment and virtualised network components such as SDN controllers, protocols, control plane approaches and API's etc.
- The partners collected the key learning's based on partner's experiences of different SDN controllers/framework including open source and commercial such as OpenDaylight, RYU, POX etc.
- A system engineering approach was adopted to define the key functional blocks within the IVM and their interfaces. In addition the key objectives for the IVM were defined. The architecture for the T-NOVA IVM layer and its composite functional entities was developed which aligned with the ETSI MANO architecture.
- Analysis of the requirements for T-NOVA IVM layer identified the need for a Wide-Area Interconnection Manager (WICM) to provide and manage inter data centre connectivity under the control of the Orchestrator as shown in Figure 4. This functional which is not included in the ETSI MANO architecture was added to the IVM. The key internal and external interfaces of the IVM identified and documented.
- Requirements for the infrastructure virtualisation and management layer and it functional entities (virtual infrastructure manager, transport network manager, compute, infrastructure network and compute) were identified, refined and documented. In total 70 unique requirements were identified and documented.
- A number of specific interfaces were identified which provide both the internal and external connectivity that integrates the various technology components into a functional IVM.
- Workflow sequence diagrams for T-NOVA use cases developed in cooperation with task 2.3 which were used to validate the IVM architecture developed by the task. The sequence diagrams were used to show in the key interactions with T-NOVA infrastructure components including the VIM. NFVI and WICM during a VNF's service lifecycle.
- Considering both existing industry oriented initiatives and currently available technologies that are being used commercially (or are in a development stage) a focused gap analysis was carried out to determine what steps need to be taken in order to move NFV/SDN from its current state to a position that can fully realise the needs of key areas that need to be addressed during the project activities.
- Joint deliverable (D2.31) with task 2.3 completed and submitted on schedule. The deliverable described the architecture for IVM and its composite functional entities, the key requirements for the IVM functional entities and their internal and external interfaces.

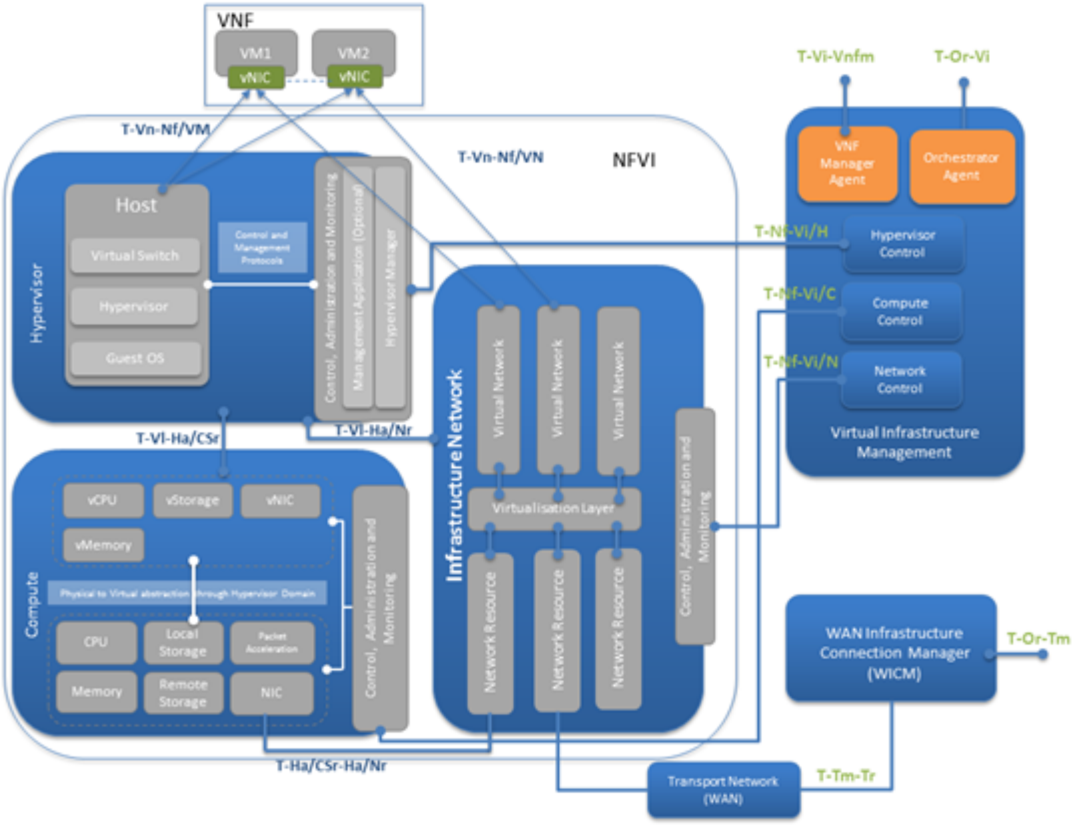


Figure 4 T-NOVA infrastructure virtualisation and management (IVM) high level architecture

3.5. T2.5. Specification of Network Function framework

Task 2.5 was dedicated to the Network Function Framework, that is the architectural element of the T-NOVA system devoted to the definition of the structure and behaviour of the Virtual Network Functions (VNFs). Moreover, in this task we worked also to the T-NOVA Network Function Store, which is the architectural component where the VNFs are stored and made available to T-NOVA as building blocks for creating network services.

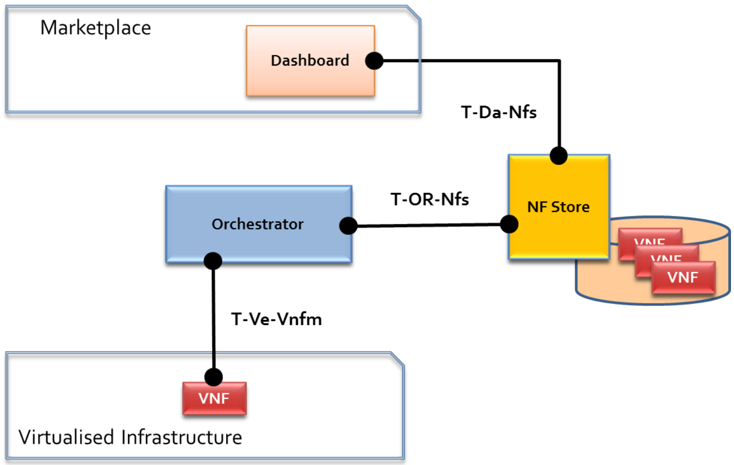


Figure 5 VNF framework high level architecture

In relation to the specification of the Network Function Framework two main tasks have been performed:

- the description and the specification of the VNFs
- the design of the NF Store.

The first one includes the APIs allowing the VNF to be managed by the T-NOVA system as well as the information elements that shall be present in the VNF metadata; a key piece of data of the Network Function Framework is the metadata descriptor associated with the actual software implementation of the VNF. Besides the structural definition of a VNF, its behaviour has been studied defining a lifecycle that is common to all the VNFs in T-NOVA. This lifecycle can be split in an off-line and on-line part. In relation to the on-line one, the lifecycle states where the VNF is up and running over a virtualised execution platform. On the other hand, the off-line lifecycle states span from the software implementation of the VNF to its uploading into the NF Store that can be thought as the place where the VNFs are stored. The NF Store APIs provide interfaces by means of the dashboard with Function Providers for uploading, updating, and withdrawing the VNF software images and metadata description, and interface with the rest of the T-NOVA system for making this information available for service orchestration.

The summary of main results is:

- Design of VNF common components.
- Study of Network Function Lifecycle.
- Specification of NF APIs.
- Design of the Network Function Store.

In the scope of T-NOVA project, a VNF is a virtual machine or a group of virtual machines that implement network functions in software. The VNF high level architectural model is represented in Figure 6. A VNF is characterised by two main attributes: the operational functionalities and the management behaviour. The operational part explicitly defines the network functions that are supported, while the management part is responsible for the VNF lifecycle (i.e. start, stop, pause, scaling, etc.).

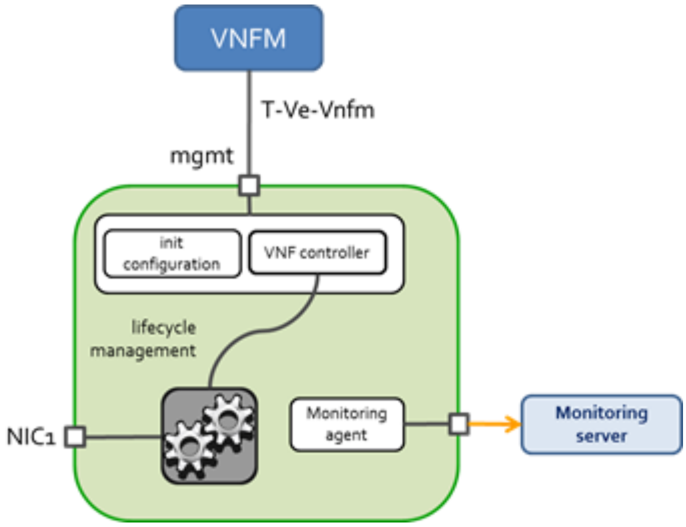


Figure 6 VNF high-level structure

Each VNF shall support a set of API operations for automating its operations in T-NOVA system. The VNF lifecycle is schematically represented in We detailed all the states and the expected behaviour of a VNF, also defining the API interface a VNF shall support.

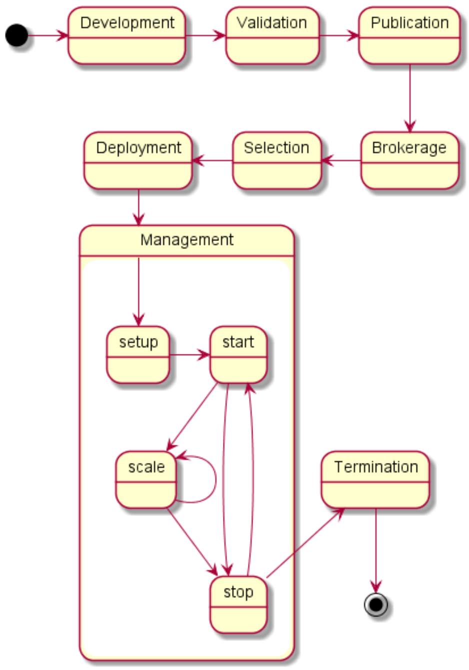


Figure 7 VNF lifecycle

We recognize that ETSI NFV as an important reference for the virtualization of network functions. Therefore we analysed the different approached between our work in T-NOVA and the ETSI standards. We found that the lifecycle management of VNFs and related packages adopted in T-NOVA is mostly aligned with the one adopted in ETSI. However, some differences exist in publication and brokerage, which are T-NOVA specific. In fact, T-NOVA target is an open market environment, while ETSI focus is limited to traditional telco operations.

An important activity in Task 2.5 was the definition of the VNF descriptor information model. The VNF metadata descriptor provides the information for describing what the VNF functionalities are and how to manage them. We analysed many available examples of metadata templates that provide the technical information for configuring, running and operating the VNF. Then we extended this information set with business relevant aspects to allow the registration and trading of a VNF in the marketplace. The detailed definition of T-NOVA VNF descriptor is on-going activity involving across all the project work packages.

Finally, we defined the high level design and requirement for the Network Function Store that is mainly a repository for the VNFs and their metadata. It contains the virtual machines (VMs) software images and the metadata descriptor composing each VNF.

3.6. T2.6. Specification of the T-NOVA Marketplace

T2.6 was responsible for the specification of the T-NOVA Marketplace. The main activities and accomplishments by Task 2.6 were:

- State of the art analysis about marketplaces with a similar approach.
- T-NOVA Marketplace lifecycle description.
- Marketplace components requirements gathering.
- Definition of the Marketplace internal architecture interfaces.
- Interaction with other tasks, mainly T2.5 (VNF framework) and T2.3 (orchestrator) for the specification of the marketplace external interfaces.
- Marketplace specification completed: dashboard, brokerage module, Access Control Module, SLA management module, accounting, billing and business service catalogue.

Based on the requirements performed at system level in Deliverable 2.1 and the initial approach for the marketplace architecture included in Deliverable 2.2, T2.6 has gathered the specific requirements for each component in the marketplace, detailing the high-level T-NOVA architecture including both the external and internal interfaces as depicted in the following diagram (Figure 8).

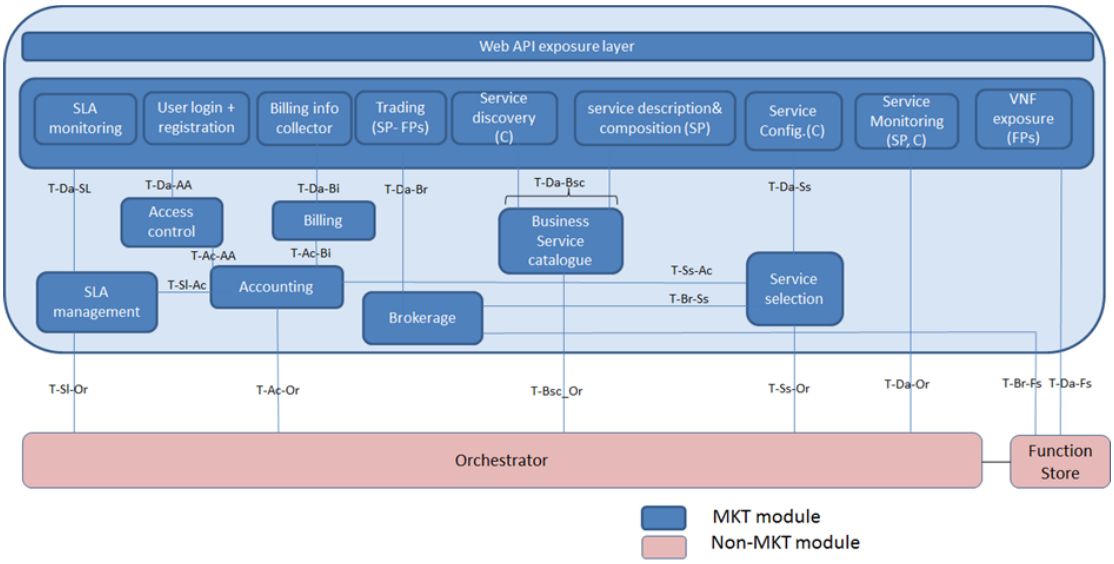


Figure 8 High-level Marketplace Architecture

3.7. T2.7. Pilot definition and Evaluation Strategy

In Task 2.7 the effort focused on the definition of overall plan for the validation and assessment of the T-NOVA system, to take place in WP7, mostly concentrated on end-to-end system-wide use cases. Precisely, Task 2.7 contributions involved the following topics:

- The assembly of a testing toolbox, taking into account standards and trends in benchmarking methodology as well as industry-based platforms and tools for testing of network infrastructures, including the guidelines drafted by ETSI for NFV performance benchmarking.
- Identification of the NFV environment challenges towards the definition of the overall T-NOVA evaluation strategy.
- Specification of performance evaluation metrics, including system-level metrics (with focus of the physical system e.g. VM deployment/scaling/migration delay, data plane performance, isolation etc.) as well as service-level metrics (with focus on the network service e.g. service setup time, re-configuration delay, network service performance).
- Specification of the experimental infrastructure including reference Pilot architecture (Figure 9) as well as definition of T-NOVA Pilots and testbeds.

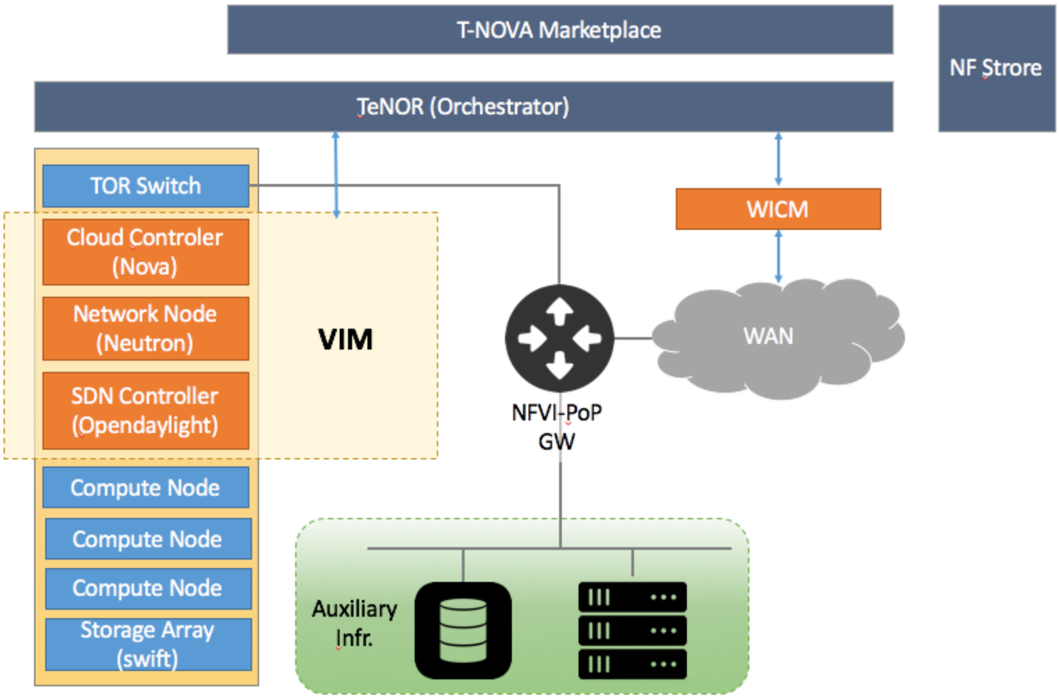


Figure 9 T-NOVA Reference Pilot architecture

- Definition of step-by-step methodology for validation of system use cases defined in D2.1 and specifying a step-by-step methodology –including preconditions and test procedure.
- Selection per VNF measurement tools, as well as a set of test procedures, defining the tools and parameters to be adjusted during test, as well as the metrics to be collected.

4. WP3. Orchestrator Platform

4.1. T3.1. Orchestrator Interfaces

Task 3.1 is focused on the (detailed) definition, design and implementation of the Orchestrator interfaces.

The main activities were:

- The study of the requirements and (high level) architecture of the Orchestrator;
- The definitions of the different kinds of interfaces needed, one emphasizing flexibility and the other resource usage efficiency and low latency;
- The study of the different formats, technologies and architectural styles that could support the high flexibility requirements needed for the definition of Network Services by composition of Virtual Network Functions (VNFs);
- The study of the different open-source tools that could support the non-functional requirements (high availability and low latency), a class of systems names Streaming Data Processing Systems: Apache Storm (initially from Twitter), Apache Spark Streaming (originally from UC Berkeley) and Apache Samza (originally from LinkedIn);
- The design and implementation of the Orchestrator interfaces.

The Orchestrator interfaces were designed in a REST-based architecture, using JSON as the format for interacting with it, over HTTP. An example of an interface (in this case, of the NS Manager module) is given in Table 2.

Table 1 Example of Interface definition

Endpoint	Header Fields	Method	Description
/network-services	X-Auth-Service-Key	POST	Registers the new NS in the NS Catalogue. The actual NS Definition (NSD) is part of the NS creation request, but is not the request in itself.
/network-services/:ns_id	X-Auth-Service-Key	PUT, DELETE	No specific use case was found for reading (GET operation) NSs that have already been registered in the Orchestrator.

The considered systems to which the Orchestrator needs to interface with are represented in the picture below (Figure 10).

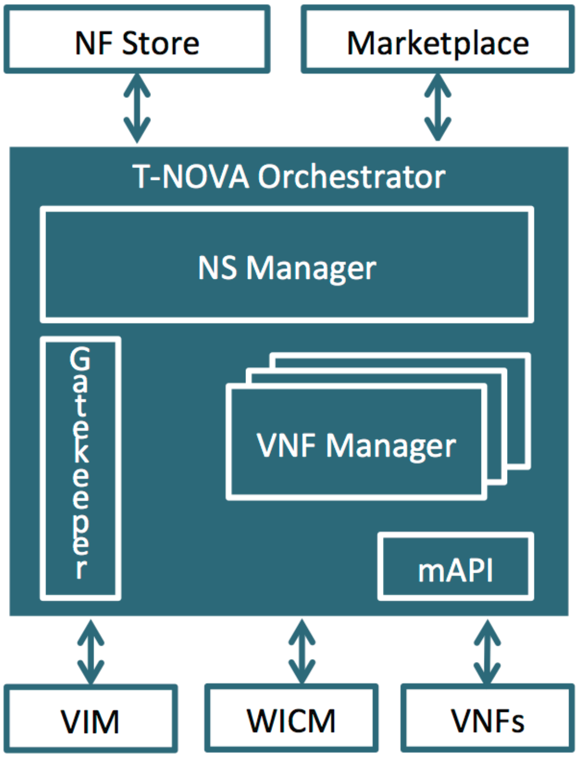


Figure 10 Orchestrator Interfaces

4.2. T3.2. Infrastructure Repository

Task 3.2 was focused on the implementation of a resource discovery and repository subsystem for the T-NOVA Orchestrator.

Within T3.2, the design of the infrastructure repository subsystem was developed as shown in below. This design and implementation comprised of 5 key components.

1. The first component was an enhanced platform awareness agent which runs on the compute nodes and collects platform specific information. This component was implemented as framework of libraries, commands and script to collect and aggregate a rich set of compute node information.
2. The second component was a set of listener services. One listener is dedicated for EPA agent messages and a second one is dedicated to OpenStack related messages.
3. The third component is the EPA controller which coordinates with listener services to process and persist updates to the repository database using data files received from the EPA agents or OpenStack infrastructure landscape change notifications.
4. The fourth component was the infrastructure repository database which is responsible for storing the infrastructure related information and the relationships between the stored information. The database was implemented as a graph database using NEO4j in order to support the encoding of the

relationships between the components of the NFVI. This approach also provided a convenient mapping of the system structures within the NFVI and the node structures of the graph database.

5. The final component was a middleware API layer which provides a common OCCI compliant REST based interface to the Orchestrator components that need to retrieve information from the repository. The middleware layer also featured a database to support the storage of NFVI PoP ingress and egress endpoints and associated parametric data for the links. The middleware implementation also provided support for multiple instances of the PoP level resources repository databases ensure appropriate scalability of the subsystem.

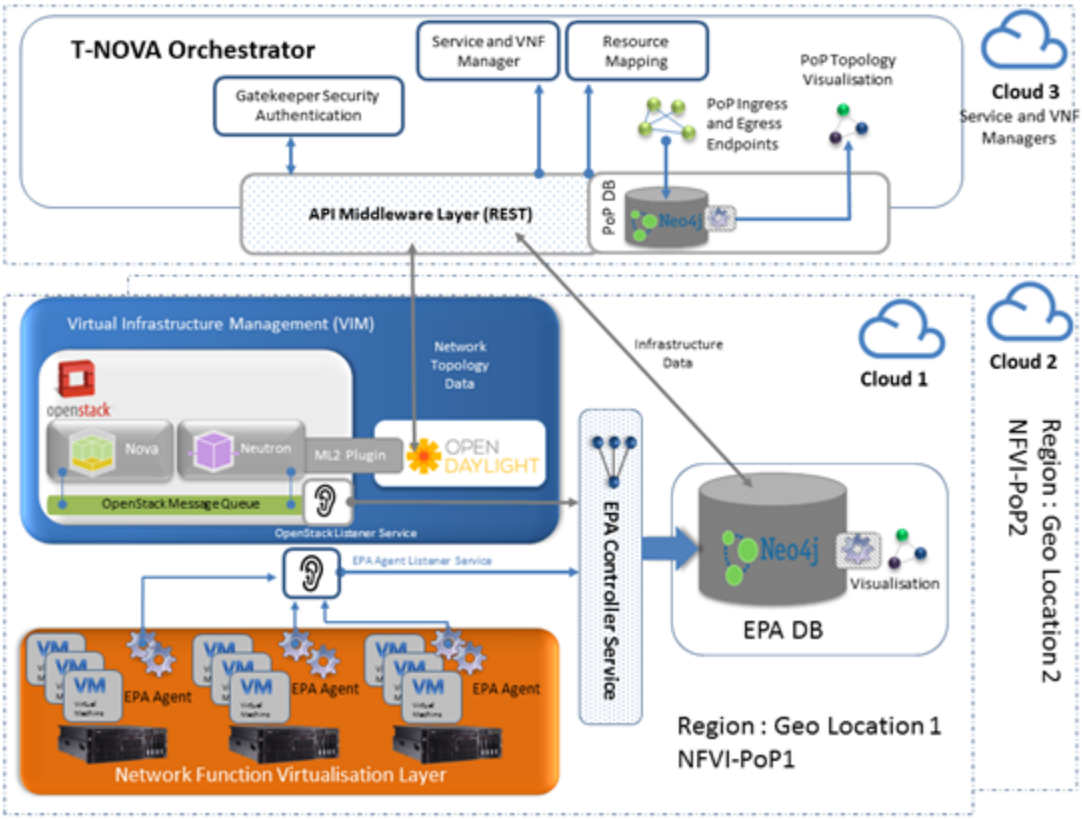


Figure 11 Infrastructure Repository Sub System Architecture

All components have been successfully implemented and integrated to deliver a fully functional infrastructure repository subsystem. The code from the implementation of the repository was open sourced is available from the Intel Labs git hub repository under an Apache II licence. In addition, Task 3.2 also:

- Developed API proxy to authenticate middleware calls using the T-NOVA authenticated component (Gatekeeper) developed by ZHAW.
- Implemented Docker container based packaging of the infrastructure repository components to support partner or third party deployments.
- Created infrastructure repository testbed for partner integration activities with infrastructure repository.
- Deployed the repository on the NCRSD testbed for integration testing.
- Completed documentation of Middleware API's on T-NOVA wiki.

Finally, the task completed deliverable D3.2 which detailed the implementation of the infrastructure resource repository, and interfaces.

4.3. T3.3. Service Mapping

T3.3 was responsible of the development of the algorithm and components for the Service Mapping. The following tasks were carried out towards this goal:

- A revision of the scientific and technological state of the art in service mapping has been carried out, in order to investigate different approaches and technological solutions, pointing out their advantages and limits with reference to T-NOVA needs.
- The service mapping has been contextualized into the T-NOVA platform, with particular attention to the interdependencies with the other modules.
- The service mapping problem has been defined in terms of input, expected output, constraints and evaluation metrics. In particular, a formal definition of the service mapping problem in T-NOVA has been given. A “two levels” mapping approach has been proposed in order to deal with the problem complexity, by defining a “first level problem” (assignment of virtual network functions to data centres) and a “second level problem” (assignment of virtual network function components to the servers inside a data center). The “flat problem” (joint solution of the first and second level problems) has been defined as well, also with the aim of having a possible benchmark for comparison of performances.
- Performed discussion on: (i) assignment feasibility of the service mapping problem (modelling of requirements/constraints on network services to be mapped and resources available), (ii) possible objective functions for the two-level service mapping problems, (iii) possibility of dynamically reconfiguring the service mapping to optimize usage of resources.
- Design of three distinct algorithms for service mapping and one algorithm for network functions scheduling. The mapping algorithms explore both first level service mapping (mapping of VNFs to data centres) and second level service mapping (mapping of VNF components inside a data center). Two of them are based on optimization techniques, the third on stochastic control techniques. The developed algorithms allow to consider multiple mapping objectives, including: i) maximisation of network service requests’ acceptance, 2) minimization of mapping costs (i.e. the costs for employing the different network resources), 3) balancing of network/datacentre load distribution. Aligned input and output interfaces for the algorithms have been specified in detail, in terms of needed input data and of required output data for a service instantiation (an activity performed in view of the design of the actual T-NOVA service mapping module). Simulations results in emulated environments have

been proposed and discussed in order to highlight the features of the algorithms.

- Design of a microservice-based service mapping module, aimed to host the service mapping algorithm, and of its integration inside TeNOR, the T-NOVA orchestrator. The activity was performed in close cooperation with Task 3.2 on Infrastructure Repository, WP 6 on Network Service descriptor. As a result, the service mapping module offers the possibility to integrate and avail of any service mapping algorithm conforming to the input/output specifications, which are aligned to the three mapping algorithms developed.
- Integration of the service mapping module including one of the developed mapping algorithms has been achieved and preliminary tested.

4.4. T3.4. Service Provisioning Management and Monitoring

T3.4 was devoted to the design and implementation of the service provisioning, management, and monitoring components within the orchestrator. T3.4 was focused thus on the design and implementation of the orchestrator kernel itself.

The first step was the design and specification of the internal architecture of the orchestrator, including the different functional blocks identified in order to satisfy and cover the requirements coming from WP2. Therefore, the architecture, which has been designed to follow the micro-services approach, aligned also to the design of the interfaces within task T3.1, is composed of three major groups of functional blocks, each one of them devoted to the following functionalities:

- Service lifecycle and management
- VNF lifecycle and management
- Orchestrator management User-Interface

The implementation of the architecture mainly selected Ruby as the main programming language, although the modularity of the architecture (see Figure below) did not prevent any micro-service to utilize an specific language, as long as the REST APIs were offered. The architecture included the set of catalogues, and repositories, as well as a set of auxiliary services required to build the system.

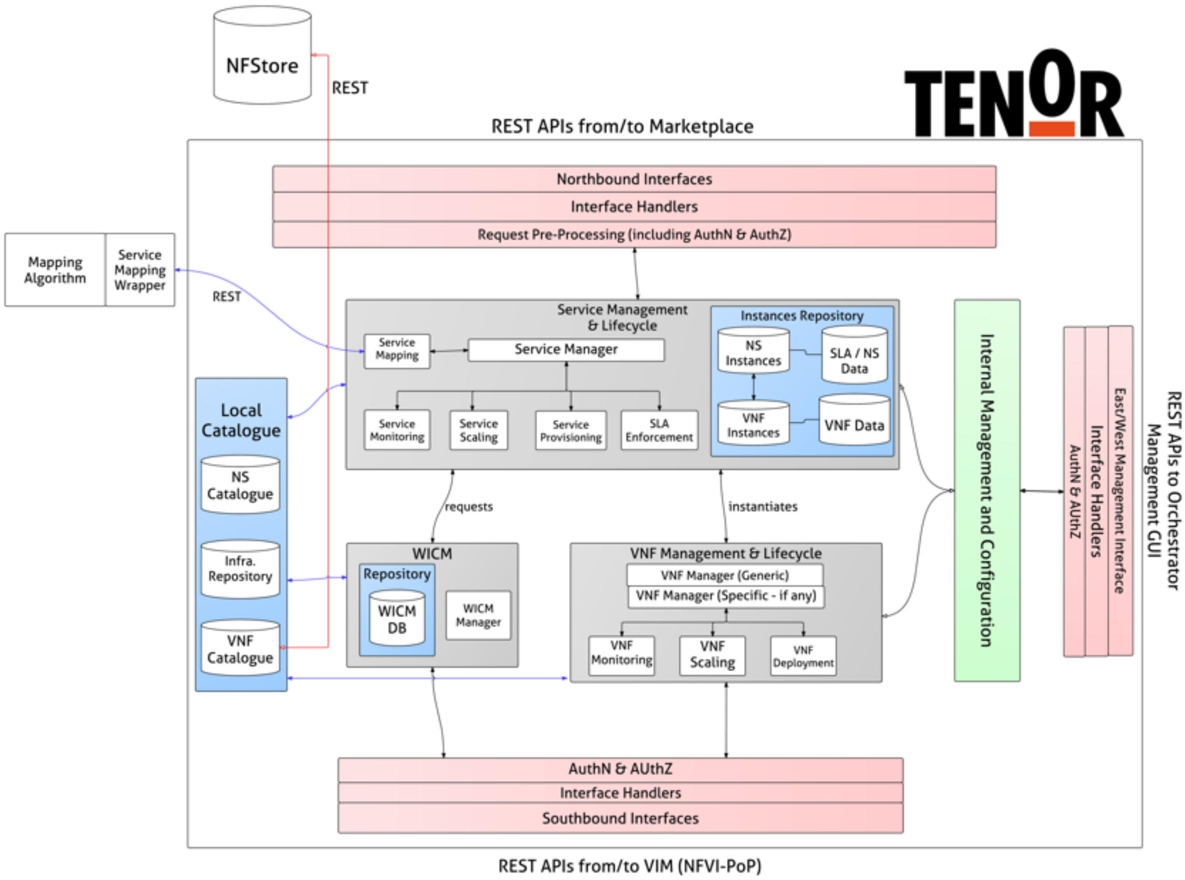


Figure 12 TeNOR Functional Architecture

Figure above (Figure 12) depicts the functional architecture, which has been implemented using microservices.

Furthermore, this task focused on the service provisioning and monitoring workflows design and implementation, which indeed led to the integration of the orchestrator with the rest of the components of the T-NOVA architecture. In fact, considering the intermediate position of the orchestrator in the T-NOVA stack, this task carried out the first integration activities of the project, together with WP4, and WP6. On the one hand, the NS provisioning workflow, which involved the provisioning of an NS from the Service Manager up to the corresponding VNF Manager modules, and which involved also the participation of different micro-services (e.g. service mapping, HOT generator, etc.). On the other hand, the NS monitoring workflow, which included the monitoring of different metrics coming from the VIM, processing and storing them in the orchestrator, and finally forwarding of the metrics to the Marketplace. The implemented monitoring mechanism included a subscription system between the orchestrator and the VIM monitoring system developed in WP4.

The last phase of the task included the implementation of the SLA module in order to detect breaches and the implementation of the scaling and auto-scaling when a monitoring breach is detected.

5. WP4. Infrastructure Virtualisation and Management

5.1. T4.1. Resource Virtualisation

Task 4.1, was focused on the identification, characterisation and optimisation of the hardware and software components used in the implementation of the T-NOVA IVM. Task 4.1 is also examining the inter-relationship of VNFs and their host virtualised environments. Finally the task is developing a set of best practices and insights regarding the appropriate configuration of the infrastructural components and the technologies to be used in the implementation of the T-NOVA IVM. The key activities and outputs of this task are as follows:

Building on the work carried out in Task 2.4 in relation to the requirements and potential technology options for the IVM, candidate technologies for the various functional entities were selected. These selections were primarily based around the Open Network Platform (ONP) Server reference architecture. ONP was selected as the reference platform given its use of standard high volume servers and a software stack based composed of open source and open standard software building blocks. The referenced open software ingredients include:

- OpenStack – Management and orchestration of physical and virtual resources in cloud environment
- OpenDaylight -Open, standards-based SDN controller platform.
- DPDK: Accelerates packet processing on general purpose processor including Intel® Architecture Processors.
- Open vSwitch (OVS): A virtual multilayer network switch.

A resource virtualisation test bed was designed and implemented using ONP. Some issues identified with default configuration which affected the setup of DPDK Open vSwitch making it unsuitable for T-NOVA requirements. Modifications were identified and implemented to resolve these issues.

As next step, the task carried out characterisation experiments on the networking technologies options which can be used to improve packet processing performance. Technologies investigated included OVS, OVS-DPDK, SR-IOV and Snabb Switch. It was demonstrated that a combination of SR-IOV and DPDK achieved in excess of 8Gbps network throughput. The performance of a non-virtualised virtual Traffic Classifier network function was compared with Docker container and KVM VM implementations, demonstrating that virtualisation through a hypervisor can affect the performance of non-optimised network workloads up to 21%. A new approach to scale switch performance based on a dual-datapath approach was investigated. This approach exhibited high forwarding performance and port density in the accelerated datapath, while the primary datapath exploited the large amount of cache and main memory available in commodity servers to store all required flow state.

The effects of core and non-uniform memory access (NUMA) pinning, core isolation and huge pages on Virtualised Network Function (VNF) performance were investigated.

The results obtained showed that the usage of processor pinning can help to achieve improved performance, if properly configured. The effect of NUMA pinning on virtual machine (VM) performance was identified as being significant with an approximately 50% increase in network throughput. Co-location of a virtualised network function (VNF) on the same NUMA node that is attached to the NIC is important for data plane type workloads. The effects of huge page configurations were found to be scenario specific and their performance contribution is higher in the presence of noisy neighbours where they can improve network throughput by 14%. The configuration of the BIOS settings in compute nodes were investigated and appropriate settings identified for a Network Function Virtualisation Infrastructure (NFVI) testbed. The effects of heterogeneous compute resources namely and FPGA system-on-chip (SoC) in an OpenStack environment were also investigated. Using a storage sensitive VNF the effect of different storage configurations were investigated. The results obtained indicate that the use of local disks is preferential to ephemeral volumes. In fact, with ephemeral volumes on local disks, Live Migration becomes bounded to a Storage Live Migration constraint.

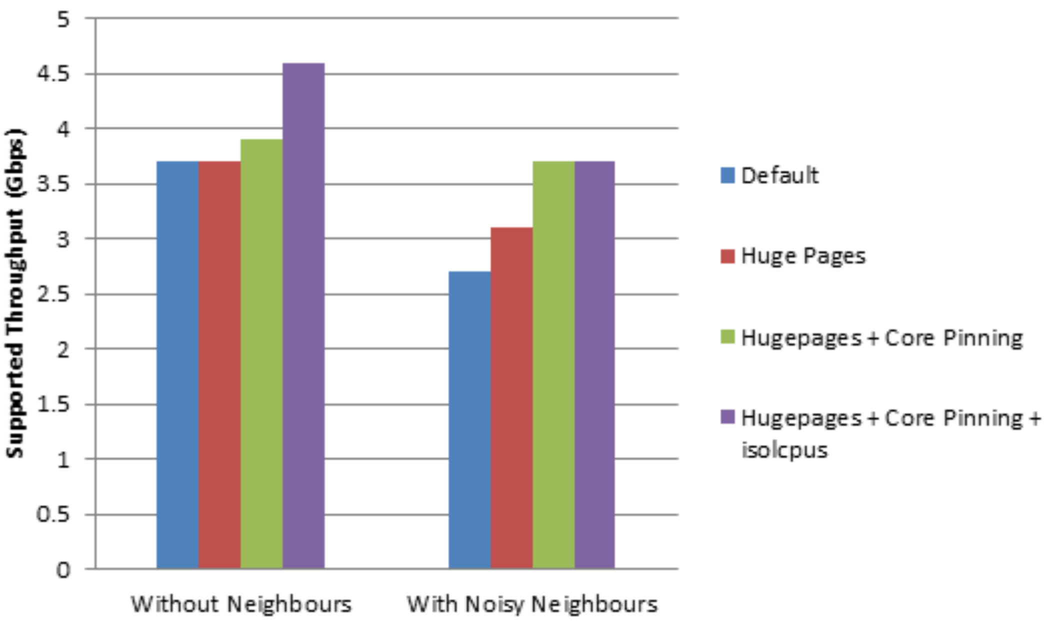


Figure 13 Packet throughput performance of the vTC VNF with different huge pages and core pinning configurations

Significant VNF workload characterisation activities were carried out in the task. A key enabling capability in the form of a VNF workload characterisation framework was developed by the task. The framework is designed to automatically test various configurations of a VNF, in an iterative manner on different target platforms. The framework provides orchestration of the full test case lifecycle. The framework was applied to the characterisation of a virtual Traffic Classifier. Specifically the framework was used to investigate the potential of 'network performance intent' deployments. The effect of the deployment configuration on the performance was demonstrated using a comparison between OVS and SR-IOV network technologies. The data collected was analysed using a machine learning approach in WEKA to generate a decision tree which relates network performance intent i.e. network throughput to the quantity and types of resources allocation during deployment of the virtualised traffic classifier.

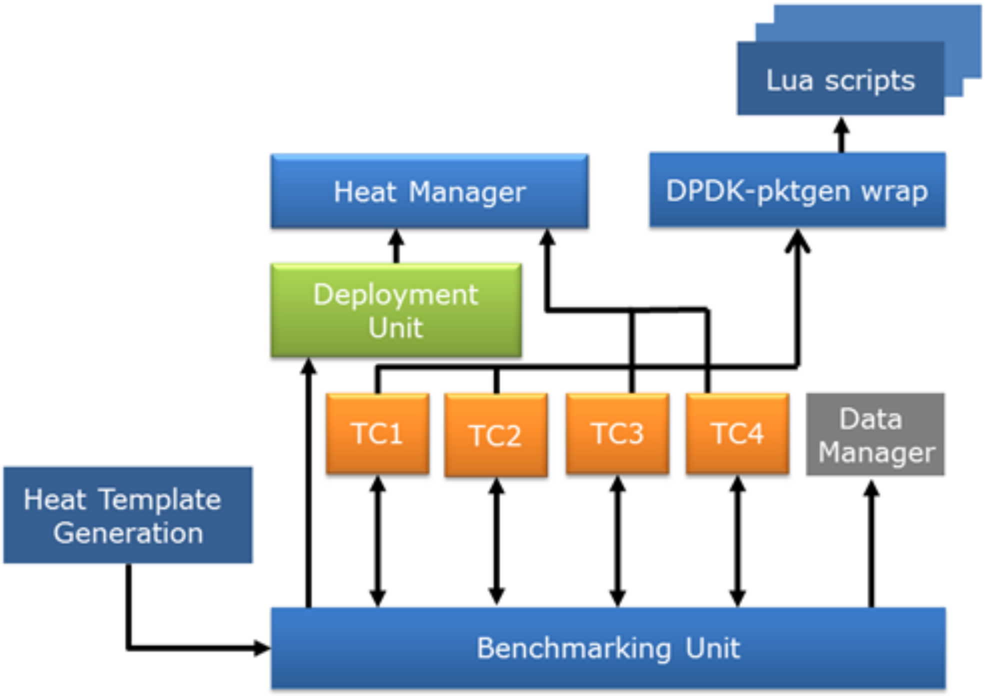


Figure 14 Architectural components of the VNF Characterisation framework.

Also, Task 4.1 developed a collaborative engagement with the OPNFV Yardstick project. A number of key contributions to Yardstick were identified which included the VNF characterisation framework, 4 test cases (VNF lifecycle and VNF data plane benchmarking test cases), virtualised traffic classifier (vTC) and support documentation. Initial contributions were completed by task 4.1 and continued on in task 4.5. The contributions form part of the OPNFV Brahmaputra release.

5.2. T4.2. SDN Control Plane

Task 4.2 was focused on the development of T-NOVA's SDN control framework which supports the management of virtual networks over a datacentre's physical infrastructure and the deployment of VNF services. To this end, relevant issues related to network isolation and traffic steering have been addressed when providing connectivity services, handling either WAN or intra-datacentre connections. In this context, leveraging on state-of-the-art network virtualisation frameworks, several approaches to support service function chaining were investigated.

A major activity within this Task is the distributed deployment of the control plane to address SDN centralisation issues. Research was focused on clustered SDN controllers, with the purpose of providing an elastic control plane able to balance control traffic with the workload. Moreover, experimental analysis on the capability of the SDN controller was conducted so as to ensure persistence of the network configuration in case of controller failures.

Starting from the analysis of the key requirements affecting the SDN control plane procedures, a survey of the current state-of-the-art SDN frameworks was carried out, aiming at selecting the reference baseline for the SDN control framework implementation. Moreover, the architecture of the Control Plane has been designed, identifying the functional components and additional features required to meet the T-NOVA needs.

The technology selection phase carried out by Task 4.2 identified OpenDaylight as the reference SDN control plane. Therefore, several testbeds including OpenDaylight (Helium/Lithium) with OpenStack were setup to analyse key features (clustering, multi-tenancy, SFC, QoS) and to evaluate the performance under a number of different scenarios (centralised/distributed CPs, single-PoP/multi-PoP deployments).

Within Task 4.2, research and development activities related to:

i. Identification of approaches for steering traffic in SDN networks
ii. Development of algorithms for balancing the workload among multiple controllers
iii. Techniques for providing network slicing and isolation with support to QoS
iv. Analysis of the persistency of network configuration
v. Identification of solutions to provide integration of WAN connectivity in T-NOVA.

The tasks activities resulted in the development of the following software components that have been integrated with OpenDaylight:

- **CP Load Balancer**, an ODL module responsible for balancing the control load by spreading the network switches across a cluster of SDN controllers. The first implementation required OpenFlow v1.0 and was integrated in OpenDaylight Hydrogen. Further enhancements to support current releases of OpenDaylight (Lithium and beyond) and OpenFlow (v1.3) are under development.

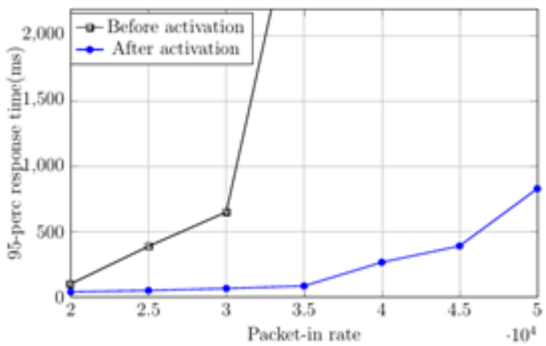


Figure 15 Mean response time of the cluster before and after the activation of the CP Load Balancer

- A Traffic Steering extension to OpenStack allowing the forced redirection of traffic across VMs, so as to enable the service function chaining.

Command line	URI	HTTP Verb	Description
steering-classifier-create	*/classifiers*	POST	Create a traffic steering classifier
steering-classifier-delete	*/classifier/{id}*	DELETE	Delete a given classifier
steering-classifier-list	*/classifiers*	GET	List traffic steering classifiers that belong to a given tenant
steering-classifier-show	*/classifier/{id}*	GET	Show information of a given classifier
steering-classifier-update	*/classifier/{id}*	PUT	Update a given classifier
port-chain-create	*/port_chains*	POST	Create a port chain
port-chain-delete	*/port_chain/{id}*	DELETE	Delete a port chain
port-chain-list	*/port_chains*	GET	List port chains that belong to a given tenant
port-chain-show	*/port_chain/{id}*	GET	Show information of a given port chain
port-chain-update	*/port_chain/{id}*	PUT	Update a port chain

Figure 16 Traffic Steering API

- WAN Infrastructure Connection Manager, a component to handle the integration of WAN connectivity services with the NFVI-PoPs that host the VNFs

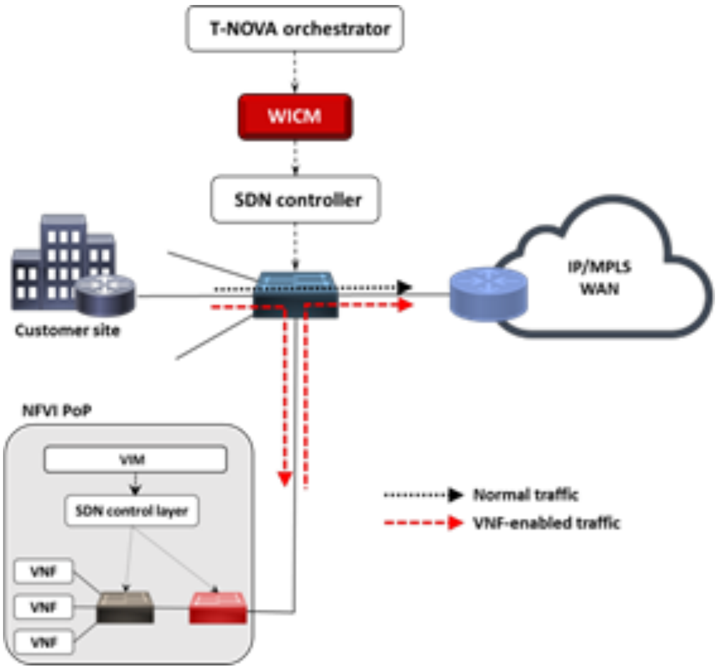


Figure 17 NFVI-PoP and WAN integration with WICM

5.3. T4.3. SDK for SDN

Task 4.3 initially focused on conducting experiments to analyse the protocol inefficiencies in packet paths from sender and receiver in a datacenter using an OpenStack cloud. The results of the experiment showed major header inefficiencies with scope for optimisation using SDN approaches. Moreover as the OpenDaylight community was progressing towards becoming the de-facto leader in the open-source space, this has made the originally proposed idea of the SDK being a multi-controller support layer obsolete. The SDK scope was thus redefined to one with greater chance of scientific and industry impact. The NetIDE project was also analysed to better differentiate the T-Nova SDK output from NetIDE. One on one interviews with use-case leaders were initiated to identify task linkages properly. Task 4.3 moreover formulated test (use) cases for SDK4SDN and finalized the deployment of the ZHAW SDN testbed: OpenDaylight Helium and OpenStack Juno using automation with Packstack. Target applications were identified for implementation using the SDK4SDN. Detailed analysis was performed on ODL focusing on ODL-Neutron interactions, OVSDB, MD-SAL, Karaf. Initial target for the SDK was the development of application capabilities which supported tenant isolation without tunnelling.

After the initial test cases and applications were established for the SDK, focus shifted towards discussions and state of the art analysis of the service function chaining mechanisms as potential implementations to be considered in T-Nova (novel solutions coming from the ODL community itself and working groups like OPNFV). Detailed ODL-OVS-OpenStack infrastructure analysis and complexity characterisation on the ZHAW SDN testbed followed. ODL code test cases related to ovsdb, openflow, md-sal and neutron libraries were performed to provide insights which could be used in the development of the designated SDK applications. The knowledge and experiences acquired from the test cases were also were used as know-how transfer to education via summer school labs, lectures and technical blog posts.

In support of the development of the SDK libraries, an architectural design of the SDK was developed. The baseline of an application in the SDK is the Network Graph data structure that is maintained by the following APIs (as shown in the image below):

- OVS components such as bridges, ports and interfaces through the OVSDB Southbound Plugin
- Host information through the Openstack Neutron API
- Topology discovery through the Openflowplugin

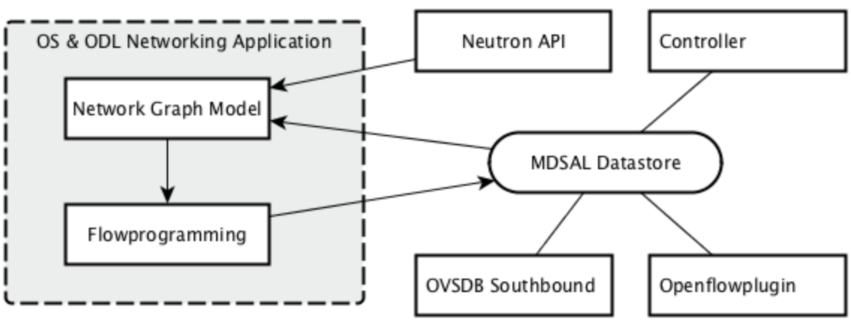


Figure 18 SDK interfaces with OpenStack and ODL components

The following figure shows the network graph of the SDK. The Network Graph models the network components seen by the OVSDB Southbound Plugin. The graph itself holds all the information given by the mentioned APIs and exposes it by providing Network Paths which model host-to-host connections.

The SDK models connection based flow programming by providing Flow Patterns (TBI). A Flow Pattern maps to a Network Path where the OVS components are divided into three logical segments. The source resp. destination bridges and the aggregation bridges.

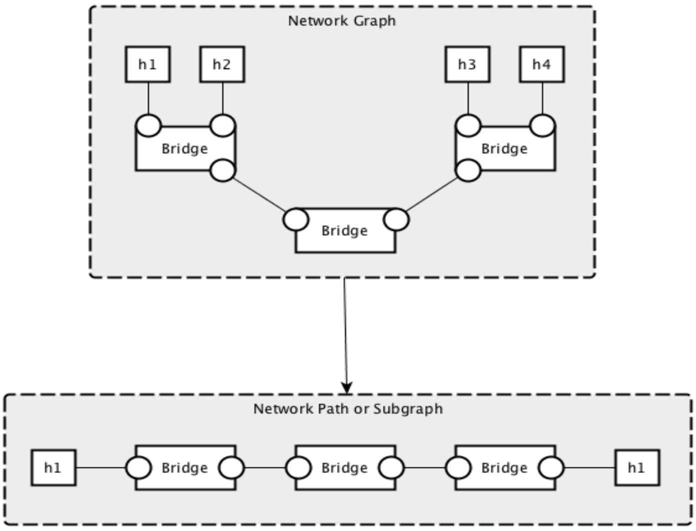


Figure 19 SDK Network Graph

Iterative development of the SDK libraries was carried until the first SDK stable version was released as open source code on github. In the last quarter of the year a major focus was given to the work on completing and reviewing Deliverable 4.31. Meanwhile SDK development progressed: Features to support Neutron in SDK were added including tenant isolation based on Neutron subnets, support for LLDP, resilience features and a service function chaining algorithm. The logic behind the FlowPattern & Flow programming concepts in the SDK was implemented. Before the integration meeting in Athens the REST API specification was completed. Development of Service chain creation f and deletion APIs was fully tested and verified as functional. An initial setup of SFC using the vTC VNF as validated as being functional. It is based on novel header rewriting algorithm and flow programing solution implementation in the SDK.

5.4. T4.4. Monitoring and Maintenance

Task 4.4 designs, implements and integrates a monitoring framework at VIM layer which collects metrics from computing and network nodes (both virtual and physical), aggregates and analyses them. Generic VNF metrics are also collected and processed. As an output of the metrics processing workflow, selected measurements and alarms/events are communicated to the Orchestrator via an API to be also defined in this task, in order to facilitate service mapping and management procedures.

To that end, Task 4.4:

- Identified and consolidated IVM requirements which affect the monitoring framework

- Conducted extensive state-of-the-art survey on monitoring frameworks for virtualised infrastructures, focusing on cloud (and especially Openstack) monitoring as well as Openflow monitoring
- Identified specific challenges and innovations for IVM monitoring in the frame of T-NOVA
- Defined the architecture of T-NOVA IVM monitoring framework
- Defined the role and functionality of the associated components (VIM monitoring manager, VIM monitoring agent) as well as their interactions with the rest IVM components

During the implementation phase, the components of the VIM monitoring subsystem (Figure 20), were developed, integrated into the T-NOVA IVM testbed and tested.

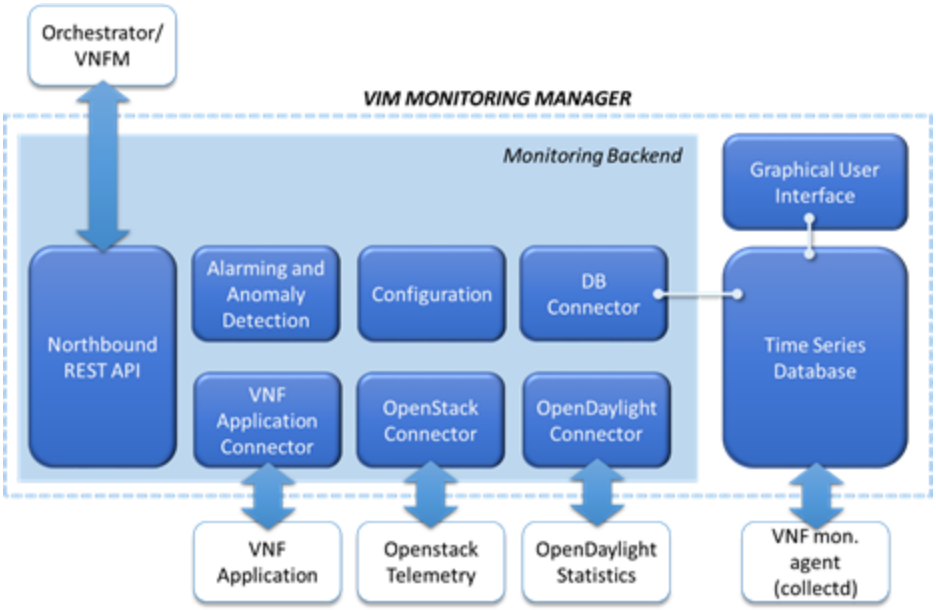


Figure 20 Architecture of the VIM monitoring subsystem

The implementation included the following components:

- OpenStack and OpenDaylight connectors, used to periodically poll the two platforms via their monitoring APIs.
- A VNF Application connector, which accepts data periodically dispatched by the VNF application. These metrics are specific to each VNF.
- A time-series database (InfluxDB) for data persistence
- A Northbound API, which communicates selected metrics and events to the Orchestrator and, in turn, to the VNF Manager(s). A REST API allows metrics to be communicated in either pull (via GET request) or push mode (via publish-subscribe methods). Initial integration with the Orchestrator was achieved using this API. (see Figure 21Figure 20)
- A Graphical User Interface (GUI) based on Grafana, which visualises the stored metrics and presents them as live, time-series graphs. (see Figure 22)

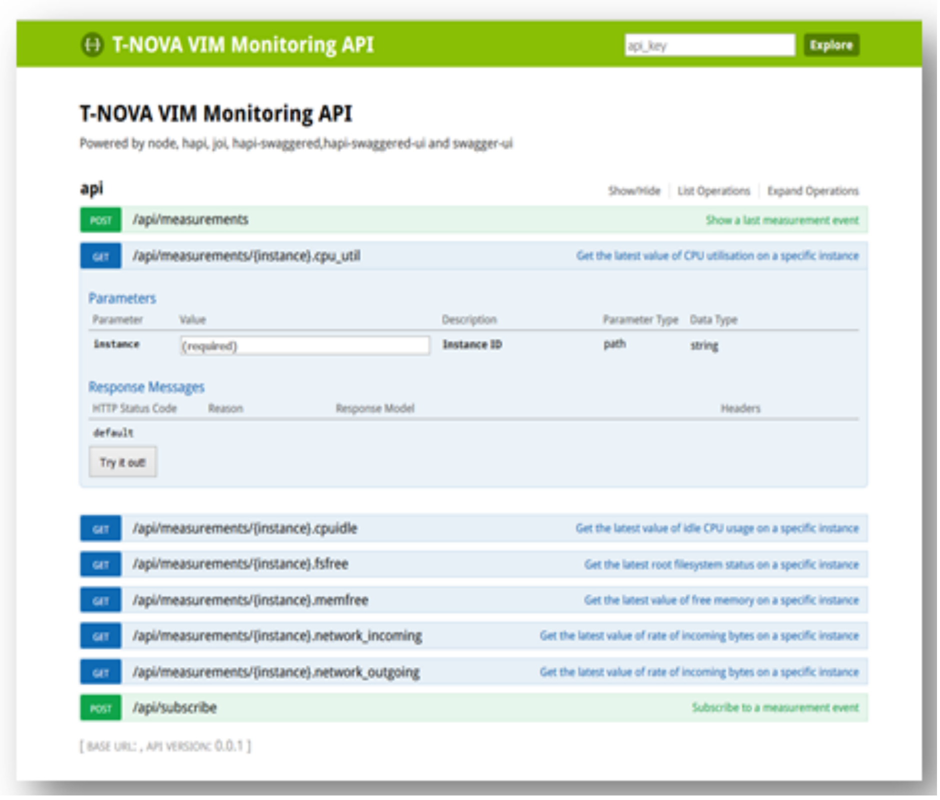


Figure 21 Live documentation (via the swagger framework) of the northbound API of the VIM MM

Significant effort was also devoted to the integration of the VIM MM with the VNFs under development in T-NOVA. To facilitate this integration (and also to assist developers of any future VNFs), a lightweight toolkit was developed which allows VNFs to easily expose VNF-specific metrics. In parallel, a monitoring agent (based on collectd) was integrated in each VNF image. It was verified that all metrics (generic and VNF-specific) are properly collected by the VIM MM, visualised and subsequently exposed to the Orchestrator.

Further technical work focused on developing the anomaly detection mechanism for automatically detecting faults in VNFs without pre-defined metric thresholds. Two separate approaches (linear regression and Mahalanobis distance) were developed and evaluated.

Finally, the VIM monitoring system was organised into a set of Docker containers, to facilitate easy of deployment and integration.

The current version of the VIM MM has been released as open-source in the Github area of the project, accompanied with supporting documentation (https://github.com/T-NOVA/vim-monitoring).

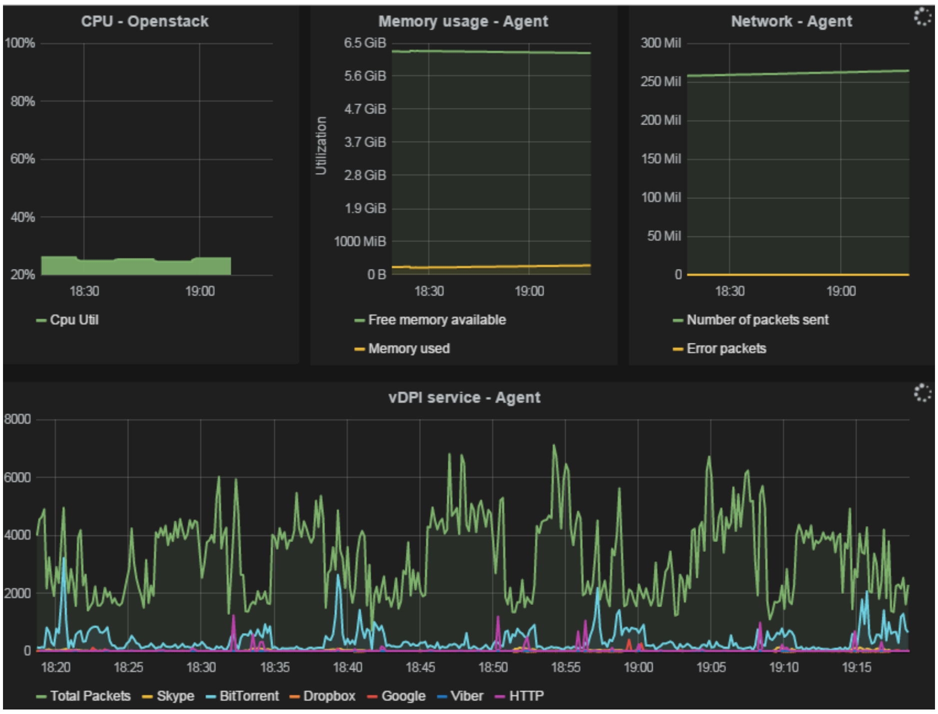


Figure 22 VIM monitoring GUI snapshot

5.5. T4.5. Infrastructure Integration and Deployment

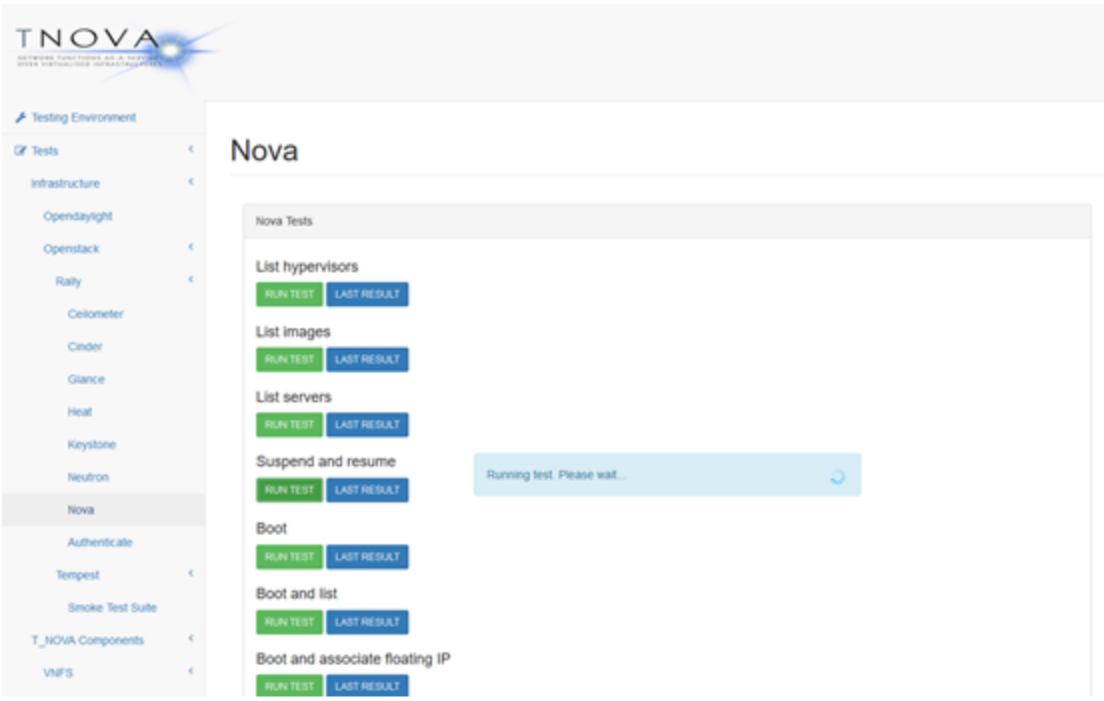
The primary focus of Task 4.5 was the implementation of a testbed comprised the implementations of the functional entities of the T-NOVA IVM layer. The task provided the necessary hands on knowledge and best known methods (BKMs) for the deployment and configuration of the selected technology components to achieve a stable and reproducible deployment for the testbed. A set of functional test cases that helped in the validation of both the components and overall testbed was defined and executed by this task.

In specific, Task 4.5:

- Specified the hardware and software components that comprise the integrated IVM layer infrastructure, based on the output from the other WP4 tasks. Detailed descriptions of the partners' testbeds and deployment information were documented.
- Investigated and worked towards the integration of the IVM layer components (OpenDaylight, OpenStack, VIM monitoring framework, SDK for SDN, FPGA SoC and developed VNFs) into the T-NOVA testbed. Moreover, produced detailed guidelines and instructions on the steps involved in the deployment and integration of the T-NOVA IVM layer components and their required configuration to produce a functional and performant testbed.
- Identified and reviewed available IT automation and infrastructure configuration management platforms (Foreman, Puppet, Ansible).

Experimentation with their open source versions was conducted to determine which tools that can facilitate the deployment of the IVM layer components.

- Researched the existing frameworks for validating, performance testing and benchmarking OpenStack at scale. Specifically, Tempest and Rally OpenStack projects for the verification of OpenStack deployments were explored. Moreover, installed and experimented with Robot Framework for OpenDaylight testing.
- Defined a set of test cases that validate the proper deployment and function of the IVM layer stack components. Specifically, test cases for OpenStack and OpenDaylight projects were defined for verifying proper installation and functionality. In addition, test cases that validate the OpenStack - OpenDaylight integration were defined and developed.
- Developed a testing dashboard application (see Figure 23) to support testing procedures, facilitating automated installation of the testing environment, execution of the tests and graphical presentation of the results.
- Completed development of VNF Characterisation Framework and implemented 4 test cases. This work that was contributed to the OPNFV Yardstick project.



- **Figure 23 Screenshot of the test page in the testing dashboard**

6. WP5. Network Functions

6.1. T5.1. Function Packaging and Repository

In the Task 5.1 the effort focused in finely defining the implementation of the network function store. The NF Store is mainly a repository for the VNFs and their metadata. It contains the virtual machines (VMs) software images and the metadata descriptor composing each VNF.

Precisely Task 5.1 contributions involved the following topics:

- Establishing the state of the art: Image repositories and OpenStack repositories from open source projects as well as software building blocks were studied.
- Formalizing general, functional and non-functional system requirements describing the high-level architecture, specifically broken down into 3 components: NF repository, NF Store manager and NF Store interfaces.
- Specification of sequence diagrams has been performed, that describe the interactions over the NF Store interfaces.
- Development and integration of the NF Store components.

The NF Store provides REST interfaces to the orchestrator (T-OR-NFS) and marketplace (T-DA-NFS and T-BR-NFS); an additional shell interface is provided by NFS manager and is used to manage the NF Store service as standard linux service. The Network Function Store function is developed like a web application running into an application server; the chosen server is Apache TomEE, an all-Apache Java EE 6 Web Profile certified stack built above the Apache Tomcat Servlet Container. The high-level architecture of the NF Store is illustrated in Figure 24.

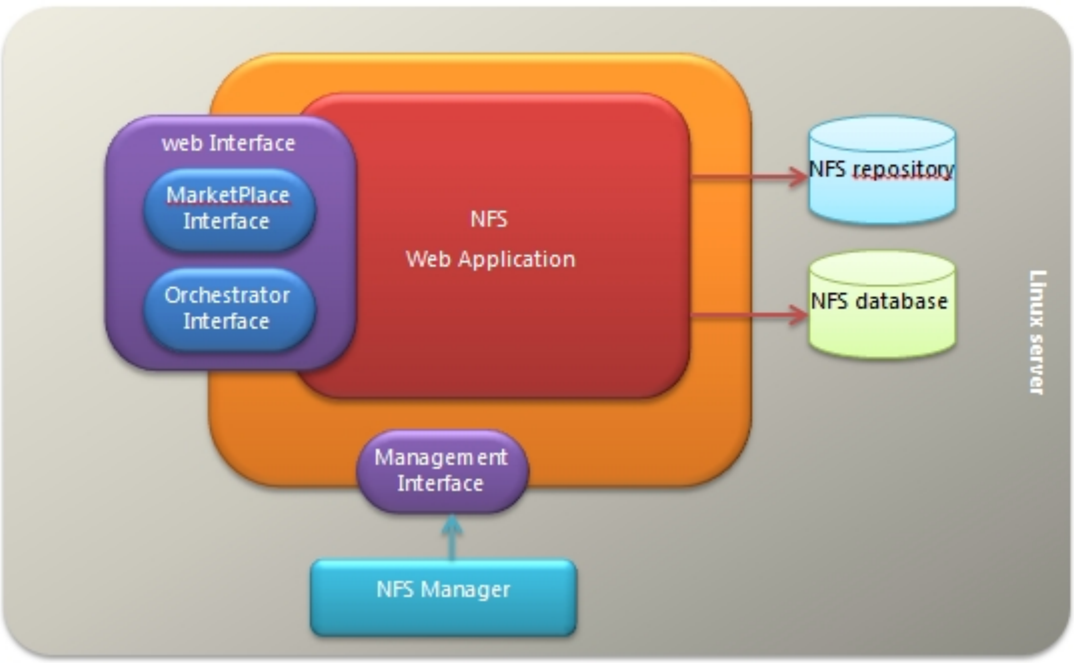


Figure 24. NF Store High Level Architecture

The NF Store provides REST interfaces to the orchestrator (T-OR-NFS) and marketplace (T-DA-NFS and T-BR-NFS); an additional shell interface is provided by NFS manager and is used to manage the NF Store service as standard linux service. The Network Function Store function is developed like a web application running into an application server; the chosen server is Apache TomEE, an all-Apache Java EE 6 Web Profile certified stack built above the Apache Tomcat Servlet Container.

The NF Store application implements the interfaces to repository, database and web interface; it is also responsible for concurrent operations that are performed (CRUD operations). In addition, it delegates some traits of its responsibility (like authentication and authorization access) to an external AA module. Security mechanisms can be adopted for data exchange over these interfaces.
The NFS database is used to archive information about VNF (metadata files); VNF image files are saved into a file system used as NFS repository.

6.2. T5.2. Function Deployment, Configuration and Management

Task T5.2 is focused on mechanisms to accommodate VNF deployment, configuration and management. These mechanisms are realized in a middleware framework which abstracts from the T-NOVA Orchestrator the enforcement of configurations on VNFs. Initially the task defined general requirements for a component such as the middleware framework. Afterwards, the task proceeded to the identification of necessary functional and non-functional elements. For the identification of these elements it was taken into account not only this component's role but also the output of the various tasks within WP2.

The task designed an architecture for the middleware framework with special focus on the interactions between the various elements. Moreover, a definition of the northbound and southbound interfaces highlighting the functionality of this component was proposed.

Follwing, an analysis was carried out on open-source software to be used in the implementation of the middleware API (mAPI). This study focused mostly on configuration platforms, traditionally used in DevOps environments, more specifically in Rundeck and Saltstack. During this period we also studied the ETSI contributions regarding the configuration and management of VNFs which provided significant help on the definition for the generic workflow in lifecycle event. This generic workflow is composed of two actions: an optional upload of a configuration file composed of instantiation specific parameters (e.g. IP address) and running a remote command on the virtual resources that support the VNFs.

This task worked closely with WP3 (responsible for the implementation of the T-NOVA Orchestrator) and VNF developers, also part of WP5, to specify the integration at the north and at the south of the mAPI respectively. The first was relevant for the specification of the mAPI northbound interfaces, namely the VNF Register Interface and the VNF Lifecycle Management Interface. The second was used to produce the specification of the southbound interfaces towards the VNFs, namely the SSH and HTTP drivers. Moreover, because the mAPI plays a pivotal role in the VNFs lifecycle management and interconnects the T-NOVA Orchestrator and the VNFs, this task provided a strong contribution to the VNF Descriptor (VNFD) specification, more specifically regarding the lifecycle operations.

The implementation of the mAPI included;

- Implementation of the northbound interfaces for VNF registering and lifecycle management
- Implementation of the mAPI database for persistent storage of the VNF data
- Integration of the Rundeck platform in mAPI
- Study and analysis of Rundeck plugin development
- Implementation of the HTTP plugin in Rundeck
- Implementation of two authentication mechanisms in the northbound interfaces: the standard T-NOVA authentication system, Gatekeeper, and basic authentication through username and password
- Validation of the alpha version of the mAPI with the lifecycle management of the vProxy VNF

6.3. T5.3. Development of Network Functions

This task effort is devoted to the development of the proposed T-NOVA VNFs. A summary of the activities is provided below:

- State-of-the-art surveys for each VNF w.r.t the algorithms supporting VNF operation and industry and open source solutions available.
- Elaboration and research over implementation methods in order to satisfy performance requirements and functionalities. Technologies that enhance the operation in virtualised environment where considered.
- Elaboration on the design of each VNF according to the selection of technologies and the decisions made for each one.
- Implementation of VNFs for testing and evaluation of the design and technical choices.
    - **Security Appliance (SA)**: device designed to protect computer networks from unwanted traffic. The vSA is able to sense potential dangerous or suspicious traffic and to respond appropriately to either block it (for example, in case of DDoS attacks) or redirect it to a traffic analysis/forensics virtual device for deeper attack pattern analysis and recognition.
    - **Session Border Controller (SBC)**: device used in multi-media telecommunication for providing network interconnection and security services between two IP networks whenever multi-media sessions involve two different IP network domains. The design of this VNF

adopts acceleration technologies (DPDK, GPU) for real-time high intensive computation.

- **Traffic Classification VNF**: DPDK supported version and Libpcap based version where implemented and tested. DPDK version stopped due to a halt in INTEL DPDK framework development. Future DPDK implementation is still considered. The implementation of the DPI engine is based on Opensource DPI library (i.e. nDPI).

- **Virtual Home Gateway (vHG):** An implementation on a HG which delegated its Caching, Transcoding and Streaming operations to a vNF was studied. This aimed at enhancing the delivery of media content to the box.

Task 5.3 has also analyzed the use of HW acceleration in the development of VNF. In particular, the use of various technologies has been tested, and has brought to the development of VNFs based on SRIOV for networking acceleration functionalities, and Graphical Processing Units for accelerated video processing.

6.4. T5.4. Performance Assessment and Optimization

The activities carried out by task 5.4 are strictly related to those of task 5.3. In particular, guidelines for the test phase of the VNFs were discussed, in strict collaboration with task 5.3.

The impact on the VNF performance has been investigated. A number of tests to verify the impact of HW acceleration has been performed: tests carried with virtualized network function, with and without HW acceleration, compared to non-virtualized network functions, have been carried.

7. WP6. T-NOVA MARKET PLACE

7.1. T6.1. Service Description

Task T6.1 is focused in the implementation of service description scheme at marketplace level. This includes the following activities:

- State of the art analysis about Service Description languages.
- Technical discussions to agree on the common information model for the whole T-NOVA system.
- Definition of the T-NOVA information model: data that need to be exchange/store within the marketplace.
- Consolidation on the decision to take about a service description language: USDL.
- Decision on implementing the Business Service Catalog by means of SQL Database according to other referenced research works.
- Business service Catalog REST API definition.
- VNF metadata description.
- Network Service Descriptor

A unique Network Service Descriptor (NSD) has been created in T-NOVA to be shared as part of the same information model between the Marketplace and the Orchestrator. T-NOVA Marketplace information model comes as a result of applying TMForum SID model to ETSI NFV information model Figure 25, adding on top of the Network Service Descriptor (NSD) proposed by ETSI for orchestration, a related customer facing Network Service.

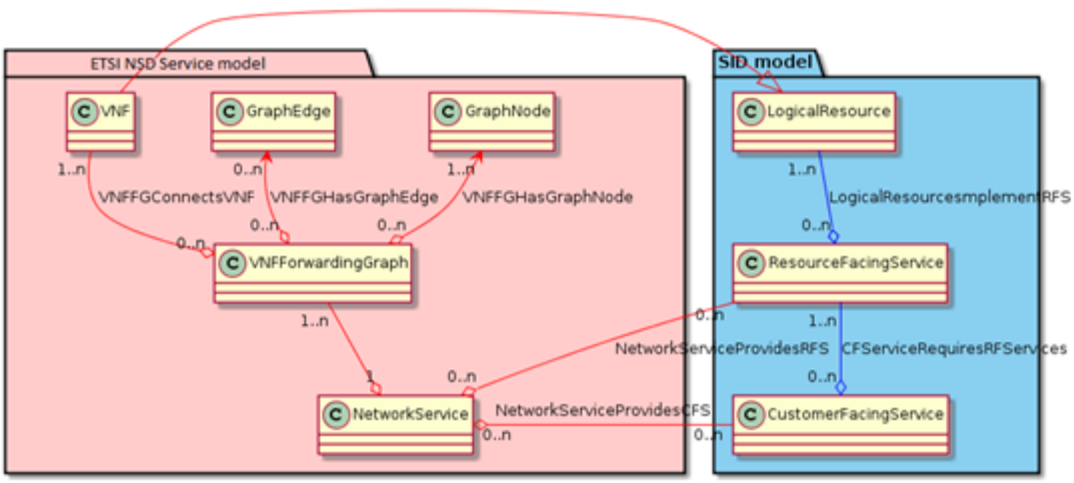


Figure 25 ETSI NSD to SID (TMForum) model mapping

Therefore, T-NOVA descriptors have been created as an extension of ETSI NSD and VNFD, adding the fields that allow business interaction among the stakeholders that interact in the T-NOVA Marketplace: SLA specification, pricing, etc (see Figure 26.

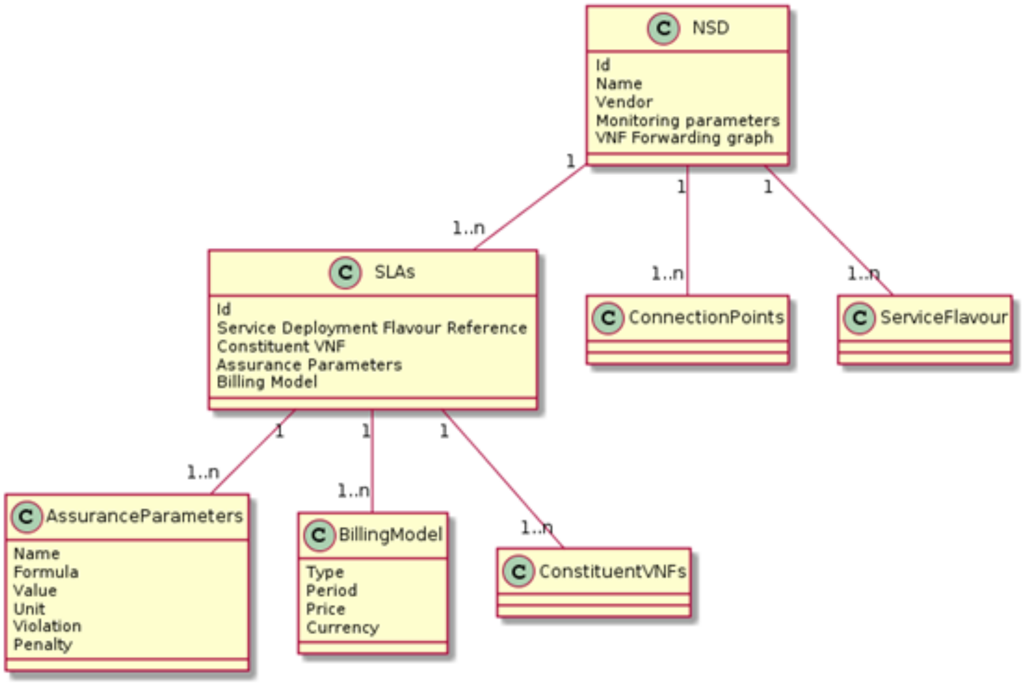


Figure 26T-NOVA NSD SLA Extension

Two main components form the T-NOVA Marketplace service description framework:

- Business Service Catalogue (BSC) (Figure 27): this has been introduced following the TMForum recommendations for business agility. The Service Provider will create service offerings for T-NOVA to generate the NSD that will be stored in the BSC, and which will be later browsable by the customer to select.

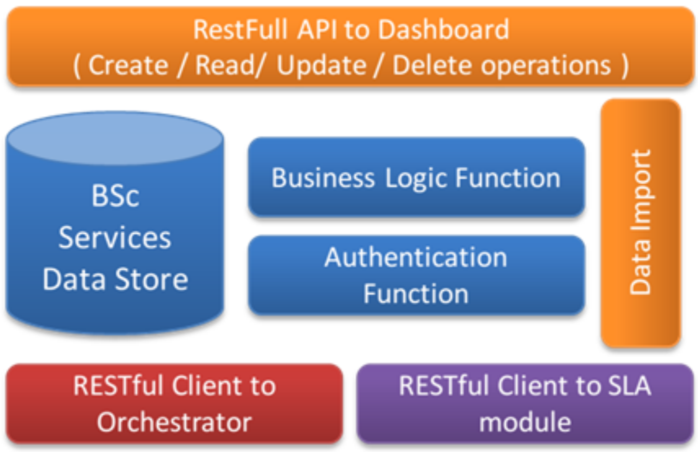


Figure 27 business service catalogue internal architecture

- Service selection module (SS) (see Figure 28): this module has been designed to ease the adaptation of the generic service to the customer network and to send this customization to the Orchestrator to proceed with the service components instantiation.

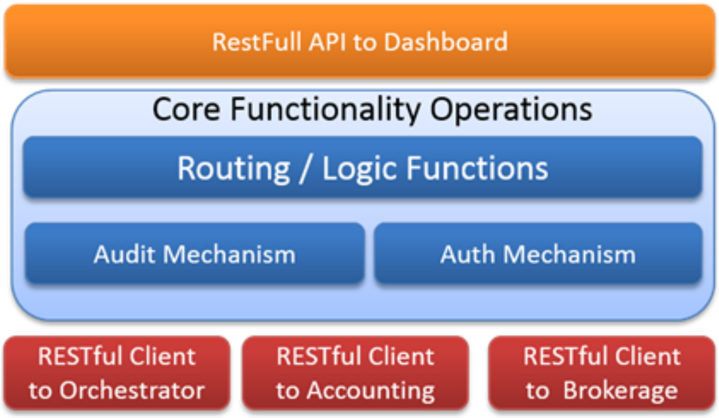


Figure 28 service selection module internal architecture

7.2. T6.2 Brokerage Module

Task 6.2 is focused in the implementation of Brokerage Module. In specific, the following activities have been done within T6.2:

- Consolidation of the definition on Brokerage module.
- State of the art on the available Brokerage platforms.
- Definition of the two main Auction/Trade functions that were used in T-NOVA.
- Implementation of the Trading mechanism between the Service Provider and the Function provider.
- Definition of the interfaces used in order to call the function of the Brokerage Module. REST API operation definition.
- Definition of protocols for the brokerage to retrieve information from the service catalogue.
- integration with the rest of T-NOVA components.
- Definition of the Trading Gui
- functional verification tests.

The T-NOVA Marketplace has been designed as a distributed platform placed on highest layer in the overall architecture which, besides including the users front-end, it comprises BSS components as billing and accounting, and innovative modules as the T-NOVA Brokerage.

Figure 29 depicts the high-level architecture of the T-NOVA Marketplace.

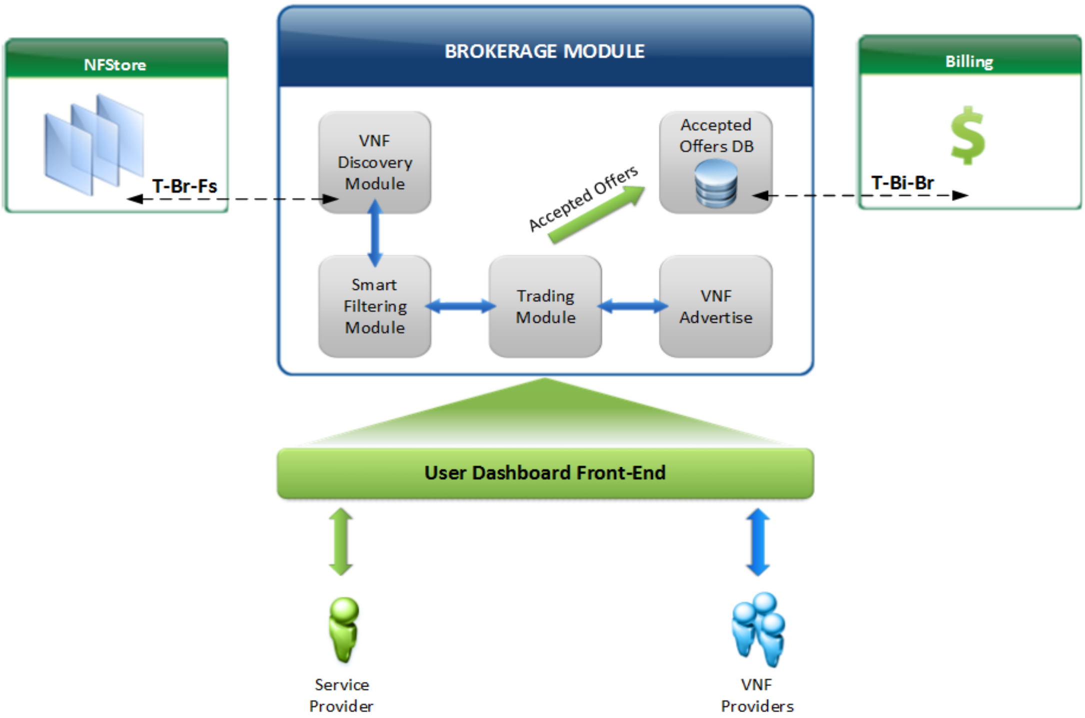


Figure 29 Brokerage module internal architecture

It consists of five main modules that are used in various interactions, as shown in the Figure above:

- **VNF Discovery Module**
    - This module is retrieving all the available and tradable VNFs from the NFStore.
- **Smart Filtering Module**
    - This module applies a smart filtering and listing to the list of the available VNFs based on the users preferences and the SLA parameters.
- **Trading Module**
    - This module is providing all the interaction between the service provider and the function providers. The Trading module is used for requesting a new trade/offer from the SP.
- **VNF Advertise Module**
    - This module is responsible to advertise/return all tradable VNFs to the SP's.
- **Accepted Offers DB**
    - This module is responsible to store the accepted offers in order to be available for the accounting module when this required for billing purposes.

According to the proposed mechanism, the T-NOVA SP browse the offerings from the Service catalogue that match his requirements. If the requested function supports Brokering/Trading the internal modules will try to fulfil the criteria set by the SP. Furthermore, the brokerage initiates the appropriate bid/trading policies according to the T-NOVA SP request inside the trading mechanism in collaboration with the NF Discovery.

7.3. T6.3. User Dashboard

Task 6.3 aims to implement the Users Dashboard.

The dashboard constitutes T-NOVA's system front-end. It enables the Service Providers (SPs) to create and publish their services, the Function Providers (FPs) to create and publish their Virtual Network Functions (VNFs) and both of them to participate in auctions for buying and selling, respectively, the published VNFs. In addition, the Dashboard allows the customers to discover (exploiting various criteria) and consume the offered services. Having in mind terminal neutrality and seamless upgradeability, whenever a new version is available, a web-based implementation of the Dashboard has been selected. Finally and towards maximising the users' Quality of Experience, T-NOVA's Dashboard allows personalization for a variety of settings such as interface, appearance and content according to each user's profile configuration.

Dashboard communicate with the majority of T-NOVA management modules, namely the Billing, UMAA (User Management, Authentication and Access Control), SLA management, Brokerage, Business Service Catalogue, Function Store and Orchestrator.

The following activities have been done within T6.3:

- Consolidation of the definition of the User Dashboard.
- Definition of the design decisions.
- Technology selection for the implementation of the Dashboard.
- Service Provider vs T-NOVA Service Provider discussion
- List of submodules that need to be implemented in the dashboard.
- First version of the Dashboard based on D2.1 Mock-up.
- Refinements of the design of Mock-up
- Second version of the Dashboard based on comments gathered by partners.

For the Dashboard's deployment, the containerised micro-service approach was selected. This enabled each developer assigned with the task of building a T-NOVA module to use its preferable development framework without having to take into account the selections of the other modules' developers (this may increase the development speed).

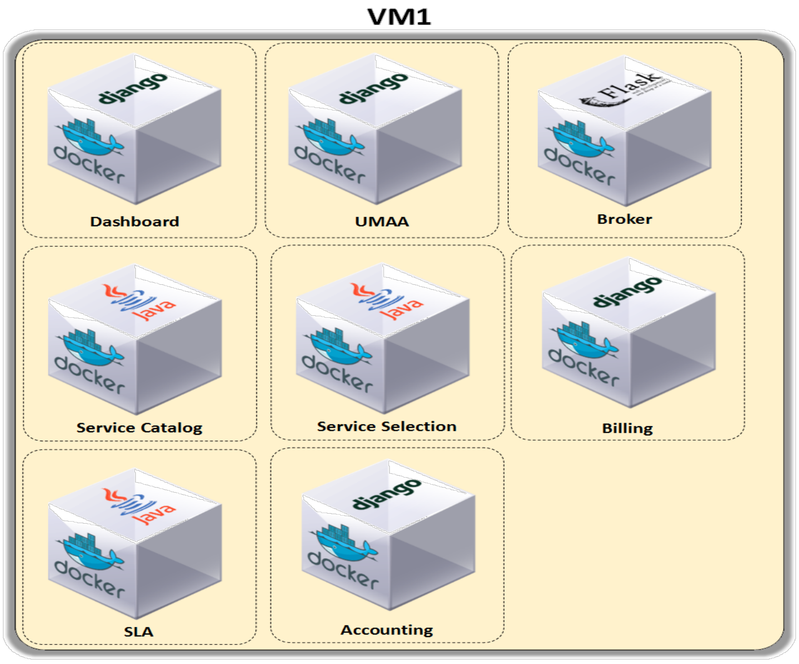


Figure 30 Dashboard deployed as a containerised micro-service

7.4. T6.4. SLAs and billing

Task T6.4 is focused in the implementation of all the necessary marketplace components to provide the SLA and billing functionalities, based on the previous research work. The following activities have been done within T6.4:

- First steps for SLA management module implementation: Open source options and WS protocol review.
- Technical discussions to incorporate SLA description in the VNF metadata.
- Study the SLA information for NFV provided by other standardization bodies, mainly ETSI: monitoring parameters for network services and VNFs.
- Research and refinement of the SLA framework: two levels: SLA between FP and SP and between SP and the customer which will be an aggregation of the previous ones. The technical monitoring parameters that T-NOVA able to monitor are the ones that could be included in the SLA with specific ranges. Therefore an important dependency has been found with the T-NOVA monitoring framework (T3.4 and T4.4) and the VNF developers which will be in charge of specifying a SLA.
- Specific meeting with the VNF developers to define the monitoring metrics than can be part of the SLA as well as typical values that should be considered.
- Definition of the internal architecture of the SLA management module: SLA composer, SLA evaluator, etc.

- Refinement of the information that need to be store for SLA management: SLA specification template, SLA agreement, billable items, and first steps for their definition considering T-NOVA particularities.
- Consolidation of the technology selection for the SLA implementation: WS agreement.
- Definition of SLA template (XML).
- Review of open source billing applications.
- Further Technical discussions about the most suitable billing modalities, mainly for the Network Function Developers/Providers.
- Research of the most suitable billing models to be used in T-NOVA considering the two different commercial relationships. The billing between FP and SP could be license based, but also we could considering revenue model between SP and FP based on service usage.
- Technology selection for the implementation of billing+accounting modules: box billing free-licensed application + accounting database.
- Study of the customization of boxbilling for T-NOVA requirements not considered in boxbilling, for instance managing of several providers.
- REST API definition of the SLA management module and accounting module.
- development of the SLA, accounting and billing modules.
- integration with the rest of T-NOVA components.
- functional verification tests.

The T-NOVA SLA and billing frameworks correspond to the two commercial interactions defined in T-NOVA Marketplace:

- The Service Provider (SP) acquires Virtual Network Functions (VNFs) from the Function Providers (FPs); SLA between FPs and SP.
- The Customer acquires Network Services (NSs) provided by Service Providers based on the combination of VNFs previously purchased. SLA between SP and the Customer.

ETSI NFV requirements for SLA have been considered as input for T-NOVA SLA framework, though not a proper complete SLA business framework has been specified by ETSI so far. TMForum gives insights about metrics and SLA relations in cloud environment that has also been taken into account.

Furthermore, T-NOVA SLA framework has been developed being compliance with WS-agreement specification, as it has been identified as the most complete and extended specification for SLA procedure. All the surveyed research projects in cloud environment have followed this WS-Agreement though there is no research project in the state of the art providing SLA framework for NFV ecosystem as T-NOVA does. SLA architecture is depicted in Figure 31

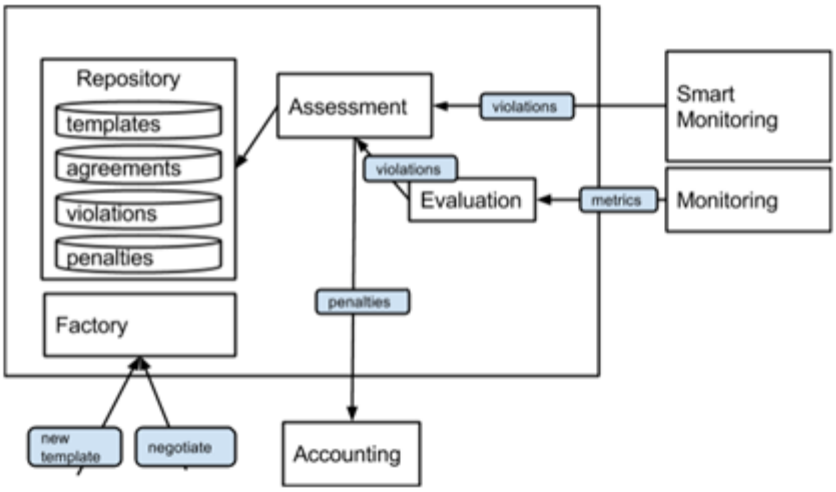


Figure 31 SLA management module internal architecture

After an exploratory work considering different options for billing mechanisms it has been concluded that Pay-As-You-Go is the most generic and suitable model to bill VNFs and Network Services in T-NOVA including an innovative Revenue Sharing model between Service Provider and Function Providers. FPs will benefit from the pay-as-you-earn model, an extension of pay-as-you-go in which the VNF provider will pay a percentage of the revenue received.

The T-NOVA billing framework is composed by 2 modules:

- Accounting module: it keeps a record of all the movements in the system that may have a potential impact in the billing.
- Billing module: it emits the bills based on the accounting information. The billing module being used in T-NOVA extends the generic rating-charging-billing (RCB) framework Cyclops, and whose functionalities have been extended to support the T-NOVA requirements.

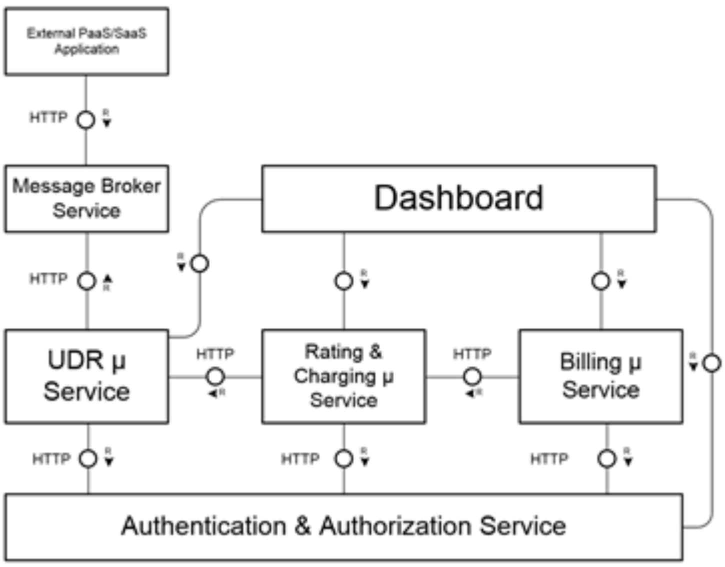


Figure 32 billing micro-service architecture

SLA module and billing module have been developed in Java, and accounting module in Python, and as the rest of components in the T-NOVA Marketplace the three of them have been developed as microservices, exposing RESTful APIs consumed by other T-NOVA Marketplace microservices.

8. WP7. Pilot Integration and Field Trials

8.1. T7.1. Pilot Site Integration and Deployment

The following activities took place within the frame of T7.1:

- Deployment of the required infrastructure elements at Athens Pilot
- Deployment and configuration of the T-NOVA component from all the T-NOVA architectural layers
    - Marketplace components
    - Orchestration components
    - IVM components
    - NFV Infrastructure components
- Deployment and configuration of the VPN remote access to T-NOVA Pilot.
- Initial testing and validation of proper operation
- Elaboration on deployment scenario (w/o ODL integration)
- Multiple NFVI-PoP environment deployment (NFVI-PoP 1 and NFVI-PoP 2)

The outcome of the aforementioned activities is illustrated in following figure (Figure 33).

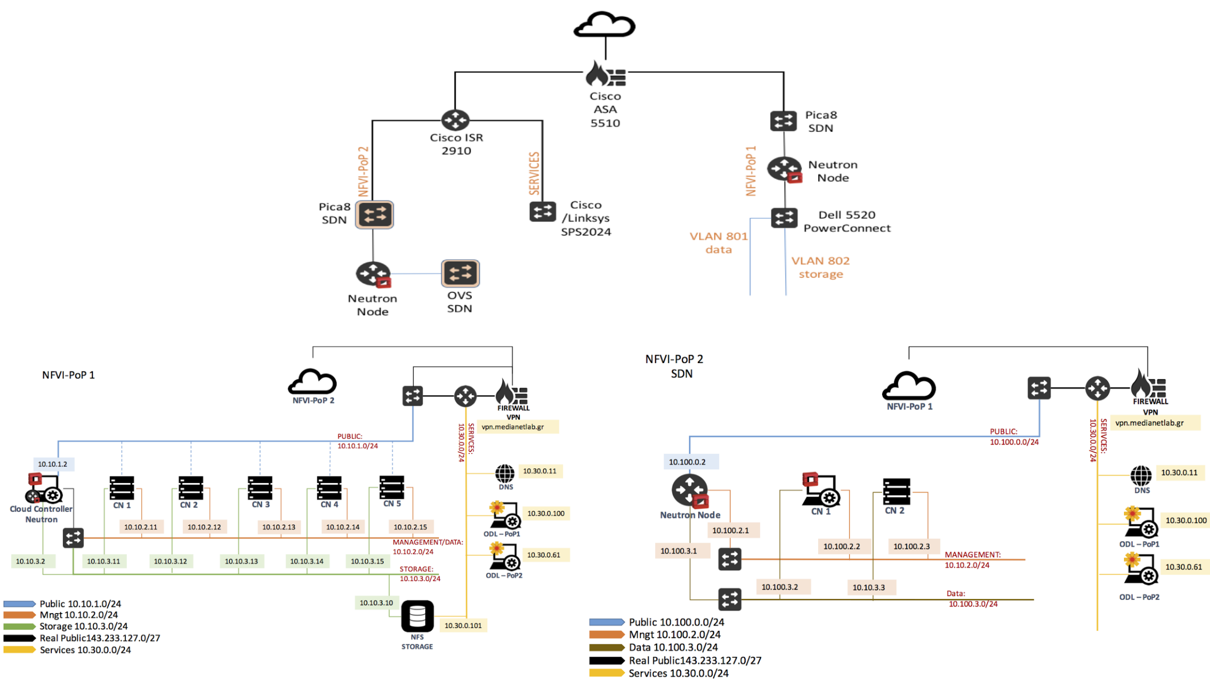


Figure 33 Deployment of T-NOVA Infrastructure

The Deliverable D7.1 “Early Pilot Site Deployment” is focusing on the description of all the previous efforts, consolidating them in a technical guide for the deployment of T-NOVA platform.

8.2. T7.2. Pilot Site Evaluation and Validation

Task 7.2 mainly dealt with the evaluation of the integrated pilot systems. Use cases with specific requirements were defined during the early system design phase to describe the expected system behaviour. The third year testing campaign, in which almost all T-NOVA partners were involved, finally validated these system use cases in the integrated pilot systems. In addition, a quantitative evaluation has taken place to measure the performance of the system components where performance was often associated with the run time metric. Beyond that, the following T-NOVA final demonstrators were presented: Scaling of a VNF (vSBC), scaling of a VNF (vHG), and end-to-end NS deployment over multi-PoP infrastructure. Above results and further details on task 7.2 are documented in deliverable D7.2 (Integrated Pilot and Evaluation Report).

Defined and documented best known methods (BKMs) for infrastructure repository development and integration. Tested functional behaviour of infrastructure repository on the NCSRD and PTIN T-NOVA testbeds. Testing of T-NOVA software in the HPE additional testbed.

8.3. T7.3. Overall Assessment and Roadmap

Task T7.3 analysed overall technical activities in the T-NOVA project and defined key lessons for major system components or technologies used. The results were materialized in deliverable D7.3, "Overall Assessment and Roadmap".

A significant part of T7.3 was the review of relevant external activities, including standardization bodies, open source projects, R&D projects and industry initiatives, thus basically updating the analysis that had been conducted by several WPs in the early stages of the project, while using this renewed vision to position the project accomplishments vis-à-vis the current state of the art.

Multiple variants or extensions of the basic T-NOVA architectural model, enabling a rich variety of use cases, business models and NFV infrastructure deployment options were analysed. A deployment roadmap was proposed, taking incremental deployment as a key requirement for sustainable growth of technology uptake.

Lessons learnt in the course of the project development were identified, with a view to providing useful guidance to other projects and activities exploring the same or similar topics, e.g. architectural concepts, practical SDN/NFV implementation issues, VNFaaS deployment guidelines.

Finally, a number of topics for future study were identified, based on the work performed and the results accomplished by T-NOVA, taking into account relevant technology evolution trends, including the exploitation of NFV in 5G systems.

9. WP8. Dissemination, Standardisation, Exploitation and Training Activities

9.1. T8.1. Market Assessment, Analysis and Planning

As a first step, the task performed an assessment of the market areas and opportunities to be addressed by T-NOVA. The T-NOVA solution is expected to be an attractive revenue source for European network/telecom service providers, who are able to monetize on their infrastructure by offering new services and by charging customers according to the actual usage of in-network resources, as opposed to claiming low, flat fees for plain connectivity services providing applications "over-the-top" with no QoS guarantees and no in-network treatment.

The Market assessment identified the T-NOVA benefits, made an assessment of the existing market also as looked into relevant solutions and market trends in the relevant area. T-NOVA is expected to offer a lot of benefits to business entities such as a network operators, network vendors, software developers, start-up companies and customers. Operators and telecom service providers, for instance, are expected to generate attractive revenue by monetizing their infrastructure and by offering new services and charging customers depending on their usage of in-network resources, charge flat fees for plain connectivity services.

The task also reported a detailed commercial exploitation plan focusing on the network operators, IT service providers and equipment manufacturers within the consortium. The project beneficiaries target to exploit different project outcomes depending on their specific interests. A distinction is made between the partners involved in T-NOVA, namely industrial partners as well as academic partners comprising research institutes and universities. T-NOVA's industrial partners (i.e., operators, manufacturer, and SME) are focusing their exploitation activities on improving their current operation and business position in existing markets, and on the creation of and preparation for new markets, with the intention to secure a strong leadership position in these new markets. Academic partners are aiming to focus and intensify their activities in thematic areas of interest for both the industrial and research communities building at the same time strong technical expertise and presence in the relevant fields.

The task also defined the benefits of T-NOVA focusing on the marketplace. It performed a SWOT analysis indicating T-NOVA's strengths, weaknesses, opportunities and threats.

The partners also looked into reports from NFV surveys and examined the views regarding the specific NFV market and the adoption of similar solutions. The outcome of this exercise indicated that telecom operators are aware of the NFV challenges and as a consequence have certain anxieties relating to the NFV adoption within their organisations.

9.2. T8.2. Standards tracking and contribution

The most important standardization activities of the project included:

- Tracking of the most relevant standardization groups (ETSI NFV ISG in particular, TMForum, OPNFV, IRTF SDNrg and NFVrg).
- Remote participation to the meetings of ETSI ISG NFV (specifically IFA WG, EVE WG, REL WG).
- Participation in the weekly OPNFV Yardstick meetings to track and provides updates on the T-NOVA contributions to Brahmaputra release.
- Participation to TMForum Live (Nice, May 2016).
- Internal Meetings with partners: report on standards and analysis of the VNF descriptor.
- Proposal and implementation of the ETSI PoC #40 VNFaaS and Service Orchestration.
- Contributions to other standardization groups and Open Source communities;
- Contributions to ETSI ISG NFV IFA WG on the VNF descriptor.
- Contributions to ETSI ISG NFV EVE WG on the VIM interconnection scenarios.
- Contributions to ETSI ISG NFV TST WG with the PoC (#40).
- Contributions to TMForum on the NFV Marketplace concept.
- Contributions to the Yardstick project included in the OPNFV Brahmaputra release.

9.3. T8.3. Dissemination and Communication

Dissemination and communication activities were performed using diverse channels towards reaching a wide range of audience. From the early phases of the project, the T-NOVA website and wiki page were created, maintained and updated on a regular basis. More specifically the goal of the T-NOVA website is to provide a high level description of the project's vision, objectives, technical approach, work organization and the involved partners. In addition to this, the website includes information on public project deliverables, publications, events and project news. The website's online activity was monitored through Google Analytics.

T-NOVA has also been exposed on social networks including Twitter, SlideShare and LinkedIn.

Overall, T-NOVA has disseminated its results through a number of events and has raised a lot of interest. More specifically the project results have been presented in number conferences such, IEEE/FIP, EUCNC, SACONET etc and published in flagship scientific journals (see detailed publication list below).

It should be noted that a significant percentage of the papers in the list are co-authored by researchers from more than one partner organisation in the project. This is a direct outcome and demonstration of the strong collaboration achieved between partners in the context of T-NOVA.

Special emphasis continued to be given on the liaisons with other relevant EU projects and the coordination and collaboration activities that have been performed and planned with the aim to exploit the complementarily and increase the efficiency of European funded research. More specifically, T-NOVA identified and had discussions with four EU projects which are relevant to the project, UNIFY and NETIDE, MCN and PACE.

Journal & Conference Publications

1. I. Giannoulakis, E. Kafetzakis, G. Xylouris, G. Gardikis, A. Kourtis, On the Applications of Efficient NFV Management Towards 5G Networking, The International Conference on 5G for Ubiquitous Connectivity (5GU)
2. Z. Bozakov and P. Papadimitriou, Towards a Scalable Software-Defined Network Virtualization Platform, IEEE/IFIP International Workshop on SDN Management and Orchestration (SDNMO 2014)
3. G. Xilouris, E. Trouva, F. Lobillo, J. Soares, J. Carapinha, M. J. McGrath, G. Gardikis, P. Paglierani, E. Pallis, L. Zuccaro, Y. Rebahi, and A. Kourtis, T-NOVA: A Marketplace for Virtualized Network Functions, European Conference on Networks and Communications (EUCNC 2014)
4. Ferrer Riera, J., Hesselbach, X., Escalona, E., Grasa, E., García-Espín, J.A., On the Complex Scheduling Formulation of Virtual Network Functions over Optical Networks, International Conference on Transparent Optical Networks (ICTON 2014)
5. Ferrer Riera, J., Batallé, J., Escalona, E., Grasa, E., García-Espín, J.A, Virtual Network Function Scheduling: Concept and Challenges, International Conference on Smart Communications in Network Technologies (SACONET 2014)
6. A. Abujoda and P. Papadimitriou, MIDAS: Middlebox Discovery and Selection for On-Path Flow Processing, 7th IEEE International Conference on Communication Systems and Networks (COMSNETS 2015)
7. D. Dietrich, A. Rizk, and P. Papadimitriou, Multi-Provider Virtual Network Embedding with Limited Information Disclosure, IEEE Transactions on Network and Service Management
8. N. Herbaut, D. Negru, The path to residential gateway cloud operations, IEEE NetSoft 2015
9. D. Dietrich, A. Abujoda, and P. Papadimitriou, Network Service Embedding across Multiple Providers with Nestor, IFIP/IEEE Networking
10. Z. Cao and P. Papadimitriou, FreeSurf: Application-centric Wireless Access Architecture, IEEE Network
11. Jorge Carapinha, Antonio Cimmino. Authors from D2.1, Requirements and Use cases system for Virtualised Network Functions platforms Journal of Telecommunication Systems and Management, Vol. 3, Issue 2, published in press.

12. A. Abujoda and P. Papadimitriou, MIDAS: Middlebox Discovery and Selection for On-Path Flow Processing, 7th IEEE International Conference on Communication Systems and Networks (COMSNETS 2015)
13. D. Dietrich, A. Abujoda, and P. Papadimitriou, Network Service Embedding Across Multiple Providers with Nestor, IFIP/IEEE Networking 2015
14. D. Dietrich, A. Rizk, and P. Papadimitriou, Multi-Provider Virtual Network Embedding with Limited Information Disclosure, IEEE Transactions on Network and Service Management, Vol. 12, No. 2, June 2015
15. A. Abujoda and P. Papadimitriou, Invariant Preserving Middlebox Traversal, 13th International Conference on Wired and Wireless Internet Communications (WWIC), 2015
16. P. Paglierani, High Performance Computing and Network Function Virtualization: a major challenge towards network programmability, IEEE BlackSeaCom 2015
17. S. Patanjali, B. Truninger, P. Harsh, and T. M. Bohnert, CYCLOPS: A Micro Service based approach for dynamic Rating, Charging & Billing for cloud, 13th IEEE International Conference on Telecommunications (ConTel), 2015
18. M. McGrath, V. Riccobene, G. Petralia, G. Xilouris, and M. A. Kourtis, Performant deployment of a virtualised network functions in a data center environment using resource aware scheduling, IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015
19. B. Meszaros, P. Harsh, and T. M. Bohnert, Lightning Sparks All Around: A Comprehensive Analysis of Popular Distributed Computing Frameworks International Conference on Advances in Big Data Analytics (ABDA), 2015
20. M. A. Kourtis et al., Enhancing VNF Performance by Exploiting SR-IOV and DPDK Packet Processing Acceleration, IEEE SDN-NFV 2015
21. N. Herbaut, D. Negru, G. Xilouris, and Y. Chen, Migrating to a NFV-based Home Gateway: introducing a Surrogate vNF approach, 6th IEEE International Conference on Network of the Future (NoF), 2015
22. J. Ferrer Riera, et al. Modelling the NFV forwarding graph for an optimal network service deployment, International conference on Transparent optical networks (ICTON 2015)
23. S. Battilotti, et al, Resource Management in Multi-Cloud Scenarios via Reinforcement Learning, 34th Chinese Control Conference
24. L. Zuccaro, F. Cimorelli, F. Delli Priscoli, C. Gori Giorgi, S. Monaco and V. Suraci Distributed control in virtualized networks, 10th International Conference on Future Networks and Communications
25. Y. Rebahi, S. Hohberg, L. Shi, P. Comi, B. M. Parreira, and A. Ramos, Virtual Security Appliances: The Next Generation Security, 3rd International Conference on Computing, Management and Telecommunications (ComManTel), 2015
26. A. Abujoda and P. Papadimitriou, DistNSE: Distributed Network Service Embedding Across Multiple Providers, 8th IEEE International Conference on Communication Systems and Networks (COMSNETS), 2016

A number of other publications have been submitted and are under review process.

Further to the scientific journal and conference publications the consortium has performed a variety of other events in order to promote T-NOVA.

Other dissemination events include

1. EU Net-Tech Future Coordination Meeting - Future Internet (FI) Cluster Meeting, Brussels, October 2013
2. Software-Defined Networking (SDN) Concertation Workshop, Brussels, January 2014
3. Pre-FIA Workshop, Athens, March 2014
4. Session on Open platforms and SDN/NFV, FIA Assembly, Athens, March 2014
5. Cloud Computing Conference, Athens, March 2014
6. ENISA Security Conference, Athens, May 2014
7. SDN & NFV workshop, Paris, May, 2014
8. European Conference on Networks and Communications, Bologna, Italy, June 2014
9. Software Defined Networking and Virtualisation Summit, September 2014
10. SDN & NFV 2014 conference, Nice, September 2014
11. EU Net-Tech Future Coordination Meeting - Future Internet (FI) Cluster Meeting, Brussels, October 2014
12. EU-Taiwan Workshop on 5G Research, Brussels, October 2014
13. MEF - SOC, T-NOVA presentation. Explore possible synergies.
14. IETF Meeting 91, Proposed Network Function Virtualization Research Group (nfvrg), Honolulu, Hawaii, November 2014
15. EUCNC Workshop on NFV and Programmable Networks, Paris, France, June 2015, http://www.eucnc.eu/2015/www.eucnc.eu/indexa069.html?q=node/113
16. NFV Workshop, Hannover, Germany, October 2015, https://nfvworkshop.wordpress.com/

Additional presentations have been given by T-NOVA partners a variety of events, such as NetFutures in Brussels and IRTF NFVRG meetings. The T-NOVA demo has been further showcased at several venues, such as NetFutures 2015, IEEE NFV-SDN 2015, and ICT 2015.

Last but not least, the following discussions and liaisons with other projects took place:

- **FP7-MCN:** Discussion with MCN core team and common partners regarding re-use of results and components. The identified areas for collaboration include Billing and Charging framework, and OCCI interfaces. Orchestration components re-use was not considered of interest due to the different view for the Orchestration within the T-NOVA framework.
- **FP7- NETIDE:** Discussions on reusing T-NOVA Marketplace for use in NETIDE App Vault. SDN Listener Service developed in NETIDE to be reused by T-NOVA.
- **FP7-UNIFY:** Discussions on bringing together UNIFY Virtualised Infrastructure and VIM with the Marketplace and the Orchestration Plane of T-NOVA. Discussions are currently addressing architectural issues, with the ultimate target of a common PoC.

- **H2020 - SONATA:** Currently sharing views and experience (via common partners) on the Infrastructure Virtualisation and the VNF and Network Service descriptors.
- **H2020 - 5GEx**: Discussion about the employment of TENOR as one of the target orchestrators to be supported by the 5GEx architecture

9.4. T8.4. Training activities

This Task promoted the training activities of T-NOVA. The training activities include the following:

- 2nd International Summer School on Emerging Architectures and Key Technologies for 5G Networks.
- ZHAW Cloud Computing Summer School 2016
- A Novel Marketplace for Trading/Brokering Virtual Network Functions over Cloud Infrastructures - George Alexiou, Evangelos Pallis, George Mastorakis, Evangelos Markakis, Anargyros Sideris, Athina Bourdena, Constandinos X. MavromoustakisBook title: Cloud and Fog Computing in 5G Mobile Networks,Author: Evangelos Markakis, George Mastorakis, Constandinos X. Mavromoustakis and Evangelos Pallis (Eds.),Year: 2017, ISBN: 978-1-78561-083-7
- Demo of the T-NOVA MANO system (TeNOR Orchestrator & Marketplace) handling the lifecycle of an NFV (Network Functions Virtualisation) service, from VNF onboarding and service creation to service deployment and monitoring
- Post-Graduate and Open Lessons through eclass.teicrete.gr
- Demo of the ETSI PoC #40 "VNFaaS with end-to-end full service orchestration" performed by Eleni Trouva (NCSRD) at SDN & OpenFlow World Congress 2016, The Hague, October 11 - 14, 2016.
- 11 Online Configuration tutorials available at T-NOVA GitHub

Training on the T-NOVA concept and results are not ending with the finalization of the project as part of the material will be disseminated to students through Lifelong Learning activities in the years to come by various partners of the Project.