TNOVA

NETWORK FUNCTIONS AS-A-SERVICE OVER VIRTUALISED INFRASTRUCTURES

GRANT AGREEMENT NO.: 619520

Deliverable D2.42

Specification of the Network Function Framework and T-NOVA Marketplace

Editor Paolo Comi (Italtel)

Contributors Aurora Ramos, Javier Melián (ATOS), Thomas Pliakas (CLDST), Simon Hohberg, Yacine Rebahi (Fraunhofer), Marco Di Girolamo (HP), Enzo Figini, Fausto Andreotti (Italtel), Ahmed Abujoda, Panagiotis Papadimitriou (LUH), Emmanuil Kafetzakis , George Xilouris (NCSRD), José Bonnet (PTIN), Dora Christofi, Georgios Dimosthenous (PTL), Athina Bourdena, Evangelos Markakis, Evangelos Pallis (TEIC) Nicolas Herbaut, Mamadou Sidibe (VIO)

Version 1.0

Date September 30th, 2015

Distribution PUBLIC (PU)

Executive Summary

This document reports the results of the activities carried out in T-NOVA EU-FP7 Project "Functions as-a-Service over Virtualised Infrastructures" by Tasks 2.5 "Specification of Network Function Framework" and Task 2.6 "Specification of T-NOVA Marketplace". It describes the requirements, features description and architectural details in a technology-agnostic manner for later implementation of the Network Function Framework and the T-NOVA Marketplace. This report is the second and final version of the specification work reported in D2.41 that was delivered in September14. The current version includes slight refinements considering feedback from one year of implementation work.

The Network Function Framework is the conceptual element of the T-NOVA system devoted to the definition of the structure and behaviour of the Virtual Network Functions (VNFs). It comprises a Network Function Store, where the VNFs are kept and made available to T-NOVA as building blocks for creating network services. Virtual Network Functions and network Services in T-NOVA are described, traded, and offered to the final users by an innovative Marketplace that opens the NFV market to software developers and traditional service providers for the benefit of large adoption of NFV solutions.

A VNF is characterised by two attributes: the operational functionalities and the management behavior. The operational part explicitly defines the network functions that are supported, while the management part is responsible for the VNF lifecycle. Therefore, a VNF in T-NOVA shall support the APIs for interacting with the T-NOVA orchestration and the virtualized infrastructure, and shall implement the VNF lifecycle described in this report. The VNF metadata is a fundamental part of each VNF. It provides the information for describing how the VNF is composed, which functionalities it provides, and how to manage it. Currently there are many approaches for implementing this concept which are focused on the technical requirements for making a virtual application running. In T-NOVA this information is extended with business aspects that allow the registration and trading of a VNF in the marketplace.

The marketplace concept has been introduced by T-NOVA as a novelty in the NFV scheme in order to facilitate the interaction between the different stakeholders that are identified in the NFV business scenarios. On one hand the VNFs can be implemented by a wide range of developers providing software implementation, and on the other hand, network service providers may want to acquire VNFs to compose network services to be provided to their own customers. The T-NOVA Marketplace has been designed as a distributed platform placed on top of the overall architecture which, besides of including the users front-end, comprises OSS/BSS components as billing and accounting, and innovative modules as the T-NOVA Brokerage to allow trading functionality.

The virtualization of network functions is being addressed by notable standardization bodies such as ETSI and IETF. In particular ETSI has developed a NFV reference architecture and has provided a common language in this area. Therefore, this report looks at ETSI to build on it. The T-NOVA Marketplace specification relies also on ongoing standardization activities such as business best practices provided byTMForum, as it is for instance the integration of a business service catalogue and SLA Management issues in virtualization.

Table of Contents

Index of Figures

Index of Tables

1. INTRODUCTION

1.1. Motivation and scope

Network Functions Virtualization (NFV) constitutes a topic of immense interest to the networking community in the the research/academic domain but also in industry since it is candidate approach for short-term exploitation. Via the concept of infrastructure "softwarisation", NFV has the potential to entirely transform the networking market and open it to new entrants. In this context, T-NOVA introduces a complete open solution for NFV deployment, focusing on the Virtual Network Function (VNF) as a service perspective with a strong business orientation.

In order to provide this business orientation to the NFV scheme T-NOVA develops a novel marketplace that will facilitate T-NOVA customers to select virtual appliances by means of a friendly front-end, "plug" them into their existing connectivity services, configure/adapt them according to their needs and, in the case of network service providers, also allow them to offer Network Services (NSs) composed by several VNFs to their own customers [1].

The service request will be carried out via a tailored customer front-end/brokerage platform that is part of the T-NOVA Marketplace. This marketplace will also provide all the T-NOVA stakeholders SLA and billing functionalities.

On the other hand, T-NOVA introduces an innovative Network Function Store (NF store) following the paradigm of already successful OS-specific "App Stores". This NF store contains VNFs by third-party developers, published as independent entities and accompanied with the necessary metadata for both technical and business description of the VNF.

Software developers willing to sell their VNFs through the T-NOVA marketplace shall extend their implementation of network functions supporting the APIs for interacting with the virtualized infrastructure and the T-NOVA orchestration for the service composition [2], and the VNF lifecycle described in this deliverable.

In this way, thanks to the NF store and the marketplace, it is expected that T-NOVA will contribute to expand market opportunities by attracting new entrants to the networking market. This capability will be particularly important for SMEs and academic institutions which can leverage the T-NOVA architecture by developing innovative cutting-edge Network Functions (NFs) as software modules that can be included in the NF store. This also will enable the rapid introduction of VNFs into the market.

1.2. Relation to T-NOVA overall architecture

T-NOVA overall architecture is described in previous deliverables [3]. The scope of the current document with respect to the T-NOVA architecture is the definition and specification of the (i) marketplace; (ii) network function store (NF store); (iii) VNFs and (iv) assorted interfaces as graphically illustrated in Figure 1-1.

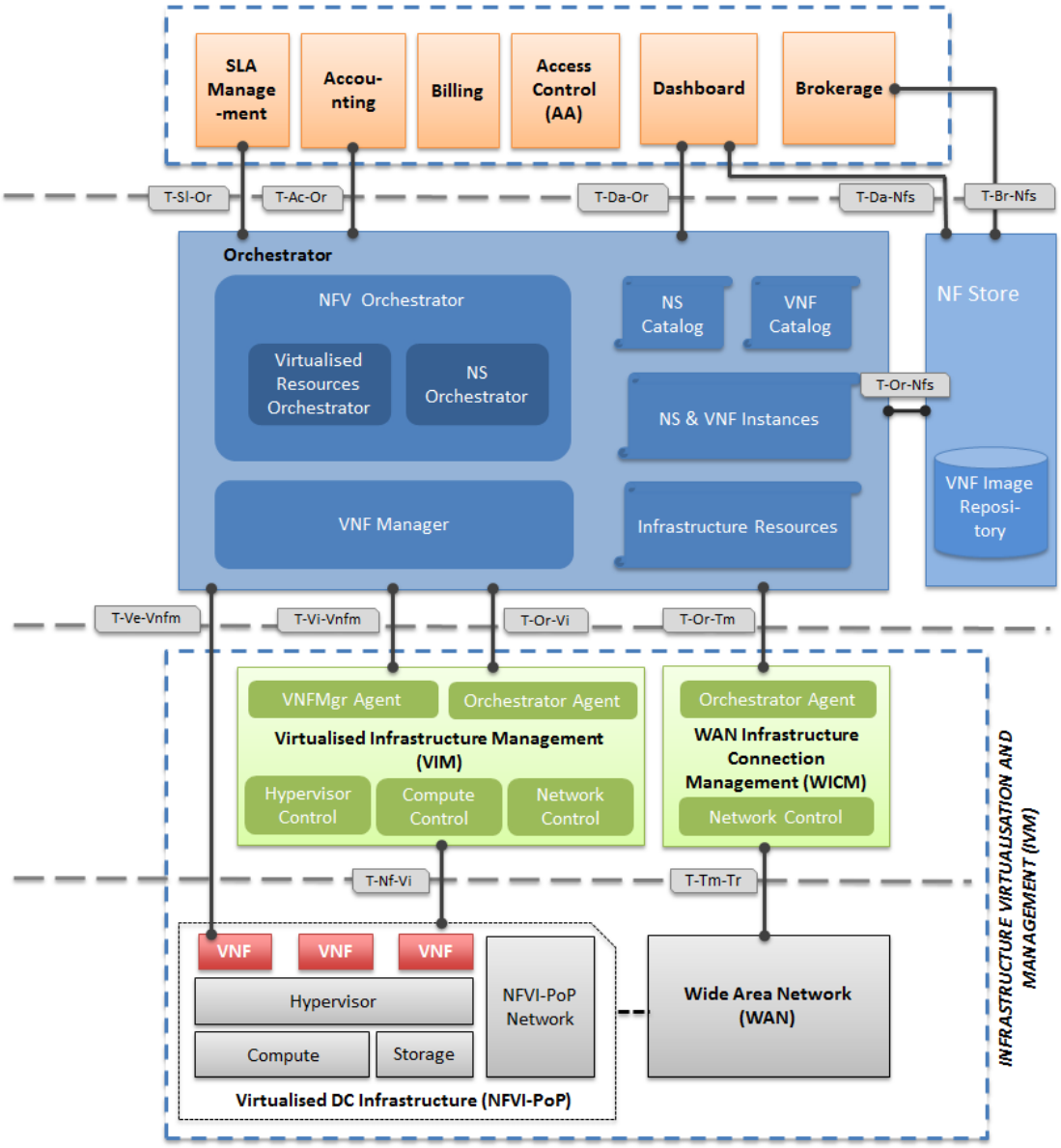

Figure 1-1 Relevance to T-NOVA Overall Architecture

In summary, the T-NOVA Marketplace is a distributed platform placed on top of the overall architecture which, besides of including the users front-end, it comprises OSS/BSS components as billing and accounting, and innovative modules as the T-NOVA Brokerage, being in charge of managing all business relationships among the T-NOVA stakeholders (see section 2.3).

The NF store is mainly a repository for the VNF images and their accompanying metadata. It plays an important role in the VNF lifecycle management as it is the component where the VNFs are published by different NF developers.

VNF lifecycle is also part of the scope for this deliverable. More specifically, information related to VNF deployment, management and termination have interdependencies with various components of the overall architecture. Other stages of the lifecycle refer to interaction between the marketplace and the NF store.

Finally, the interfaces as defined in [3] are also specified in the current document.

1.3. Document structure

This document is structured as follows:

Firstly, Section 2 focuses on the specification of the T-NOVA Marketplace. After an overall description of the objective in section 2.1, section 2.2 contains the state of the art analysis over which the specification of the T-NOVA Marketplace has been performed. Section 2.3 gives an overview on how each T-NOVA stakeholder will iteract with the system and Section 2.4 summarizes the T-NOVA Marketplace lifecycle in order to provide the reader a whole picture of the marketplace functioning. The requirements gathering procedure is explanained in section 2.5, while section 2.6 contains the proper specification of each component in the marketplace according to the requirements specifications listed in Annex A.

Section 3 is devoted to the specification of Virtual Network Function (VNF) structure and behaviour and the design of the Network Function Store (NF store). Section 3.1 explains the scope of the section providing high level description of the subset of the addressed T-NOVA architecture. The description of the VNF structure is the topic of section 3.2. It describes also the information model of the VNF metadata descriptor. A first comparative analysis with ETSI NFV is provided. Section 3.3 completes the VNF description with the definition of the network function lifecycle. The description of VNF structure and behaviour is completed by section 3.4 providing the high level specification of the APIs supported by both the VNF and the NF store. Finally, chapter 3.5 outlines the high level design of the NF store.

The objective of section 4 is to make clearer and remark the relation between the marketplace and NF store, both from a functional and architectural point of view.

Section 5 contains the conclusions gathered from the contents of this document.

Finally, Annex A provides the list of all the requirements gathered for the different marketplace components and NF store, while Annex B contains the initial mock-up of the dashboard done in the specification work;next versions and final design will be included in future deliverables in the project [4] [5].

2. SPECIFICATION OF THE T-NOVA MARKETPLACE

2.1. Objective

The marketplace in the NFV scheme is an innovative concept that T-NOVA introduces with the aim of promoting the VNF service offerings and facilitating the commercial activity and fluent interaction among the different business stakeholders identified in [1]. Besides of providing the Graphical User Interface (GUI) for all of the stakeholders, the T-NOVA Marketplace will facilitate all the necessary features related to the market activity, such as trading, SLA (Service Level Agreement) negotiations and billing.

The components identified in the previous work in T-NOVA [3] that are part of the T-NOVA Marketplace and represented in Figure 1-1 and high level described in Table 2-1:

Name	Description
SLA Management Module	The marketplace functional entity which establishes and stores the SLAs among all the involved parties and checking if the SLAs have been fulfilled or not will inform the accounting system for the pertinent billable items (penalties or rewarding).
Accounting Module	The marketplace functional entity which stores all the information needed for later billing for each user: usage resources for the different services, SLAs evaluations, etc.
Billing Module	The markeplace functional entity that produces the bills based on the information stored in the accounting module.
Access Control Module	The marketplace functional entity which administers security in a multi-user environment, managing and enabling access authorization/control for the different T-NOVA stakeholders considering their roles and permissions.
Dashboard	The marketplace functional entity which provides the user front-end, exposing in a graphical manner all customer-facing services.
Brokerage Module	The marketplace functional entity which enables the interaction among actors for service advertisement, request and brokerage/trading.

Table 2-1 Main T-NOVA Marketplace components definitions

2.2. State of the Art

In order to design and later implement the marketplace context in T-NOVA, we have firstly looked at the most relevant ongoing standardization works when applying network services provision business processes to Network Function Virtualization (NFV). In Sep' 14 TMForum provided some first inputs mapping their standardization document about business process to the ETSI NFV MANO architecture [6]. Further ETSI and TMForum insights are summarized in 2.2.1.

In section 2.2.2 recent research projects implementing a marketplace to provide network services and other projects focused in NFV and software networks are surveyed. In section 2.2.3 we summarize several current commercial solutions that may be aligned with T-NOVA Marketplace approach.

Finally, in section 2.2.4 we extract some conclusions gathered from this state of art in relation to T-NOVA Marketplace.

2.2.1. Standardization activities

2.2.1.1. ETSI ISG NFV

A network operator-led Industry Specification Group (ISG) was setup in the last quarter of 2012 under the umbrella of ETSI (European Telecommunication Standard Institute) to work through the technical challenges of NFV. ETSI ISG NFV in its document on global architecture [6] illustrates the high-level NFV framework, where three main working domains can be identified:

- VNF, as the software implementation of a network function which is capable of running over the NFVI.
- NFV Infrastructure (NFVI), which includes the diversity of physical resources and how these can be virtualised. NFVI supports the execution of the VNFs.
- NFV Management and Orchestration (NFV MANO), which covers the orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualisation, and the lifecycle management of VNFs. NFV MANO focuses on all virtualisation-specific management tasks necessary in the NFV framework.

Relation to T-NOVA Marketplace

As it has been described in the previous work to define overall T-NOVA architecture described in [3], and according to the diagram in

Figure 2-1, the marketplace is completely novel in regards to ETSI view [7]. T-NOVA introduces the marketplace concept aiming at opening the NFV market to third party developers for the provision of VNFs, a concept that currently falls outside the technical view of ETSI NFV.

On the other hand and also introduced in [3], where ETSI MANO has been deeply explained, ETSI NFV does not provide yet any more insight on the OSS/BSS (Operating Support System / Business Support System) of the operator besides the definition of an interface (Os-Ma). Though OSS/BSS systems are not within the scope of T-NOVA, the proposed marketplace contains partially some OSS/BSS functionalities (i.e. billing, accounting, SLA monitoring, Authentication, Authorisation, and Accounting (AAA)), which will be implemented/adapted.

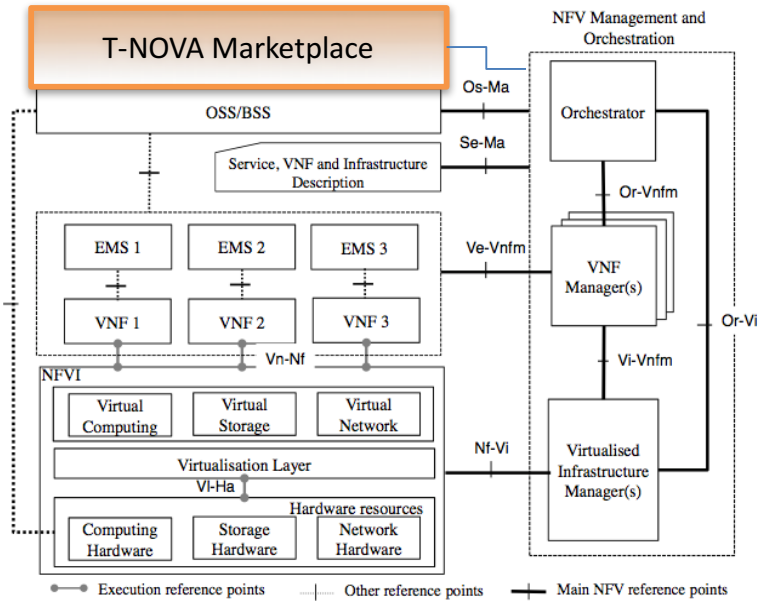

Figure 2-1 **ETSI**

NFV architecture

2.2.1.2. TMForum

The general objective of Tele Management Forum [8], as a global trade association of service providers and suppliers, is the improvement on business agility and the growth of business through knowledge, tools, standard, training and best practices. The specific TM Forum's Agile Business and IT Program aims at optimize service providers' operations reducing costs, risks, and time to market by providing a set of integrated offerings that collects the experience and best practices gleaned from the major players within the industry. The TMF's standards are collectively known as *Frameworx*, which is composed of four underlying components, each aimed at standardizing information models, interfaces, and lexicon:

- *Business Process Framework (eTOM, Telecom Operations Map):* the industry's common process architecture for both business and functional processes. This framework is meant to aid in the creation of a comprehensive, multi-layered view of all of the business processes necessary for a carrier's operation. It provides both guidelines and process flows, and aligns with standards from ITIL (*Information Technology Infrastructure Library*) and other external bodies.
- *Information Framework (SID, Shared Information/Data model):* provides a common reference model for enterprise information that service providers, software providers and integrators use to describe management information. It is used to develop databases and provide a glossary of terms for business processes. The framework is intended to reduce integration costs and to reduce project management time and cost by minimizing the number of necessary changes to underlying architecture during the launch of a new product or service offering.
- *Application Framework (TAM, Telecom Application Map)*: it provides a common language between service providers and their suppliers to describe systems and their functions, as well as a common way of grouping them. It attempts to group the information and processes defined by the eTOM and the SID into recognizable applications.
- *Integration Framework (TIP, TM Forum Integration Program (TIP):* it shows how the business process, information and application frameworks interact to:

- Create a catalog of business services that define functional and non-functional aspects of a service based on service oriented principles;
- Develop a platform or domain-based enterprise architecture that provides the business agility required to compete in today's market;
- Define critical standard interfaces that speed integration.

2.2.1.3. Applicability of TMForum standards to ETSI NFV

TMForum ZOOM [9] is the initiative launched by TMForum focused in NFV, whose TR227 TM Forum Specifications document [10] contains a description of the set of TM Forum documents that are relevant for ETSI NFV MANO work. It identifies areas where each TM Forum document may contribute to standarize the information presented and interfaces of the MANO reference points. Those that may be applicable to the specification of the marketplace are the following:

eTOM:

- Enables MANO to design interfaces and APIs that better reflect how an organization performs configuration, monitoring, and other processes.
- Enables MANO to achieve a better alignment between business processes of an organization and the process flows that are defined by MANO.
- Enables MANO to ensure that the reference points that it designs are appropriate for the business processes of an organization.

SID:

- Provides detailed models in an object-oriented form that can be used to further define MANO service and resource concepts.
- Provides a detailed model of how services and resources are managed, including definition of metrics to represent key characteristics and behaviour of services and resources as well as SLAs.
- Provides a framework to design interfaces and APIs for various business and operational processes.

TIP:

- Creates a common shared integration environment.
- Provides detailed interactions in the form of a set of messages exchanged as a protocol.
- Links business processes to information elements (e.g., the SID model elements).
- Defines standardized templates for the development of new interfaces.
- Increases interface reusability through addressing a broad set of business process scenarios.

Relation to T-NOVA Marketplace

Analysing the document TR228 TM Forum Gap Analysis related to MANO Work [11] we can gather that one of the topics that TM Forum points out as missing from NFV MANO, is a detailed implementation model on how to manage operational and business support systems in a hybrid legacy and virtualized environment, something that ETSI is not addressing so far.

Though being out of the T-NOVA scope the interface between the MANO architecture and the existing OSS/BSS system of operators, T-NOVA aims to provide a first step on the

direction of this research line by means of the implementation of the marketplace, which will implement some of the functionalities of a BSS system of an external operator, and what could be a first input for latest studies in the interoperability with OSS/BSS existing systems, that TM Forum ZOOM intends to address in the future [8].

2.2.1.4. Update on TMForum NFV related work January'15- July 2015

As stated in the previous SOTA survey on September 2014 [12] TMForum did not provide at that date specifications on NFV business management but generic business practices for business agility, which we considered to make T-NOVA Marketplace compliance with, such the SID model and the inclusion of a business service catalogue.

Also in December 2014 TMForum activity was surveyed within T-NOVA [4] specifically in relation to service description, billing and SLA management, when good practices or first insights towards NFV were still a set of objectives of TMForum ZOOM project [9] that we considered to build T-NOVA marketplace framework.

In February 2015, TMForum launches its first research report on NFV: *Virtualization: When will NFV cross the chasm?* [13]. More than technical guidelines this report surveys members and other experts to explore when NFV will 'cross the chasm' to widespread availability, including:

- Definition of NFV based on ETSI NFV architecture.
- Role of SDN in end-to-end provisioning and management of services over NFV infrastructure.
- Barriers for NFV to get into the market.
- Results of a survey of service providers and industry experts on the drivers for NFV, the business and technological inhibitors, plans to adopt a DevOps.

TMForum Catalyst projects

TMForum Catalyst projects [14] are in the form of Whitepapers, Case Studies, Best Practices, lessons learned as input to provide to other teams through to Interface Specifications, Models, Frameworks & Reference Code. In previous work in T-NOVA marketplace [12] we identified one of them particularly in line with T-NOVA marketplace approach:

- *Service Bundling in a B2B2X Marketplace* [15]. This TM Forum catalyst project aims to show how a buyer can bundle a collection of services sourced from different suppliers and deliver them seamlessly to a customer. These components could include traditional network access products, as well as NFV and IaaS products. Concrete business roles and process touchpoints enable a well-defined relationship among players in the value chain to ensure seamless delivery.

More recent current TMForum catalysts projects that can be relevant to T-NOVA marketplace are:

- *Maximizing Profitability with NFV Orchestration* [16]*:* this catalyst aims to demonstrate how to instantiate, monitor and scale Virtualized Network Functions (VNF) based on technical parameters (SLA, QoS, etc.) and business metrics (cost of power, etc.).
- *NFV Management Ecosystem* [17]: an implementation of management and orchestration (MANO) as defined by the ETSI NFV ISG is the platform for this catalyst to try to demonstrate real-time, dynamic management of capacity, performance,

quality of service (QoS) and service level agreements (SLA) as well as enabling real-time billing and compensation.

TMForum reports

Recent TMForum reports that we have identified potentially relevant to T-NOVA Marketplace are the following:

- *IG1123 NFV Readiness: Packaging Virtualized Network Services for Procurement and Operation* [18]**.** This guide reviews the current ETSI NFV terms which might be used to define NFV packaging concepts used by Service Providers (SP) for procurement, packaging used by suppliers selling NFV capabilities, packaging used to define NFV software modules, and packaging for deployment of software images.
- *TR244 TM Forum Information Framework Enhancements to Support ZOOM* [19]. This document describes a portion of the ZOOM information model, which has been integrated into the Information Framework. It defines four concepts fundamental to modeling NFV-based systems (VirtualResource, NetworkFunction, NetworkService, and Graph), as well as two general-purpose concepts (Catalog and Event) that have been used by existing Catalysts, and are also critical for realizing SDN and NFV systems. This report is further explained in T-NOVA service description framework [20], where it will be describe how T-NOVA is applying SID TMForum model to ETSI NFV approach.

Focusing in SLA management in NFV, T-NOVA SLA specification work has been ahead of TMForum, but in line with it, since it has been built following its approach as explained in [4] coming from SLA management in cloud services. In the last year TMForum has released the next two reports:

- *IG1120 Virtualization Impact on SLA Management* [21]: this exploratory report provides initial thoughts on the impact of end-to-end SLA management in a fully software-defined and virtualized environment, i.e., Cloud-SDN-NFV. Its objective is to leverage knowledge and experience from the TM Forum SLAM work, and apply it to virtualized environments.
- *IG1127 End-to-end Virtualization Management: Impact on E2E Service Assurance and SLA Management for Hybrid Networks* [22]*:* This Application Note brings out the challenges and impacts on end-to-end Service Assurance and SLA management in a hybrid physical/virtualized environment. This change introduces a host of actors, each responsible for part of the overall solution, and creates new SLA/OLA constructs. It also necessitates new means of operations – having SLA-linked Policy Orchestration, new RCA rules for Fault Correlation, and automated closed loop controls, thus reaping the benefits of virtualization as design principles.

These last reports will be detailed in the future deliverable devoted to SLA and billing in T-NOVA [23].

Relation to T-NOVA Marketplace

From the service provider point of view, the marketplace has been included into the T-NOVA system taking into account the approach proposed by TM Forum, as it is for instance the provision of business interaction agility for all the stakeholders, the creation of a business service catalogue to provide the service offerings easier for the customer, or the adoption a common language for service description that it will be detailed in future T-NOVA deliverables [20] . TMForum SID model will be applied to NFV with in T-NOVA. Also T-NOVA

SLA management framework has been built considering the references from TMForum. However, it has not been found any reference for the accounting and billing framework of NFV services that T-NOVA is building.

2.2.2. Other projects

In this subsection we reference research projects implementing marketplaces for network services provision:

- *XIFI* [24]. This is a project of the European Public-Private-Partnership on Future Internet (FI-PPP) programme with the objective of facilitate the uptake, deployment and federation of several instances of a common market platform to pave the way for a unified European marketplace that is crucial for enabling commercial exploitation of FI resources. Inputs from this project in relation to service description languages and catalogues have been surveyed when building T-NOVA service framework.
- Other recent research projects that may be relevant to T-NOVA as it was identified in previous work [3] are: MCN [25], Unify [26] and NETIDE [27]. None of these includes a marketplace as it is in T-NOVA in which several function developers co-exists with full support for their commertial activity. In order to re-use existing solutions, T-NOVA has considered the MCN charging platform to be adapted to T-NOVA ecosystem. This is further explained in section 2.6.2.8.

2.2.3. Commercial products

We have identified some commercial products that may be related partially to the T-NOVA Marketplace:

- *CENX Ethernet Lifecycle Manager (ELM)* [28] is a software-based solution that automates lifecycle management specific to carrier Ethernet and IP services across data network infrastructures and evolving the concept of SDN/NFV. It provides a visual representation of all inter-carrier Ethernet services, in order to allow Service Providers, Access Providers and Cloud Exchange Providers to deliver quality connectivity services. Some features are ordering automation, big data analytics, service orchestration and delivery across multiple operator networks and visualisation of transport services end-to-end.
- *Equinix Platform - Marketplace* [29] is a platform where users can buy and sell connectivity among other services. Provides service-offering capabilities for sellers and buyers have access to wide area of services across multiple operators.

HP has some exemplary cases of marketplaces built up around ecosystems which coalesce the relevant stakeholders. Such cases are:

- HP OpenNFV Partner Ecosystem [30]: around its OpenNFV technical platform, HP created an ecosystem of partners, classified as Technology, Application or Service partners. Part of this ecosystem are the OpenNFV Labs, where HP and its partners can test and validate applications to make sure they run as expected on the OpenNFV reference architecture. Based on the outcomes of testing, each VNF is labelled as either silver, gold, or platinum level.
- SDN App Store [31]. In 2014, HP launched its SDN App Store, meant to act as a market accelerator tool for its ecosystems partners, complemented by a portfolio of consulting services addressed to the end customers. The SDN App Store makes

available to the developer community a centralized platform, able to interconnect with customer premises.

- Cloud28+: The Cloud28+ [32] initiative created a European ecosystem of cloud service providers, builders and developers, leading to the creation of a larger and unified service catalogue. Cloud28+ is not actually a marketplace, since it doesn't take any compensation in return of its service of matching customer demand with service offering. It is a federated service catalogue, defining the services through a common specification, describing their performance, and connecting them to the demand side (-> customers).

Relation to T-NOVA Marketplace

When looking at some commercial solutions we have identified the HP solutions above as ones of closest ones to T-NOVA Marketplace approach but with differences. More concretely, the OpenNFV Partner Ecosystem is a testing platform for developers more than a marketplace, however this open view of a testing environment it can be a very interesting approach to potentially extend T-NOVA Marketplace with open access following a sandbox concept, including free training documentation and possibility of free trials. This will make T-NOVA exploitation roadmap [33] shorter. SDN app Store is aligned with approach of T-NOVA Function Store, while Cloud 28+ can be considered a marketplace but fully cloud oriented, not network services are considered.

2.2.4. Conclusions

Analysing the state of art, we have found some solutions from which we can build on in order to develop the T-NOVA Marketplace; however at this stage it does not exist a proper marketplace to deliver VNF as a Service.

The T-NOVA Marketplace is designed considering the above solutions, the current NFV ETSI architecture, the on-going TMForum Best Practices for business services delivery and SLA management.

2.3. T-NOVA stakeholders interacting with the Marketplace

Starting from the business roles analysis performed in Deliverable D2.1 [1], we concluded that all of the T–NOVA roles identified can be grouped to be played by three different stakeholders which we have called the basic stakeholders in the T-NOVA landscape. For simplicity it has been decided from the implementation point of view that only the basic stakeholders will be considered when specifying the T-NOVA system. Nevertheless, interesting future work for T-NOVA system in general, and for T-NOVA Marketplace in particular, will be its extension for multiple non-basic T-NOVA stakeholders.

Therefore, three different stakeholders will exploit the T-NOVA Marketplace: Customer, Service Provider (SP), and Function Providers (FPs). This situation is reflected in Figure 2-2.

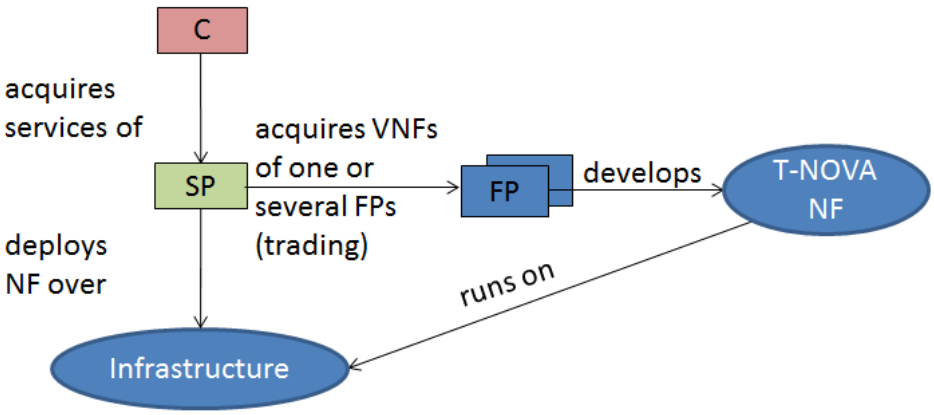

Figure 2-2. Stakeholders interacting in T-NOVA Marketplace

The T-NOVA Marketplace will facilitate the following functionalities for the T-NOVA stakeholders:

- Customers will be able to browse and select network service offerings that best match their needs, as well as negotiate SLAs and exchange their billing information with the SP, keeping track of all the services purchased.
- A Service Provider will be able to acquire VNFs, interacting with different network function developers through a brokerage/trading procedure not only for its own needs but also to offer composed Network Services (NSs) to its own customers. Therefore, also SLA and billing information between SP and FPs will be managed.
- Several Function Providers (network function developers) will be able to publish their VNFs to trade them by means of T-NOVA Marketplace.

Besides these three basic stakeholders, T-NOVA will have its own T-NOVA operator that will be in charge of the system, and the access control for the rest of the stakeholders.

2.3.1. Subscription management

As the T-NOVA Marketplace will be accessed by three previous main categories of users or stakeholders, and each of these categories has a specific task, it is crucial that the marketplace implements a Role Based Access Control (RBAC) framework that:

- restricts account access only to authorized users,
- links a user to an account and assigns the user to some specific roles,
- links each role to specific permissions or profile. For instance, a FP will be allowed to upload VNFs in the function store or to upgrade them. If this provider wants to purchase a service, it will also be assigned the role of a customer. In other words, this framework will allow users to perform actions according to the assigned roles.

As a first step, the stakeholders need to register on the T-NOVA system. Here typical information such as provider name, user name, address, wished role, email address, etc., will be requested in order to create an account on T-NOVA for this user. If this user wishes to use an existing account (on Google, Yahoo, etc) for authenticating himself, an account on the T-NOVA system will be automatically created for him in (see details in section 2.6.2.2.).

2.3.2. Trading mechanisms

One of the added values of the T-NOVA marketplace is the possibility of auctioning among several FPs. In this way, the SP, and T-NOVA customers in general, will benefit from this feature by receiving the best price option, based on their requirements of service description and SLA level.

The principal objective of the T-NOVA auction is to trade the VNFs between the SP and several FPs to achieve best price option for T-NOVA customer on a competitive basis, while maximising the revenue that the auctioneer raises and efficiently exploiting the VNFs. Given those objectives, the design of T-NOVA auction system must also address some challenges an auctioneer faces. Briefly, these challenges include winner's curse (this situation occurs when one or more users pay too much for the auctioned item(s)), collusion (most of the times caused in small size markets) and signaling (less signaling simplify the auction process and attract bidders).

Considering trading models, two main approaches exist, the reserve price estimation (or fix-price) and the auctions. The former case refers to the lowest price the seller is willing to accept for a given item (i.e. service), without performing bids. The latter case performs bids based on single, multiunit or combinatorial auctions. There are plenty of auction protocols, including the sealed first-price auction, the sealed second-price auction (called Vickerey auction), the open ascending-price (English) auction and the open descending-price (Dutch) auction. The main ones are collected in Table 2-2. More information regarding the auction theory and the types of auctions can be found in [34].

In T-NOVA marketplace, the auctioning will be performed by means of the brokerage module (see section 2.6.2.3.).

Trading Mechanisms	Usage
Fix-price	In cases that the offer of NFs is lower to the demand
Auctions	In cases that the offer of NFs is higher to the demand
Vickerey/Sealed bid auction	• Optimize social welfare (the price at which charged the winner depends on the bids of competitors and none of the individual offer) • Avoid signaling • Truethfull in case of second price
English auction	• Quite profitable type of auction for the seller/ Reach to large amounts • Winner's curse is not possible to be prevented (bidders overestimate the value of the item) • Requires signaling • Not truethfull
Dutch auction	• Very profitable type of auction for the seller. The purchase price was not allowed to drop too much, due to the fear that another bidder will first acquire the item • Requires signalling • Not truethfull
Combination of English/Dutch and Vickery	• Decrease the collusion

auctions	
Combinatorial auction - Vickerey-Clark-Grooves	• Sell package of items – winner claim all or nothing • Truethfull • Very Complex

Table 2-2 Overview of Trading mechanisms

2.4. T-NOVA marketplace use cases - lifecycle

Based on the T-NOVA use cases in collected in [1], we describe in this section the whole marketplace lifecyle for a customer accessing the system to purchase VNFaaS, or Network Services (NSs) based on VNFs. This general procedure is represented in Figure 2-3.

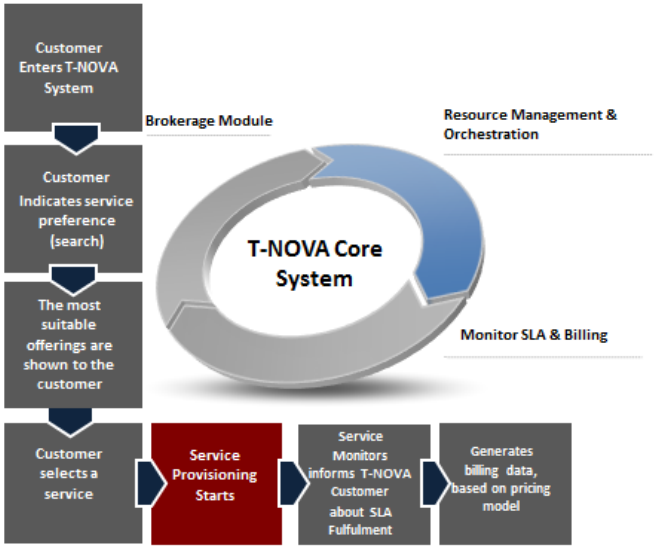

Figure 2-3 Marketplace lifecycle from the customer's perspective

Background:

- The Function Providers (FPs) that want to sell their VNFs through T-NOVA system enter the system providing their VNFs required information (according to sections 3.3.2 and 3.4.2).
- The Service Provider (SP) that wants to purchase VNFs in order to later sell NSs based on VNFs through T-NOVA login to the system.
- A business service catalogue is filled offline in the marketplace when a new service composition has taken place triggered by the SP. This business service catalogue will contain all the information of the commercial offerings available (more information in 2.6.2.4. Business Service Catalogue).

1. The T-NOVA customer enters T-NOVA system through the dashboard identifying (or registering if it is its first time).

2. The T-NOVA customer indicates the service (VNF, several VNFs or Network Service) he/she would like to purchase (perform search).

3. The most suitable offerings available in the marketplace business service catalogue will be shown to the customer in order for him to select one, including price and SLA options. In the event that there is not any available service offering in the business service catalogue matching the customer request, a new service composition should have to take place and trading mechanisms will be performed among FPs if several FPs offer similar VNFs dynamically.

4. The customer selects one of the offered services, together with the configuration of specific technical configuration parameters needed for the service provision, and the SLA agreement procedure will be initiated.

5. All the related information is stored in the different marketplace modules (customer profile, SLA, accounting, etc.).

6. Service provisioning starts.

7. Service monitoring information, SLA fulfilment information and billing information will be made available through the dashboard each time it is required by the customer when accessing the T-NOVA marketplace.

8. Billing will take place when finishing the services in pay-as-you-go services, or each time the assigned bill cycle finishes for the rest of services (between customer and SP, but also between SP and FPs).

2.5. Requirements for T-NOVA Marketplace

The requirements capture process has focused on identifying the desired behaviour for the T-NOVA marketplace and its components, which were most of them identified based on the previous requirements analysis performed at T-NOVA system level [1].

None of the marketplace components requirements in this deliverable specifies how they will be implemented; implementation details will be specified in the future work in technical T-NOVA WorkPackages (WPs) as the implementation-specific descriptions are not considered to be requirements. The goal of these requirements is to develop an understanding of what the marketplace components need, how they interact between each other, and their relationship to the overall T-NOVA architecture [3]. Additionally, the T-NOVA use cases [1] were also considered and cross-referenced with marketplace components requirements.

Requirements were primarily anchored to the existing T-NOVA use cases and the interactions with the whole system both in terms of the actions and requests that would be expected. Additionally the high-level data/information that is be required by the marketplace to successfully deploy its functionalities was also identified. Identified requirements were primarily functional since they are related to the behaviour that is expected from the marketplace.

Using a systems' engineering approach the high level architecture for the marketplace was described in Deliverable 2.22 [3], each component of the overall system was specified in terms of high-level functional blocks. This approach identified the following functional blocks:

- Dashboard
- Access Control
- Brokerage module
- SLA management module
- Accounting module
- Billing module

Also a Business Service Catalogue has been identified to be part of the Marketplace matching TMForum proposal for business agility.

The requirements of T-NOVA Marketplace components can be found in Annex A - Requirements Specification as well as the requirements describing how they should interface one with one another. These requirements were used as a foundational input into the specification of the overall marketplace architecture and its constituent components, which are presented in section 2.6. The coverage of these requirements by the T-NOVA system will be explanained after implementation work.

2.6. Specification of the T-NOVA Marketplace architecture: components and interfaces

Based on the requirements performed at system level [1], the initial approach for the marketplace architecture included in Deliverable 2.22 [3], the requirements gathered for each component in the marketplace, and refinements done in the WP devoted to marketplace implementation [4] is depicted in Figure 2-4, including the overall diagram for the T-NOVA marketplace architecture with both the external and internal interfaces:

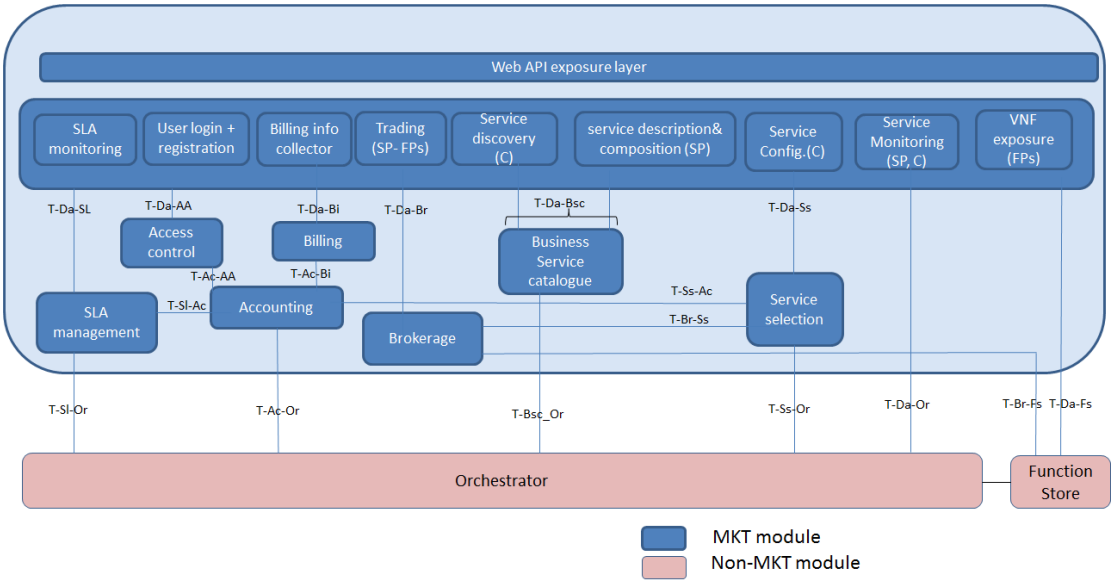

Figure 2-4 Marketplace architecture

Table 2-3 and Table 2-4 collect a brief description of the purpose of each external and internal interface despicted in Figure 2-4. Extended information of external interfaces of the marketplace is explained Section 2.6.2. The specification of each module's functionality, internal architecture and their interfaces are addressed in sections 2.6.3 to 2.6.8.

Marketplace External Interface	**Description**
T-Da-Or	It is used to get monitoring information of the service by the customer and SP.
T-Ss-Or	It is used to notify the orchestrator about a new NS instantiation including the service configuration.
T-Sl-Or	SLA module requests currently running NS metrics from the monitoring system in the orchestrator.
T-Ac-Or	The accounting is notified about any status change of each Network Service (NS) or VNF instances.
T-Bsc-Or	The BSC uses this interface to push the NSD relevant fields to the orchestrator when a new service offering has been created. Also once the orchestrator validates it, the availability of a service is notified to the BSC to be offered to the customer.
T-Br-Nfs	The brokerage module will use it to retrieves information about the

	available VNFs.
T-Da-Nfs	It is used to upload VNF and metadata by the FPs to the Function Store

Table 2-3 Marketplace external interfaces

Marketplace internal interfaces	Description
T-Ac-AA	It is used by the accounting module to access the "user profiles".
T-Ac-Bi	All the information needed for billing is stored in the accounting module.
T-Sl-Ac	SLA module is accessed by the accounting module to extract information about SLA violations for penalties to be applied.
T-Br-Ss	This interface is used by the Service Selection module to check any potential change in the price as a result of the trading process before creating an entry in the accounting.
T-Da-Sl	It is used to introduce SLA templates in the SLA module when a new service of VNF is created and also to provide the dashboard with SLA fulfilment related information.
T-Da-AA	It is used to provide and collect all the information necessary to authenticate the T-NOVA users.
T-Da-Ss	Once a Customer selects a service in the BSC from the dashboard, it is managed by the service selection module in order to provide the custom service configuration.
T-Ss-Ac	It is used to create the entries in the accounting module to track every service or VNF instance created in the orchestrator for billing purposes.
T-Da-Bi	The three stakeholders use it to visualize billing information.
T-Da-Br	It is used to request VNFs in order to facilitate auctioning among FPs.
T-Da-BSC	It is used to publish offerings by the SP, and to browse offerings by the customer.

Table 2-4 Marketplace internal interfaces

2.6.1. External Interfaces to the T-NOVA Marketplace

The marketplace modules will communicate with other two T-NOVA components: the orchestrator, and the function store.

2.6.1.1. Orchestrator

The T-NOVA Orchestrator, which specification can be found in Deliverable 2.31 [2], deals with the optimal deployment of network services instances, as requested by the customer or the Service Provider (SP) on the marketplace, according to a yet to be designed algorithm, the required SLA and the current status of the available infrastructure.

While all network services instances have been instantiated and are running, it is also the orchestrator's responsibility to follow the available metrics, both from the infrastructure and from the service metrics. In order to meet the agreed SLAs, the orchestrator may scale out or up the supporting insfrastructure, communicating such changes to the marketplace, so that a change in accounting is registered and later billed to the customer. Later, if the scaled

(out or up) infrastructure is perceived as being more than enough to fulfill the SLA, it can be scaled in or down. Throught all this process, the orchestrator must provide the marketplace with meaningfull metrics showing how Network Service (NS) instances are working. Details of the interfaces between the orchestrator and marketplace modules are explained in the following sections: Dashboard, Business Service Catalogue. Accounting module Service Selection moduele and SLA management module.

2.6.1.2. Network Function Store (NF store)

As it will be explanained in section 3.5, this T-NOVA component will store the VNFs images and metadata that the marketplace, more concretely the brokerage module, will use to perform trading mechanisms among Function Providers (FPs), to later include those VNFs in the service composition process performed by the orchestrator.

For a VNF to be part of a service composition process, it is necessary that the orchestrator make it available, according to T-NOVA Orchestration specification in D2.31 [2]. Whenever a VNF is uploaded, updated or removed from the NF Store, the orchestrator is informed in order to update its internal registers. This process makes the VNF available in the Function Store to be retrieved by the brokerage module.

2.6.2. Marketplace modules specification

2.6.2.1. Dashboard

With the aim of creating a single entry to the T-NOVA system that provides simplicity for the different T-NOVA users or stakeholders, a unified T-NOVA Dashboard will be designed, taking into account the different roles of the T-NOVA Marketplace. This common dashboard for the whole T-NOVA environment will host three views for the three basic stakeholders that will access the T-NOVA Marketplace: the Service Provider (SP), the Function Provider (FP) and the Customer.

Starting from the dashboard requirements in Annex A, in this section we include the general description of the information that the dashboard will have to show, first ideas for its design and the general information that will have to be collected by the different APIs of the dashboard coming from the rest of the T-NOVA components. The complete design information and later implementation of T-NOVA user dashboard will be done in T6.3, and its outcomes will be gathered in the Deliverables D6.01 and D6.3.

Functionality

The main features of the dashboard are presented in Figure 2-5. Each view is allocated with specific functionalities stemming from the requirements gathered from Annex 6.1.1.

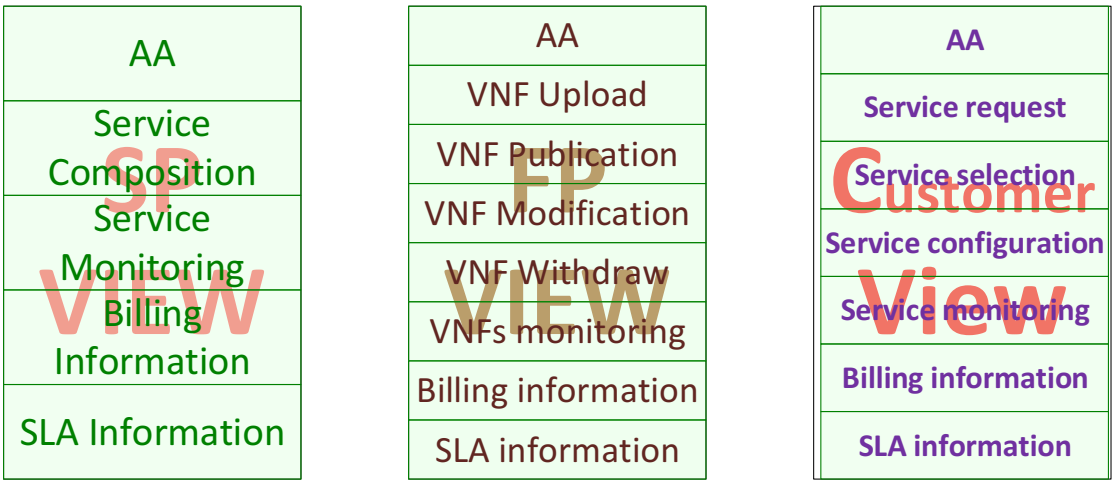

Figure 2-5 Dashboard views

The SP view of the dashboard will allow the SP to perform the functionalities shown in Table 2-5.

Functionality	Short Explanation
AA	Authorization and Authentication of the respective role into the T-NOVA Dashboard.
Service composition	Graphical wizard that will help the SP to compose a new Network Service (NS) starting from the brokerage among the FPs owing the available VNFs.
Service monitoring	Graphical representation of all monitoring data for a selected or "consumed" Service.
Billing information	Graphical representation of the billing outcomes of selected or "consumed" service. There will be two types of billing information for the SP: - Charges for the SP's customers (BSS functionality). - Invoices on behalf of its own suppliers, the FPs.
SLA information	Details of the selected or "consumed" service based on how they respect the agreed SLA. The SP will have accessed to two different kinds of SLA contract and SLA monitoring information: - SLA between SP and its customers (BSS) - SLA agreed with his its suppliers, the FPs

Table 2-5 SP dashboard view

The FP view of the dashboard will allow the FPs to perform the functionalities shown in Table 2-6.

Functionality	Short Explanation
AA	Authorization and Authentication of the respective role into the T-NOVA dashboard.
VNF Upload	Graphical wizard that will help the FP to upload his VNF with

	the necessary parameters.
VNF Publication	Graphical representation for the FP to provide the last check in order to publish the uploaded VNF
VNF Modification	Small graphical wizard that provides the ability to the FP to modify the uploaded VNF.
VNF Withdraw	Graphical representation that gives to the FP the ability to remove an already published or uploaded VNF
VNFs monitoring	Graphical representation of all monitoring data for a selected or “consumed” NF.
Billing information	Graphical representation of the Billing outcomes for a selected or “consumed” NF.
SLA information	Information of the selected or "consumed" NFs based on the agreed SLA and its fulfilment.

Table 2-6 FP dashboard view

The customer view of the dashboard will allow the customer to perform the functionalities shown in Table 2-7.

Functionality	Short Explanation
AA	Authorization and Authentication of the respective role into the T-NOVA Dashboard.
Service request	Graphical representation of the Services/Functions returned by the T-NOVA business service catalogue.
Service Selection	Graphical representation assisted by a check box providing the ability to the customer to select a service for consumption.
Service configuration	Small Graphical wizard providing to the customer predefined parameters for defining the selected service.
Service monitoring	Graphical representation of the data gathered from the monitoring modules.
Billing information	Graphical representation of the billing outcomes of selected or "consumed" service.
SLA information	Details of the selected or "consumed" Service based on how they respect the agreed SLA.

Table 2-7 Customer dahsboard view

Design

The dashboard constitutes the T-NOVA system front-end, as offered to the Customer, the SP and the FPs for service consumption, discovery, interaction, publication, etc. In order for the dashboard to be as up-to-date as possible and terminal-agnostic, a web-based design has been selected.

Furthermore, the dashboard shall be able to meet and, if necessary, to adapt to the specific stakeholder’s needs/requirements as much as possible providing the best experience to a specific stakeholder. This implies that the implementation shall achieve a flexible service

presentation by means of an appropriate choice of technologies and tools. T-NOVA will allow every role to personalise some settings such as interface, appearance and content according to its profile.

More specifically, in the authentication stage, all stakeholders share a common layout that displays the generic graphical interface composed by the basic controls that enable stakeholder specific authentication. Once authenticated, every stakeholder will be able to customize the overall experience according to a set of preferences and his profile.

The main design decision gathered from the requirements in [1] has been to have a common dashboard with different customized views based on different roles. Furthermore and by gathering all the requeriments we have designed a first version of a mock-up for the dashboard that will be used as a Pilot for the upcoming work in Task 6.1 – User Dasboard. The mock-up is available in Annex B 6.2.

Interfaces

The information that will be collected from the rest of T-NOVA components to be used by dashboard will be provided through the following APIs:

AA: the Authentication and Authorization access control system will provide an API to the dashboard to provide and collect all the information necessary to authenticate the T-NOVA users or stakeholders.

SLA management: the goal of this API is to show the users the following information coming from the SLA management module:

- SLA template specification to be filled by the SP and FPs.
- SLA offering to the customer and associated to each service.
- SLA fulfilments by all the stakeholders.

Note: the SLA selection performed by the customer to manage the SLA negotiation process and the SLA contract information will come from the brokerage module that performs the trading.

The SLA front-end tool that will be integrated in the dashboard will be also responsible to make the correct request to the SLA API and then gather and show the results.

Brokerage

This interface is exploited for trading issues, among the T-NOVA users (i.e. SP, FP) and the brokerage module. The information that will go through this API will be related to:

- VNF request and selection: by means of this API the SP requests and selections will be sent to the brokerage module.

- Advertise VNF: this functionality is exploited for the communication between FP and the brokerage module, as the latter perform the intermediate communication, this is trading.

Orchestrator

The interface between the dashboard and the orchestrator will be used to manage service usage data: through this interface the SP and customer will be able to get the monitoring information of the service.

Billing

The billing API for the dashboard will have to manage the following information between dashboard and billing module:

- Bills charged per user and per service (SP and customer).
- Charges done to SP's customers (BSS functionality to the SP).
- Charges done to FP's customers, which is the SP.

Business Service Catalogue

The Business Service Catalogue API for the dashboard will be used to:

- Publish and on-board service offerings by the SP to T-NOVA.
- Browse available service offerings by the customer.

Service Selection module

This API will have to manage the information needed to configure each service to be customized for the customer when selecting each service, for instance providing customer's network details for later the orchestrator properly deploy the service. Also used to update on configuration or need to remove a running NS.

Network Function Store

This interface allows the FPs to publish and manage their VNFs into the NF Store. The publication consists in uploading the VNF image, registering the VNF and its metadata into the function store. The VNFs are versioned allowing the FPs to provide further upgrades. Finally, the FPs can remove their VNFs. In summary, the information managed with this interface is:

- VNF image and VNF metadata descriptor.
- VNF version.
- Upload, upgrade and delete the VNF package.

2.6.2.2. Access control (AA)

In T-NOVA, different stakeholders are foreseen. Each of these stakeholders will have a specific role and accordingly some associated permissions (see below in this section *Policy Enforcement Service*). For instance, a Function Provider (FP) will be able to upload a VNF and upgrade it if needed. The SP will be able to select the VNFs that he is willing to deploy and use, and should not be allowed to upload/remove a given VNF from the NF Store. One of the main challenges in T-NOVA is how to administer security in a multi-user environment. To address this issue, T-NOVA will specify and develop a lightweight Role Based Access Control (RBAC) system where decisions are based on the functions a given stakeholder is allowed to perform within T-NOVA.

The requirements gathered for this module are collected in Annex A. The main conclusions are summarized in the following bullets:

- The different stakeholders should be authenticated before any operation on the T-NOVA system.
- The different stakeholders should be authorized to perform tasks that are associated to their roles and permissions.
- Roles are created according to their functions in T-NOVA, and stakeholders are assigned roles based on their responsibilities and qualifications.
- Roles can be reassigned or granted new permissions if needed.
- Roles and permissions should be updatable and revocable.

Functionality

The RBAC system will be offering two main functionalities:

- *Authentication:* authentication is the process by which the system will verify that a user of T-NOVA is exactly who he is claiming to be.
- *Authorization:* authorization is the process by which a user is allowed to perform the tasks he wants to.

Architecture

The general diagram of the access control system is depicted by Figure 2-6.

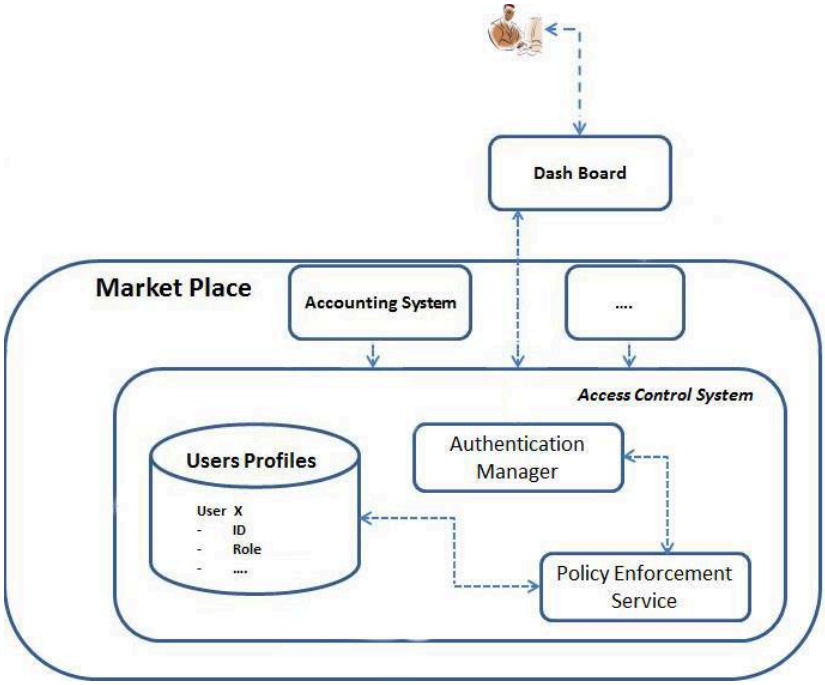

Figure 2-6. RBAC high level architecture

Authentication Manager

To enable the T-NOVA system to provide different functionalities to the stakeholders, a mechanism for authenticating a stakeholder is required. In T-NOVA, this is performed by the *Authentication Manager,* allowing a user to register and login with username and password. In the registration case, the user has to provide the information required (username, password, email etc) by filling out a registration form. Finally thought the authentication process the *Authentication Manager* returns a JWT authentication token reflecting that the user is logged in.

Policy Enforcement Service

The *Policy Enforcement* service implements a Role Based Access Control (RBAC) mechanism that allows assigning users different roles resulting in different rights. Such an access control mechanism allows the T-NOVA system to implement functionality such as uploading a VNF or purchase a service. When a new user registers a user profile will be created containing the name and email address of the user. Furthermore the profile will also contain the current role of the user.

The roles foreseen at this stage of the project are:

- T-NOVA operator: in charge of the T-NOVA system
- Service Provider: it purchases several VNFs to compose a service to be sold to its final customers.
- Function Providers: the entities that are allowed to upload and upgrade a given VNF on the T-NOVA system.
- Customer: the entity interested in purchasing a T-NOVA service.

Interfaces

Several interfaces are foreseen to ease the communication with the other parts of the T-NOVA system. This includes:

- *Interface to the dashboard:* the T-NOVA Access Control module will provide an API to the dashboard allowing it to authenticate the T-NOVA users.
- *Interface with all the other compoments in the marketplace:* the T-NOVA Access Control System will provide an API to access the "User profiles" and features are needed to handle information, such as user permissions, personal information etc.

2.6.2.3. Brokerage module

Towards facilitating trading between diverse actors in the NFV scene the T-NOVA marketplace includes an innovative brokerage module, in which VNFs by several Function Providers (FPs) can be brokered/traded.

Functionality

Via the brokerage module API in the dashboard, the SP place their requests and requirements for the corresponding VNFs, receive offerings and make the appropriate selections, taking into account the price and the offered SLAs. Trading policies such as long-term lease, scheduled lease, short-term lease or spot markets (these leasing types refere to the duration of VNFs exploitation) can be based either on fixed-price or action-based strategies (see section 2.3.2 Trading mechanisms).

In T-NOVA there are several objectives in order to select an auction mechanism, which should be taken into account. The first objective is to avoid too much signalling overhead. This objective may be satisfied with the *sealed-bid auction*. In this auction scheme, bidders simultaneously submit sealed bids so that no bidder knows the bid of any other participant. Hence, bidders cannot change their bids after the announcement of the other bids. In the case of sealed-bid auction the first price auction model should be implemented. Sealed-bid auction may not be truthful (truthfulness prevents market manipulation, since the bidding is performed considering the true value of the item), however the VNFs auctions are often organized to maximize the payoff, and not to be truthful.

To be precise, the implementation of the second price auction model is also possible since there is no problem in switching the payment method in an auction engine (this may be an optional feature implemented in the T-NOVA Brokerage Module), thus, making this auction truthful. The T-NOVA Brokerage Module may change the pricing rule in a flexible manner. It is a matter of implementing an extra policy in the brokerage module operation mechanism. Additionally, the call price may be used to provide rational item valuation. The brokerage module will determine the proper call price for each VNF based on marketing factors. It is also possible that the bidders use their own valuation tools along with both the brokerage

module, so that the former (i.e. bidders) to be able to learn the optimum call price. Future work for auctioning implementation will be done in Task 6.3 – Brokerage module.

In summary, the brokerage module will provide the following functionalities:

- VNF discovery: this process is required in order the brokerage module to seek for the requested VNF.
- Trading: this process enable the brokerage module to trade the VNFs, especially through auctions, in case that one VNF is offered by more than one FP. Figure 2-7 depicts the sequence diagram of general auction trading.

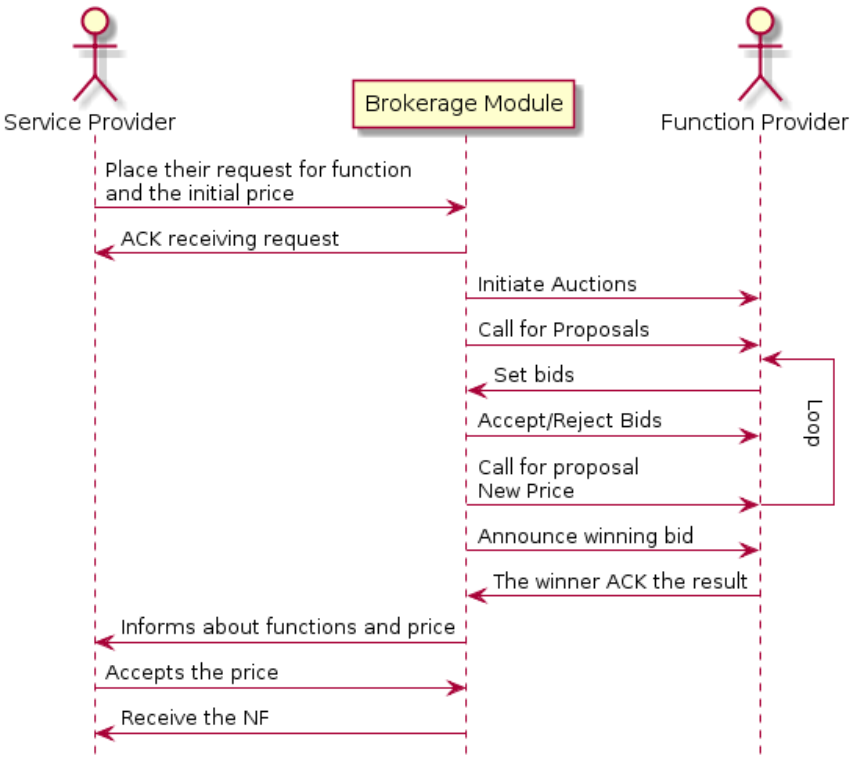

Figure 2-7 Trading process

1. The SP provides to the brokerage module the VNF request and the initial price.
2. The brokerage module sends an ACK that initiate auctions.
3. The brokerage module informs the FPs regarding the request and the initial price
4. FP sends their bids for the functions (Price + SLA specification)
5. The brokerage module solves an auction to maximize its revenue.
6. The brokerage module informs the bid results.
7. Depending on the type of auction, an iteration (3-6) continues until the bid winner is found.
8. The brokerage module announces the final results.
9. The winner acknowledges the results.
10. The brokerage module indicates the the VNF's price, which is provided by the FP that won the bidding, to the SP.
11. The SP accepts the price and SLA.
12. The SP receives the VNF.
13. (Price will be stored in the accounting module, and SLA agreement in the SLA management module).

Architecture

The overall architecture of the brokerage module and its interfaces is depicted in Figure 2-8.

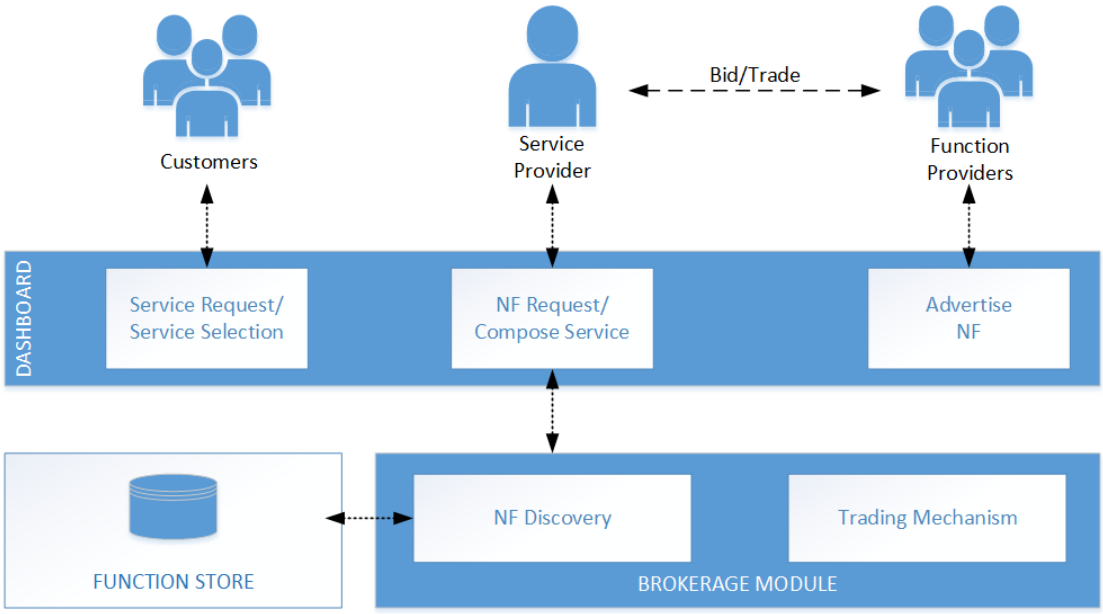

Figure 2-8 Brokerage module internal architecture

In case the customer would like to ask for a service that is not already in the catalogue, he will have the option to perform a request for a new service composition taking place. Therefore the Service Provider will use the brokerage module to query for specific VNFs. The process of trading between SP and FPs is then initiated according to the sequence diagram depicted in Figure 2-7.

Interfaces

The required interfaces of the brokerage module for the proper communication with the other parts of the T-NOVA system are:

- *Interface to the dashboard*: this interface is required in order for the users of T-NOVA system (i.e. SP, FPs) to be allowed to trade. For this purpose, functionalities such as service composition/VNF request by the SP and advertise VNF/trading by the FPs are exploited.
- *Interface to the service selection module*: with this interface the service selection module will check if there could have been a change in the price for the VNFs as a result of the trading process. Here the SS module is an intermediate component to finally provide that information to the accounting for billing purposes.
- *Interface to the SLA management module*: this interface is exploited in order for the brokerage module to provide information to the SLA management module regarding the SLA agreed between SP and FPs as a result of the trading process. (The SLA management module requires such information in order to create and store the SLA contract and for SLA monitoring issues).
- *Interface to the Function Store*: this interface is required in order for the brokerage module to retrieve information about the available VNFs for a service composition.

2.6.2.4. Business Service Catalogue

According to the system requirements gathered in Deliverable D2.1 [1], in order to a T-NOVA user (typically the customer) to be able to easily know the services already available in the T-NOVA system, and access the description of those services, the T-NOVA Marketplace will store all this information in what we have called the "business service catalogue", matching also with the approach suggested by TM Forum in its "integration framework", in which functional and non-functional aspects of a service based on service oriented principles are defined. This work will be continued in T-NOVA under Task 6.1 – service description.

Starting from the requirements in Annex A - Requirements Specification for this component, which are mainly related to the need of the catalogue to be browsable, including the service description, SLAs offered by the Service Providers and price for each services&SLA, we explain next its functionality and the way the information will be stored.

Functionality

The business service catalogue will be used by the Service Provider (SP), to store/create, services and update, delete the services. All stored services will be browsable based on criteria such as price, SLA and other service description characteristics, by Service Provider and Customer, which are defined in marketplace search view, through Dashboard module.

The business service catalogue will be filled with the service offering information manually and offline after a service composition has taken place by the SP through the orchestrator.

Design

The business service catalogue contains service offering entries, each of them is composed by: service description + SLA offer + price, according what

Figure 2-9 shows.

Figure 2-9 Business Service Catalogue

Interfaces

- Dashboard - the business service catalogue will be accessed directly and only by the dashboard module in both read and write mode (by the customer and SP respectively).
- Orchestrator – the business service catalogue will push the NSD relevant fields to the orchestrator when a new service offering has been created. Once the orchestrator validates it, the availability of a services is notified to the BSC to be offered to the customer.

2.6.2.5. Service Selection module

Based on the requirements gathered for the Marketplace and the initial implementation steps in order to improve in modularity it has been decided to include an intermediate component between dashboard, business service catalogue and orchestrator that will be responsible to activate in the orchestrator the selected network service by the customer: the Service Selection module.

Functionality

The Service Selection module will provide the T-NOVA customer the ability to activate a network service that matches his search criteria. It will communicate with orchestrator in order to activate the service, and depending on Orchestrator resources the selected network service will be provisioned, or it will be discarded. In case of successful activation, that would be notified by the Orchestrator, the Service Selection module will pass the network service information to the accounting module in order to be forwarded later to the billing module. It will not take the action over the accounting in case the service is not provisioned.

Before introducing the information in the accounting module it will make sure the information is updated and consult the brokerage module in case there has been any change during the trading process.

Design

Figure 2-10 depicts the overall architecture of the service-selection module.

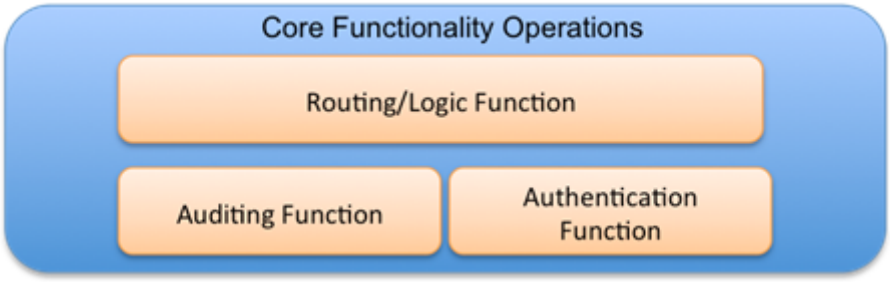

Figure 2-10 Service Selection Overall Architecture

The Routing/Logic Function is responsible to forward the request to orchestrator and depending on the answer, to forward it to accounting module.

The service module is based on modular architecture, in order to enhance the logic of the module, and to support additional interfaces with other modules if needed.

Interfaces

- Dashboard - Interface between the Customer and the service selection where the current service selected is being configured.
- Orchestrator - It is used to notify the orchestrator about a new NS instantiation and pass the custom service configuration.
- Accounting module – It is used to create the entries in the accounting module to track every service or VNF instance created in the orchestrator for billing purposes
- Brokerage module - This interface is used by the Service selection module to consult any change in the information regarding the price as a result of the trading process in the brokerage module before creating the accounting entry

2.6.2.6. SLA management module

Service Level Agreements (SLAs) represent a contractual relationship between a service consumer and a service provider in order to provide a mechanism to increase trust in

providers by encoding dependability commitments and ensuring the level of Quality of Service is maintained to an acceptable level.

In T-NOVA there will be a SLA agreed between Function Provider (FP) and Service Provider (SP) and between the Service Provider and its customers, and per each service, since the same service could have different SLA levels associated. One VNF can be offered by a FP with different flavours. Depeding on the technical characteristics of the virtual infrastructure in which the VNFs will be deployed, the performance achieved for the network service will be different. The performance garantees that a FP can offered for each VNF will be part of the negotiation through the trading mechanisms implemented by the brokerage module (see section 2.6.2.3.). Therefore one Network Service (NS) can be offered with different SLA levels and different prices, being part of different offerings as it is represented in Table 2-8:

SLA per service
service1, SLA11, price11
service1, SLA12, price12
service2, SLA21, price21
service2, SLA22, price22

Table 2-8 SLA per service

SLAs describe the service that is delivered, its properties and the obligations of each party involved. Moreover, SLAs establish that in case the guarantee is fulfilled or violated, rewards or penalties, monetary or not, can be applied, respectively. T-NOVA SLA management module will provide information for later accounting, depending on the terms and conditions gathered in the SLA and on whether this SLA has been met by all parties or not.

The requirements for the T-NOVA SLA management module, listed in Annex A, are mainly related to the need to provide mechanisms to get an agreement presented and agreed, store all the SLA agreements, to inform the orchestrator, and to know all the SLA fulfilment to inform the billing system for possible penalties.

Functionality

The SLA management module is in charge of providing mechanisms to get an agreement presented and agreed, informing the involved parties (Customer, SP, and FPs) and storing the SLAs, it will later receive and will process all measurements related to the SLA from the monitoring system (in the orchestrator) and, checking if the SLAs have been fulfilled or not, will inform the accounting system for the pertinent billable items (penalties or rewardings).

A SLA basically consists of two main steps:

1. Paper-signed contract, in this case, between the customer and the SP, and between the SP and FPs, including the description of the quality of service and the penalties to be applied (could also be on a web site by agreeing terms and conditions).
2. eContract: It is automatically negotiated between parties for each customer, depending on the demand. Always based on a paper-signed framework contract (step 1).

The SLA management module needs to be able to provide the following functionalities: publication, discovery and negotiation of SLAs requirements, in order to manage the SLA lifecycle that can be split in the following phases:

1. *SLA Template Specification*: for the SP (and FPs), a clear step-by-step procedure describing how to write an SLA template to provide a correct service description.
2. *Publication and Discovery*: publish the provider offer and possibility for the customer to browse/ compare offers.
3. *Negotiation*: agreement on SLA conditions between the customer and the SP and between the SP and the FPs. This could be a bargain-like transaction or simply a combo list selection of predefined choices when the customer selects a specific offering.
4. *Resource Selection*: depending on the chosen SLA for every service, the SP by means of the orchestrator will map that specification to the resources that need to be assigned to the service in order to meet this SLA.
5. *Monitoring and Evaluation of the SLA*: comparing all the terms of the signed SLA with the metrics provided by the monitoring system (from the orchestrator), in order to internally prevent upcoming violations.
6. *Accounting*: invoking the charging/billing system according to the result to inform about billable items as penalties or rewards.

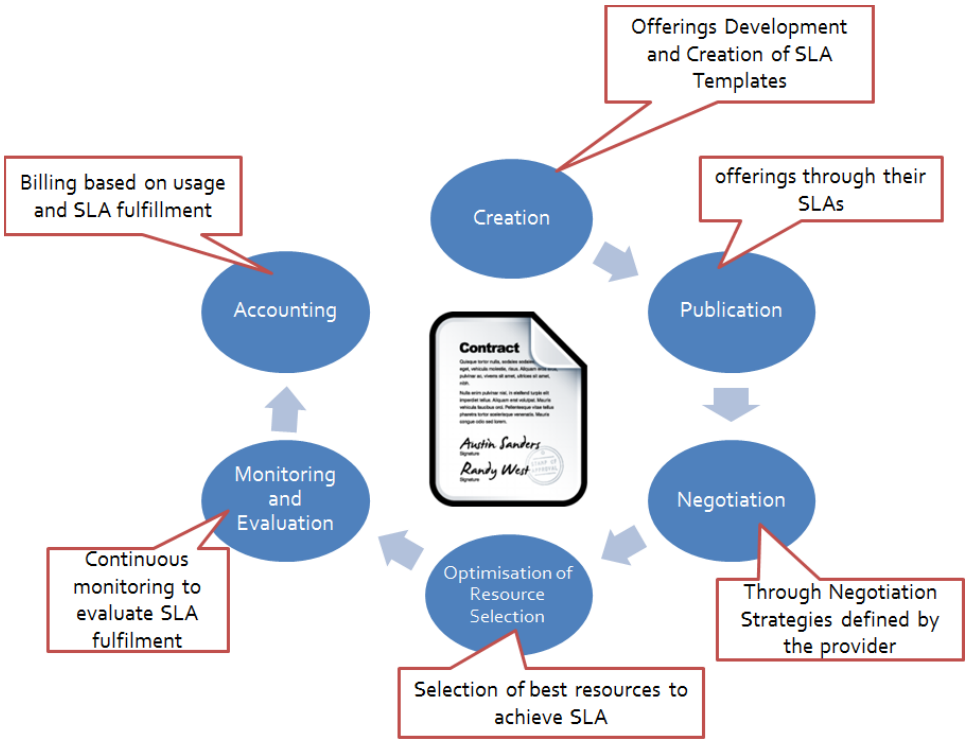

Figure 2-11 SLA lifecycle

Design

The information that shall be stored in the SLA management module is high level described in Table 2-9.

Item	**SLA template specification**	**SLA contract** *Parties involved* *Parameters* *Penalties*	**SLA fulfilment**	**Billable items**
	Input from the SP	Input from the	Input from the	Output of the SLA

	and FP from thedashboard	dashboard as the output of the SLA negotiation	the monitoring system in the orchestrator	management module (to be sent to the accounting module)
VNF ID				
Service ID				

Table 2-9 SLA management module information

Interfaces

So far, several interfaces are foreseen to ease the communication with the other parts of the T-NOVA system. These are:

- *Interface to the dashboard:* the SLA management module will provide an API to the dashboard to show the pertinent SLA information (template specification, agreement, SLA fulfilment, etc.) and to introduce the SLA templates.
- *Interface to the accounting system:* the SLA management module will be consulted by the accounting module about billable items as penalties or rewards when the SLA has not been achieved. Also, the Accounting Module will be in charge of introducing the final agreements once a purchase has occurred and to start/stop the SLA enforcement once a service has been provisioned or stopped.
- *Interface to the orchestrator:* The Orchestrator will get the information about the terms agreed on the SLA and generate the monitoring information for each metric wich will be consulted periodically by the SLA module to determine the level of fulfilment of the SLA for each service and function.

2.6.2.7. Accounting module

Functionality

The accounting module in T-NOVA will be in charge of registering all the business relationships and events (subscriptions, SLA evaluations and usage) that will be needed for billing. The accounting module will be the intermmediate component between the billing module and the rest of the system.

Design

The high-level information that shall be used by the accounting module is described in Table 2-10.

Type	It could be a Service or a standalone VNF.
Instance ID	ID of the instance of the service (or function) in the system once it's been instantiated. It's used for interactions with the Orchestrator.
Client	Purchaser of the Service (Customer), or Function (SP).
Provider	Seller of the Service (SP), or Function (FP).
SLA	Id of the SLA agreement of the transaction (in order to get possible penalties to be applied)

Status	Current status or the service (or function): Running or stopped,
Billing	Information on how to bill the Service (or function) to the client.
Dates	Date when the service (or function) was instantiated.

Table 2-10 Accounting module information

Interfaces

- *Interfaces to the SLA management module:* the SLA management module will be consulted from the accounting module about billable items as penalties or rewards when the SLA has not been achieved.
- *Interface to the service selection module*: the Service Selection module will be in charge of creating the entries in the Accounting module. While the Orchestrator is instantiating a service, the Service Selection stores in the Accounting module all the information necessary to track the newly created instances and link them to the SLA agreement and the pricing data for a more accurate later billing.
- *Interface to the orchestrator*: the monitoring system in the orchestrator will use this interface to update in the accounting system the status of the different services and functions.
- *Interface to the billing module*: by means of this interface the billing module will get all the information it will need to issue a bill.

2.6.2.8. Billing module

The billing module in T-NOVA is in charge of generate the bills for users and providers andrevenue sharing reports between the Service Provider (SP) and the Function Providers (FP) at the end of a billing cycle. In order to reused existing solutions, the billing module in T-NOVA will be an adapted version of the opensource rating-charging-billing framework Cyclops [35] .

Functionality

Cyclops allows data collection, normalization and persistence of resource consumption data for services consumed by the customers. The framework does not implement metering itself; it assumes that the resources that are being consumed are being measured. It provides a rich set of APIs for (non-natively supported) applications to report the consumption data to Cyclops. Natively it extracts the usage values (of resources) from a few supported IaaS and PaaS cloud platforms.

Cyclops allows custom set of meters to be defined and further allows providers to customize the rating and billing rules associated with such meters. Natively, it supports all built in meters for OpenStack and support for CloudStack is under development.

The framework generates usage data reports periodically, which has been extended to support events based usage reports generation for the T-NOVA billing scenarios. Charge data records are generated periodically also. This in T-NOVA is governed by the billing models, which are associated with service or VNFs instances belonging to customers.

Cyclops provides REST APIs for bill generation by specifying any desired period, hence the user of Cyclops must determine the end of billing cycle event and use the bill generation API from Cyclops.

All the data stored within the Cyclops framework have associated timestamps, thus justifying the data storage into time-series data store. It naturally supports various data analytics and has rich APIs for data visualization.

The framework has been extended to allow revenue share computation and report generation for monetary settlements between the SP and the FP taking into account the existing revenue sharing agreement between them.

Design

Cyclops framework is inspired by micro-services design approach for distributed application development. The key micro-services in Cyclops are:

a) Usage Data Record (UDR) mService,
b) Charge Data Record (CDR) mService,
c) Billing mService

The external applications send relevant data into Cyclops asynchronously over highly available message bus. The framework is integrated with Gatekeeper authentication and authorization service. The high level architecture is shown below:

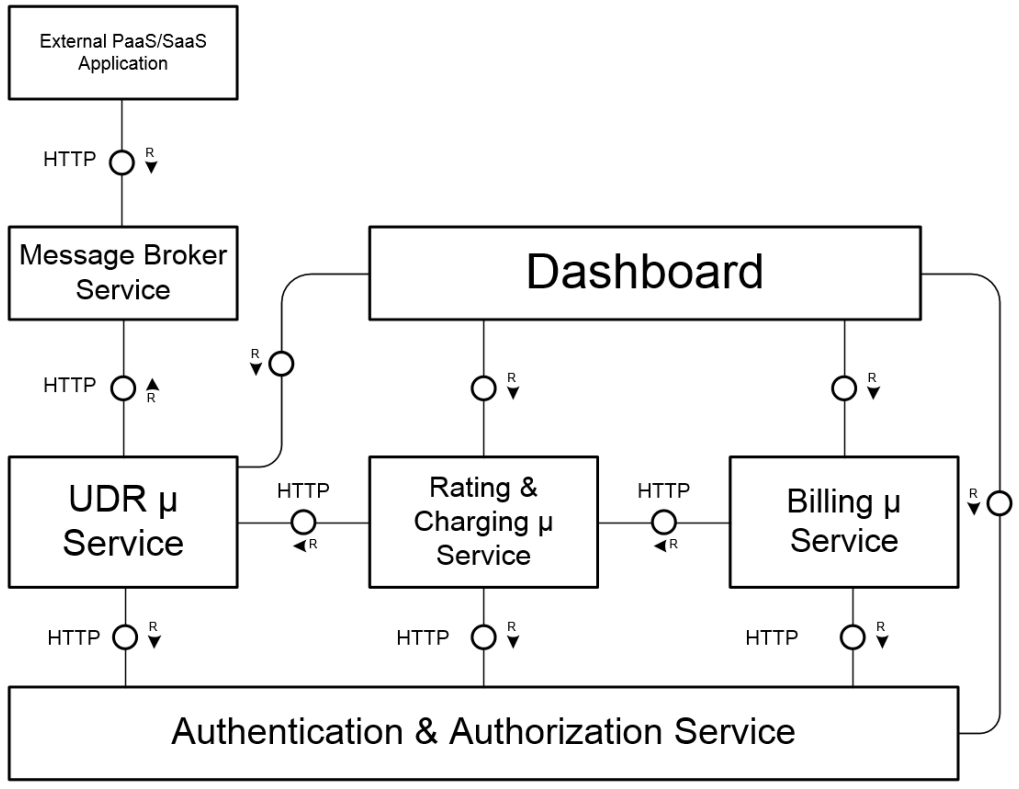

Figure 12: Cyclops micro-service Architecture representation

Interfaces

In T-NOVA, Cyclops has interfaces with the Accounting module, and the marketplace dashboard. The lists of various interfaces and the functionality that are (or could be) achieved over these interfaces are described below:

- Cyclops-UDR-Accounting - supports event based usage reports generation for active service instances
- Cyclops-CDR-Accounting – supports event based charge reports generation for active service instances, billing model details,

- Cyclops-billing-Accounting – information flow allowing generation of revenue sharing report between SP and FP.
- Cyclops-billing-SLA – SLA violations data query from billing micro-service for a given period
- Cyclops-billing-dashboard – bill generation for any desired period.
- Cyclops-UDR-dashboard – data extraction API could be used to provide rich visualization to customers
- Cyclops-messaging-Accounting – allows sending on billing relevant service lifecycle events to be sent to Cyclops

At a much higher level, these interfaces can be aggregated into these 3 listed below:

1. Cyclops-Accounting
2. Cyclops-Dashboard

3. NETWORK FUNCTION FRAMEWORK

This section is devoted to the definition and specification of the Network Function Framework that is the T-NOVA architectural element related to the structure and behaviour of the Virtual Network Functions (VNFs). This section describes also the Network Function Store (NFS) that contains the VNFs' (Virtual Network Functions) software images and their metadata description. This section provides also the high level description of VNF design. Finally, it describes the lifecycle of a VNF.

In summary the content of this section is:

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

We recognise ETSI NFV as a notable body working on VNF specification. In this document we will make several references to ETSI work. We will also compare T-NOVA with ETSI NFV with the aim of identifying synergies and gaps.

3.1. High level description

The Network Function Store (NF Store) contains all the information of the VNFs provided by different software developers or Function Providers (FPs) that want to offer them through the T-NOVA marketplace. The NF Store is maintained by the T-NOVA Operator.

The NF Store contains VNFs:

- provided by several (third-party) developers,
- published as independent entities,
- accompanied with the necessary metadata.

The diagram in figure 3-1 provides a very high level architectural description of the relationships of NF Store and VNFs with the other elements of T-NOVA architecture.

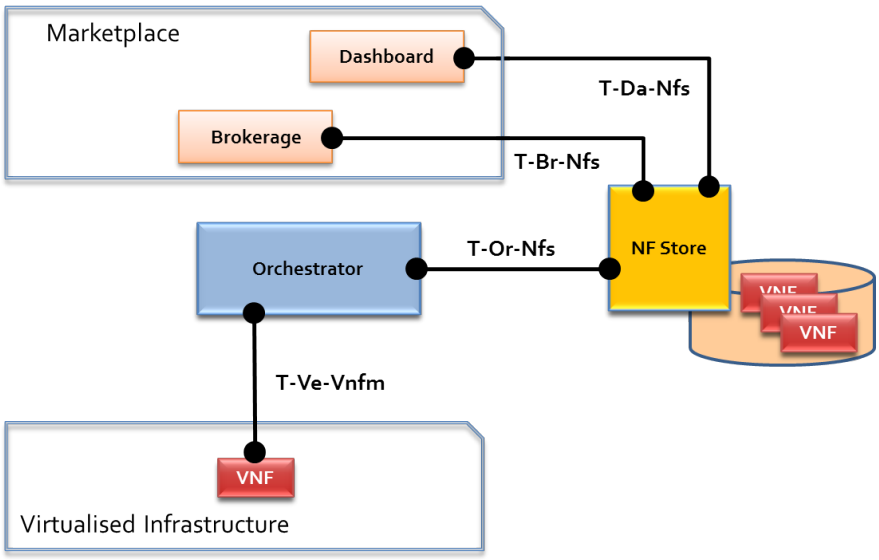

Figure 3-1. VNF framework high level architecture

The VNFs images reside in the NF store. They are selected via the marketplace dashboard and then instantiated and deployed over the execution platform (NFV Infrastructure). The T-NOVA orchestrator, specifically the NFV Manager, controls the execution of VNFs over the NFV Infrastructure.

The NF Store is mainly a repository for the VNFs and their metadata. A VNF can be versioned. Whenever a new VNF is added to the repository, the NF Store informs the orchestrator. Then, the orchestrator retrieves the VNF image and metadata from the NF Store. The same procedure applies for VNF removal, or for any other modification to a VNF in the NF Store.

According to this approach, the T-NOVA Orchestrator can be considered a single point of knowledge, meaning that the information shall always be easily and effectively accessible to the orchestrator.

The VNF Manager controls VNFs through the T-Ve-Vnfm interface. The operations over this interface are related to the VNF lifecycle, which must be supported by each VNF to be compliant with T-NOVA system. During the active phase of a VNF, the monitoring system provides information to the VNF Manager about the status of the virtualized infrastructure resources (e.g. remaining cores in the VNF Infrastructure), some generic metrics about the sate of each Virtual Machine running each VNF (e.g. % CPU, used RAM) and finally some application-specific indicators directly provided by the VNFs (e.g. concurrent sessions). This information allows the Orchestrator to take decisions about resource allocation and initiate scaling procedures if needed. This detailed architecture is represented in figure 3-2 diagram.

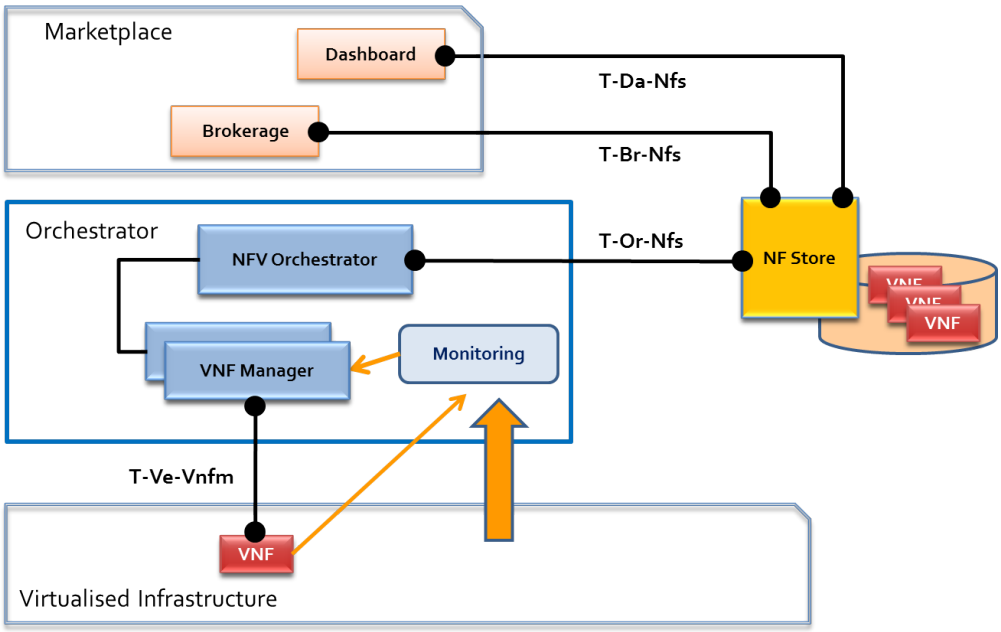

Figure 3-2. VNF framework detailed architecture

3.2. NF Common Components

This introductory chapter describes the components and behaviours of T-NOVA VNFs. All the topics will be deeply analysed in dedicated sections. The specific details for implementation will be described in other deliverables of technical workpackages (from WP3 to WP6).

3.2.1. NF structure and properties

Generally speaking, a Virtual Network Function (VNF) is an executable software program that implements the whole or part of some network functions and is hosted in a virtualised platform. VNFs can be deployed in a network solution between computer machines (virtualised or not) and physical networking devices.

Given the quick convergence of computing, storage and networks and the very high volume of standard servers that are shipped, the idea is to exploit the flexibility and agility of software implementations in comparison with hardware proprietary boxes. In this way, the launch of new telecom services in short time is relatively easy. Most of custom hardware appliances (e.g., network processors, digital signal processors, firewalls, etc.) can be implemented as VNFs and moved where and when needed.

In the scope of T-NOVA project, a VNF is a virtual machine or a group of virtual machines that implements Network Functions in software. The VNF high level architectural model is represented in figure 3-3. A VNF is characterised by two attributes: the operational functionalities and the management behavior. 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.).

Starting from the operational part, a VNF application may support one or more application network interfaces for communicating with other VNFs at the application layer level. Incoming and outcoming data traffic (after being processed by the VNF) are passed through Virtual Network Interfaces (VNIs).

On the other hand, two specific interfaces exist for management purposes. On the one hand, a VNF management socket interacts with Virtualised Infrastructure Manager(s) -VIM(s) - and

on the other hand, a management interface is used to communicate with the T-NOVA orchestrator framework (see T-NOVA deliverable D2.21 [3]for more details).

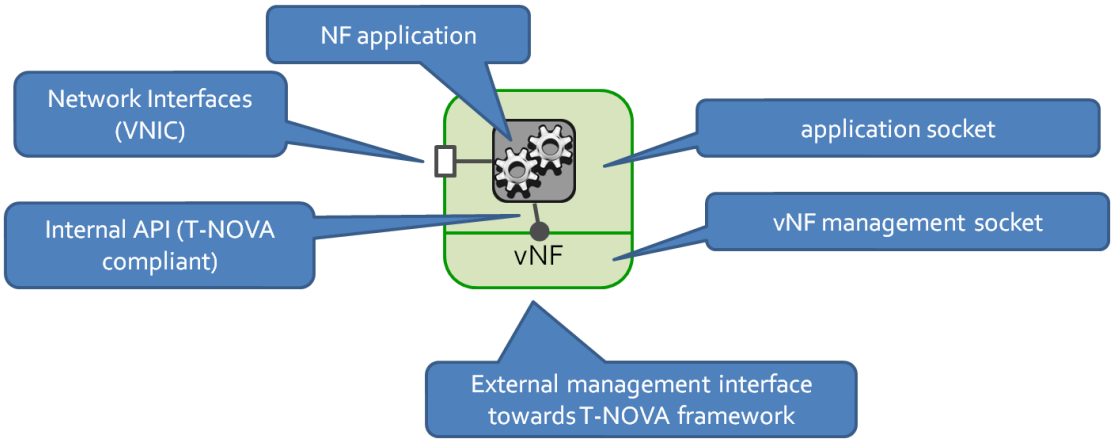

Figure 3-3. VNF high-level structure

Even if internals of a VNF are left to the VNF Developer, we suggest adopting a common structure for VNF components depicted in figure 3-4.

The "VNF Controller" is the internal component devoted to the support of the VNF lifecycle. The "Init Configuration" component is responsible for the initialization of the VNF that happens at the beginning of the VNF execution. The Monitoring Agent component transmits application-level monitoring data towards the Monitoring System.

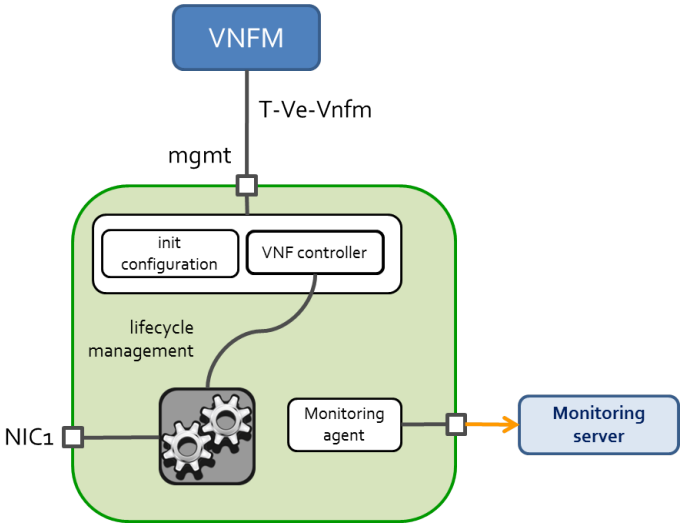

Figure 3-4. VNF internal components

All of the abovementioned VNF internal components are optional, except the VNF Controller which must be present in each VNF. To be more precise, the VNF Controller will interface the VNF manager to support the lifecycle management operations within the VNF. More details about the allocation of the VNF internal componets require to introduce the concept of Network Function Component as explained in the following section.

3.2.1.1. VNF composition

More that one Virtual Machine can be necessary to realize a single VNF. In this case, each VM is called a Virtual Network Function Component (VNFC).

The diagram in figure 3-5 represents a VNF composed by two VNFCs, one of them exposing one network interface, while the other being internal to the VNF.

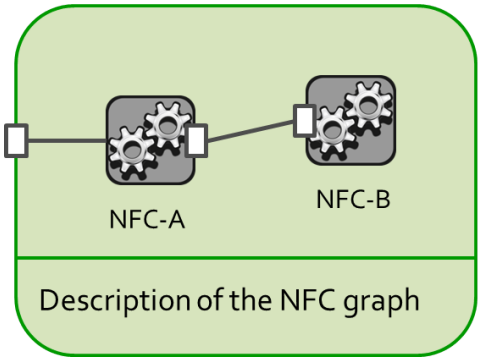

Figure 3-5. VNF composition

Both Operational (specific to each VNF) and Management behaviours (ie. Lifecycle events) are described in VNF Descriptors and VNFC Descriptors files. For example, a VNF Descriptor imposes the minimal VNF infrastructure resource requirements that a VNF instance needs. Moreover, VNF Descriptors provide details about the instance internal topology and lifecycle operations.

In addition, the VNF provides the description of the group of VNFC that composes it.

The only component of the VNF directly interacting with the Orchestrator is the VNF Controller through the V-Ve-Vnfm interface.

The VNF and VNFC properties, contained in corresponding descriptors, will be exposed through the VNF Management socket.

In the case of a composed VNF, the internal VNF components required by T-NOVA can be allocated in many different ways. The minimal mandatory requirement is that each VNF must have only one VNF controller. The other components, i.e. Init Configuration and Monitoring Agent, can be optionally allocated in the different VNFCs. In figure 3-6 we provide an example of a VNF composed by two VNFCs. In this case the Init Configuration component is allocated in all the VNFCs, while there is only one Monitoring Agent. Of course other configurations are possible depending on the particularities of the VNF.

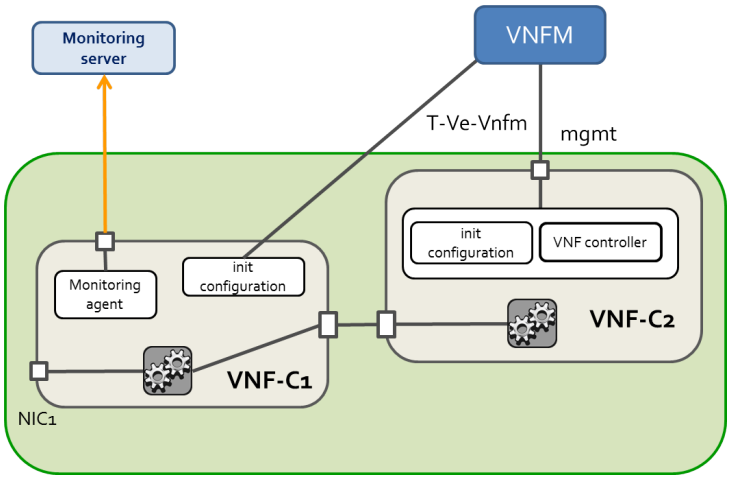

Figure 3-6. VNF internal components in multi VNFCs VNF

In T-NOVA, all relevant information is expressed using a metadata language. Since metadata permits the autonomous operation of each VNF and VNFC, direct interaction between the T-NOVA Orchestrator and VNFCs located inside of VNFs has been considered. The interaction with the marketplace is indirect through metadata descriptors.

Service graph or forwarding graph

Service Chaining can be used to build End-To-End service. They can be built by composing several VNF, thanks to the metadata used in the descriptors. The VNF Forwarding Graph depicts this interconnection of VNFs and the traffic flows between them. More specifically, several VNFs can be connected to provide a service, and these connections can be visualised through a graph. The T-NOVA Orchestrator can use the VNF descriptors for creating this Service Graph (or forwarding graph) by interacting with the VNICs of the VNFs as presented in figure 3-7.

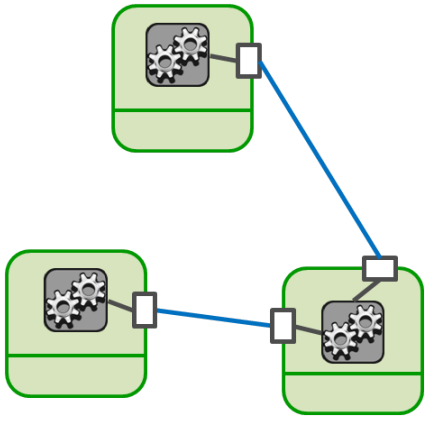

Figure 3-7. VNF Service Graph

3.2.2. Metadata in T-NOVA

The VNF metadata descriptor (VNFD) provides the information for describing what the VNF functionalities are and how to manage them. Many approaches are currently available on how to implement this template, namely:

- Amazon AWS CloudFormation [36]
- OpenStack Heat Orchestration Template (HOT) [37]
- ETSI VNF Descriptor Template [7]

They provide the technical information for configuring, running and operating the VNF, i.e. the virtual machines with the software application images providing services in a cloud environment.

In T-NOVA, we will extend this information with business-relevant aspects that allow the registration and trading of a VNF in the marketplace.

The VNF Descriptors in T-NOVA will include:

- the VNF name and the version;
- the vendor of the VNF;
- the VNF behaviour (i.e., the description of what the VNF does);
- an SLA description offer;
- a price offer;
- the reference to the software image associated with the VNF;

- the minimum required hardware resources (e.g. CPU unit numbers, virtual memory size, virtual disk size);
- the network interfaces (i.e., the connection points with the network);
- selected performance metrics (i.e. how VNF shall perform in typical configurations);
- the default deployment parameters;
- the definition of some statistics (i.e. monitoring data generated during the active state of the VNF);
- information about how to properly scale the VNF (i.e., scaling in procedures);
- possible functional or network constraints;

From the moment that all metadata of a VNF will be available to the T-NOVA platform, the marketplace can use operational and business information for selecting the most suitable VNF according to customer's needs and willingness to pay. At the same time, the Orchestrator will exploit the metadata for managing the VNF according to the VNF lifecycle.

Besides this high level approach, an in-depth analysis about metadata structure and template creation have been made in the technical workpackages WP3, WP4, WP5, and WP6. We decided to publish the in-depth analysis directly in T-NOVA public web site (http://www.t-nova.eu). A practical reason is that the architectural definition and the specifications in this document remain vaild while the research and definition of the VNF decriptor will continue after the final issue of this document, therefore we would have risked to publish incomplete or inconsistent information. Most important is the ambition to encourage the adoption of T-NOVA framework by a large number of VNF developers and this information is fundamental for them for developing their VNFs.

3.2.3. T-NOVA NF Framework and ETSI NFV comparison

This paragraph makes a comparison between the Virtual Network Function (VNF) framework that will be used in T-NOVA and the one specified by ETSI.

In ETSI NFV specifications, the objective of NFV is to separate software that defines the network function (the VNF) from hardware and software used to create generic hosting functions infrastructure which executes the VNF (the NFVI: Network Function Virtualization Infrastructure). Therefore, the requirement is that VNF and NFVI are separated, as depicted in Figure 3-8.

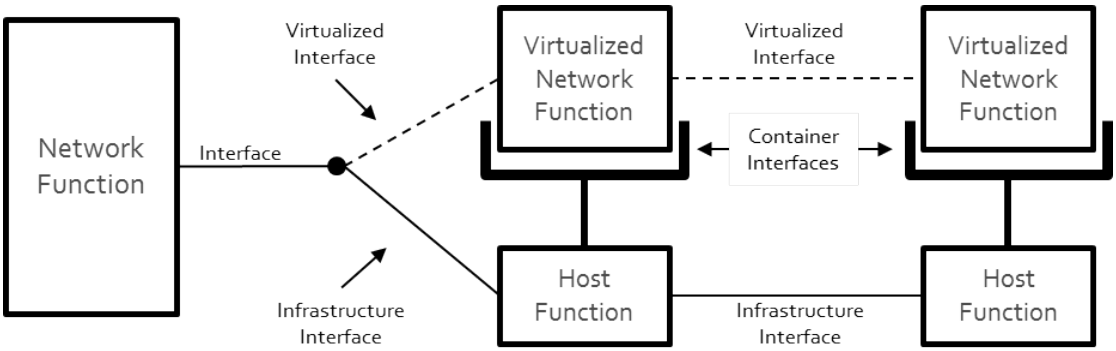

Figure 3-8. Virtualisation of network functions in ETSI

As a result:

- the Network Function functional block is split into a host function plus a Virtualized Network Function (VNF);
- a container interface between the VNF and the host is created;

- the NF Interface is split into virtualized and infrastructure interfaces;
- the interfaces between non-virtualized and virtualized network functions are homogenous.

An important concept must be taken into account: the VNF cannot be considered a functional block independent of its host function. The VNF cannot exist autonomously, since it is an abstract view of the host function when configured by VNF.

The NFV architecture is thus not defined by functional blocks but using the following entities:

- Host Functions and their associated container and infrastructure interfaces;
- VNFs with their associated container and virtualized interfaces.

Regarding management and orchestration, the ETSI framework is split into management and orchestration of infrastructure (NFVI) and network functions (VNFs), as depicted in figure 3-9. A similar approach for management and orchestration has been adopted in T-NOVA, as described in [3] and [2].

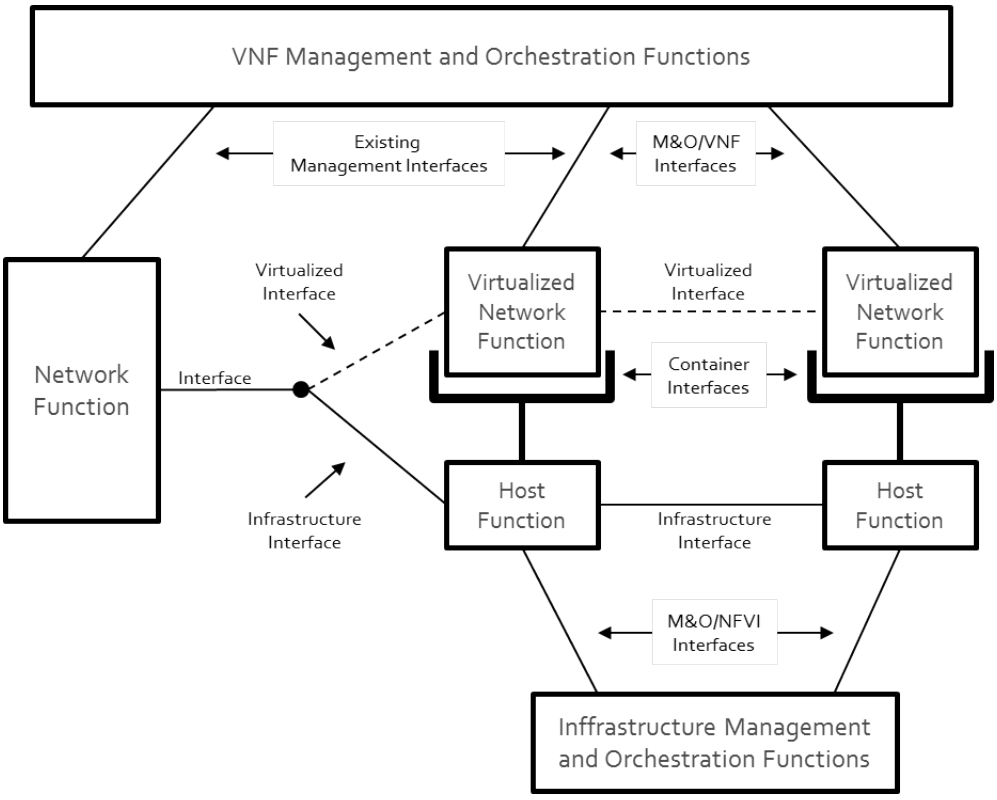

Figure 3-9. Management and orchestration of NFVs in ETSI

With particular regard to the structure of the VNF, which is the focus of this section of the document, ETSI specifies a number of interfaces, which are relevant to the VNF software architecture, as depicted in figure 3-10.

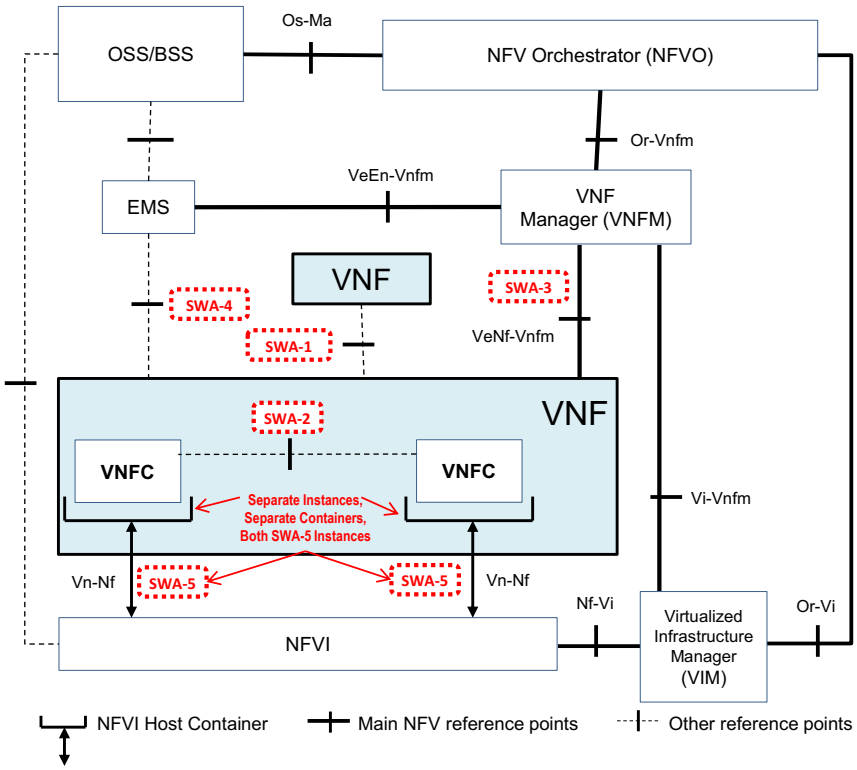

Figure 3-10. NFV architectural framework and interfaces in ETSI

In the following list, a brief explanation of these interfaces is provided.

- SWA-1: this is a well-defined interface used to connect various network functions in a forwarding graph.
- SWA-2: it refers to VNF internal interfaces, i.e. for VNFC to VNFC communication. In fact, a VNF can be decomposed and made up from sub-parts or components (VNFCs) which are themselves VNFs interconnected by the infrastructure.
- SWA-3: it interfaces the VNF with the NFV management and orchestration, specifically with the VNFM.
- SWA-4: it is used by the EMS to communicate with a VNF.
- SWA-5: it corresponds to VNF-NFVI container interfaces.

In T-NOVA, the VNF structure has been completely described in paragraph 3.2.1. Figure 3-11 shows the T-NOVA VNF high level structure and the mapping with the VNF's interfaces specified by ETSI.

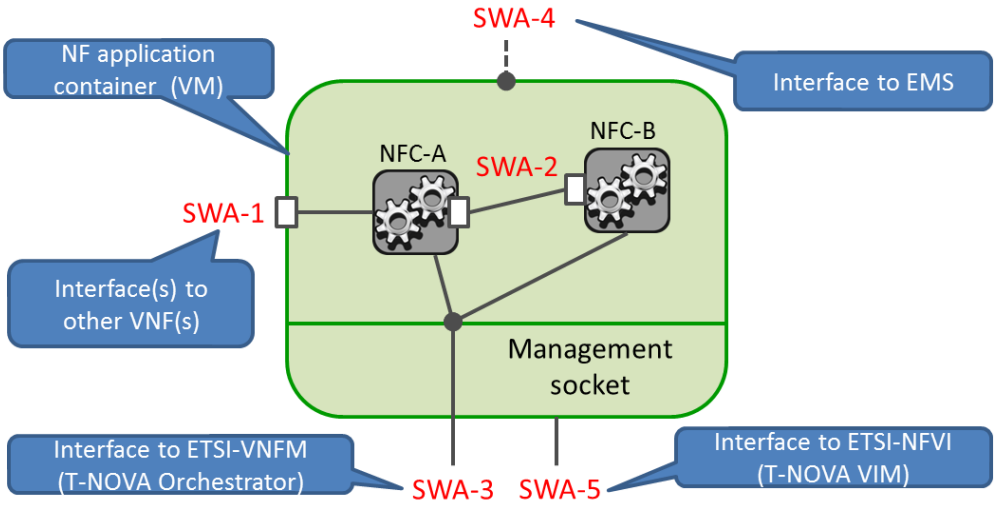

Figure 3-11. T-NOVA NFV structure mapping to ETSI framework

It is worth noting that ETSI interfaces SWA-1 and SWA-2 correspond to interfaces used in T-NOVA for communication between VNFs and VNFCs, respectively. Regarding management, the T-NOVA framework is aligned with ETSI SWA-3 and SWA-5 as well, by adopting a couple of interfaces for the control of VNFs from the T-NOVA Orchestrator and the control of infrastructure by the T-NOVA VIM. The interface SWA-4 with the Element Management System (EMS) is not further specified by ETSI because it is considered already present in current network architectures. For the same reason in T-NOVA we consider this interface as available.

Besides the comparison of the ETSI and T-NOVA VNF frameworks, a brief explanation of common patterns in VNF design and operations defined by ETSI and the corresponding support of relevant features in T-NOVA is provided in Table 3-1.

ETSI VNF feature		*Description*	**T-NOVA support**
VNF design patterns	VNF Internal Structure	VNF(C) composition is possible	Supported.
	VNF Instantiation	Each VNFC of a VNF is either parallelizable or non-parallelizable.	Supported.
	VNFC States	Each VNFC of a VNF may need to handle state information.	Supported.
	VNF Load Balancing Models	There are different types of load balancing, typically 4 models are identified (internal, external, e2e and infrastructure).	Ready to support it. Not studied in the deliverable.
	VNF Scaling Models	Auto, on-demand and management based scaling.	Driven by T-NOVA Orchestrator.
	VNF Component Re-	Different models have	Component re-use

ETSI VNF feature		*Description*	**T-NOVA support**
	Use	been studied regarding component re-use but only one case has been agreed as relevant for ETSI NFV.	is possible and encouraged. Not studied in details because out-of scope for T-NOVA.
VNF Update and Upgrade	VNF Update and Upgrade Overview	The key difference is that VNF upgrades may require planning on network service level, while VNF updates can be deployed without coordination with other VNFs participating in the same VNFFG.	Update and upgrade require creating new service instance using updated VNFs versions available in the Function Store.
	VNF Update & Upgrade Requirements for VNF Provider	The VNF package shall provide an automatic procedure for upgrade/update of the VNF instance. The procedure shall support control of the progress of this process, including the allocation of virtual resources from the NFV management and orchestration. Roll-back will be supported as well.	Same as above.
VNF's Properties	Hardware Independence	HW dependent, partly COTS, COTS.	Only COTS considered until now.
	Virtualization and Container Awareness	Hypervisor (agnostic, dependent), OS containers, HL container tech, no virtualized & no container, partly virtualized.	Hypervisor agnostic.
	Elasticity	No, any, in/out only, up/down only	Only in/out.
	Scaling Automation	No, VNF-triggered, triggered by the NFV management and orchestration functions.	Triggered by T-NOVA Orchestrator.
	VNF Policy Management	Full policy, not policy based. This feature is related to the ability of a VNF to support rule-	Full policy is supported in T-NOVA metadata.

ETSI VNF feature		*Description*	**T-NOVA support**
		based provisioning needed for automated tasks (e.g. scale-in, scale-out, or migration based on threshold crossing).	
	Migration operations	No live, Live, Partial, other.	Not addressed yet.
	VNF State	Stateful, stateless.	Supported.
	VNF Internal Structure	Simple, structured.	Supported.
	Reliability	Dependent on fault detection and fault reporting mechanism.	Reliability is up to each VNF. It is provided at application level.
	Location Awareness	Dependencies on position.	Location independent.
	Application Management	No, proprietary, standard, multiple service providers.	Supported.
	Diversity and Evolution of VNF Properties	In a VNF with multiple VNFCs, each VNFC may have different properties.	Supported.
Attributes describing VNF's Requirements	VNF Topological Characteristics	The deployment configuration and operational behaviour of a VNF shall be described according to a template called Virtualized Network Function Description (VNFD).	T-NOVA metadata.

Table 3-1 T-NOVA support of ETSI VNF features

Finally, the data model used to describe VNFs in T-NOVA is considered.

In ETSI, the Virtualized Network Function Description (VNFD) is a specification template provided by the VNF Provider for describing virtual resource requirements of a VNF. It is used by the NFV management and orchestration functions to determine how to execute VNF lifecycle operations (e.g. instantiation, etc.). VFND is fully described in [7].

The template captures the general characteristics of each VNF and is used to on-board the VNF, in order to support on-demand instantiations of the VNF in an operator's network.

The deployment configuration and operational behaviour of a VNF shall be described according to the same VFND template. The deployment configuration defines the state and environment for a VNF to be deployed, whereas the operational behaviour defines the needed functions for a VNF to be operated and managed properly [6].

The VNFD is composed of the following main information elements groupings:

• VNF identification data, including:

- Data to uniquely identify the VNF vendor/provider.
- Type and description of the VNF, which help to identify the Network Function that is implemented as a Virtual NF, and enable interoperability of VNFs manufactured by different VNF Providers.
- Version.

• VNF specific data, including:

- Specific VNF configuration data.
- Connectivity requirements and inter-dependencies of VNFCs.
- VNF lifecycle workflow scripts.
- Deployment flavours, specifying how many instances of each VNFC type and version to be instantiated based on service KPIs.
- Deployment constraints.

• VNFC data, including:

- Type and identification, uniquely identifying the VNFC type.
- Specific VNFC configuration data and scripts.
- Deployment constraints.
- Virtual container files/images references, including the possibility to define packages of: VNFC binaries plus operating system, empty operating system, and/or empty virtual container (i.e., unloaded operating system).

• Virtualized resource requirements, including:

- Compute resources, e.g. virtual CPU and virtual RAM assigned to the virtual container.
- Storage resources, e.g. size of the virtual disk assigned to the virtual container.
- Network resources, e.g. number and type of virtual NICs that are assigned to the virtual container, including also requirements for network bandwidth.

Besides all this information, the metadata used in T-NOVA (section 3.2.2) includes some additional information for the marketplace, which is used to build and maintain an ecosystem for an easy and efficient brokering of VNFs among different stakeholders in the market (FPs, SPs and Customers).

3.3. NF Lifecycle

The VNF lifecycle encompasses the different stages that a VNF needs to pass through. In the T-NOVA project, we have defined the following stages:

- Development:
  - Software implementation of Network Functions (NFs) is performed by Function Providers. NFs are published and aggregated in the T-NOVA Function Store.
- Validation:
  - The validation procedure aims at providing a certification that the developed NFs will work as expected.
- Publication:

- NF publication is performed at the NF Store, whose repository will host both the function image (as stand-alone application or integrated VM) and the associated description/metadata

- Brokerage:
  - Brokerage is undertaken by the brokerage module in the marketplace that will match users' service requirements with the technical capabilities provided by the NFs, thus ensuring that the resources required for NFs deployment are available.
- Selection:
  - Finally the customer selects the most suitable NFs according to his or her needs.
- Deployment:
  - The VM image and its metadata are transferred from the NF Store
- Management:
  - This is the running phase of the NF. An Application Programming Interface (API) will be exposed to the ochestrator for setup and real-time control or management. The running phase can be preceded by a network re-configuration procedure.
- Termination:
  - Involves the removal of the NF instance from the virtualised infrastructure, including network re-configuration, if needed.

The VNF lifecycle is represented in figure 3-12.

Figure 3-12. VNF lifecycle

Actual interaction between T-NOVA Orchestrator and VNF happens in management and termination states of the VNF lifecycle, as highlighted in figure 3-12. This consideration brings to the definition of the extended VNF lifecycle presented in figure 3-14. It highlights the status diagram internally to the management status.

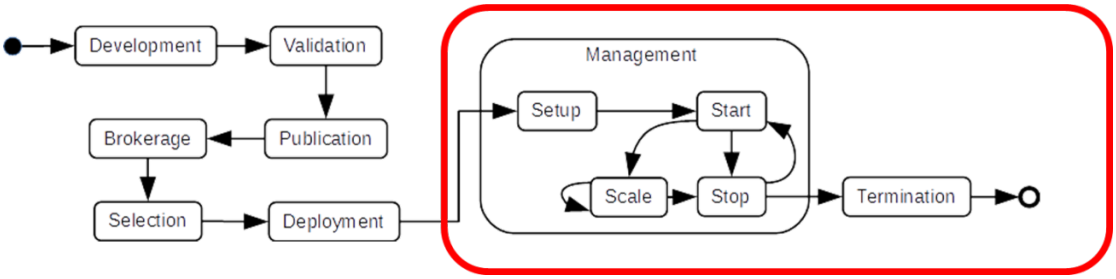

Figure 3-13. Extended VNF lifecycle

The VNF lifecycle is implemented by events generated by the VNFM towards the VNF Controller Function in each VNF (see: figure 3-2).

The remaining of this chapter is devoted to the detailed descrition of this VNF lifecycle.

3.3.1. Development

The development covers the implementation phase of the VNF.

The VNF shall be uploaded on the Function Store using the Marketplace dashboard. The NF application will run in the execution environment under the coordination of T-NOVA Orchestrator.

The VNF developer shall also provide the metadata description of the VNF. The description includes both functional and non-functional information that will be used by different elements of T-NOVA framework.

T-NOVA does not intend to use a specific programming language nor to impose a given SDK. The result of the VNF development process consists of one or more virtual machine images and the related metadata files. The information model is described in a section 3.2.2 of this document and published in the T-NOVA public web site (http://www.t-nova.eu). As for the implementation details, they will be addressed in the WPs dedicated to VNF implementation.

3.3.2. Validation

The validation process ensures that the VNF:

- supports T-NOVA T-Ve-Vnfm API,
- operates as expected,
- performs as expected.

In T-NOVA it is required that only validated VNFs can be uploaded to the NF Store.

VNF validation is performed off-line and prior to any upload. This process will in particular generate some information that will be reflected in the VNF metadata.

Implementing automated tools for VNF validation is out-of-scope of T-NOVA. Nevertheless, T-NOVA will provide guidelines and describe validation best practices based on the experience acquired from the effective implementation of VNFs in the T-NOVA framework.

The validation process shall include executing a number of test suites. The minimal requirement is to verify that the VNF supports the T-NOVA API for the whole VNF lifecycle. This goal can be effectively reached by using the "TDD kit for VNF developers". This is a Test Driven Development environment a VNF develper can download from the T-NOVA public web site. They can use it for validating their VNF with respect to the VNF lifecyle. The TDD kit consists of a VM composed by a test generation engine and test suites. These test suites are run against the VNF under development that actually is another VM.

3.3.3. Publication

The publication process consists of registering the VNF and its metadata into the NF Store. Only validated VNFs can be uploaded to the NF Store. The interaction with the NF Store occurs through a user interface supported by the T-NOVA FP dashboard.

This stage of the VNF lifecycle needs to be further detailed to address the updating of a VNF or also its withdrawal from the NF Store.

The publication stage in the VNF lifecycle is therefore composed by:

- Publication,
- Modification,
- Withdrawal.

of a VNF image and its metadata.

In all the cases, the NF Store informs the T-NOVA Orchestrator about any modification affecting a VNF in the NF Store. In case of publication or modification of the VNF, the orchestrator gets the updated VNF image and metadata from the NF Store. In case of withdrawal the orchestrator simply removes the VNF from its internal database.

The diagram in figure 3-14 describes the publication procedure. The same workflow applies to the modification of the VNF by adapting the function calls accordingly.

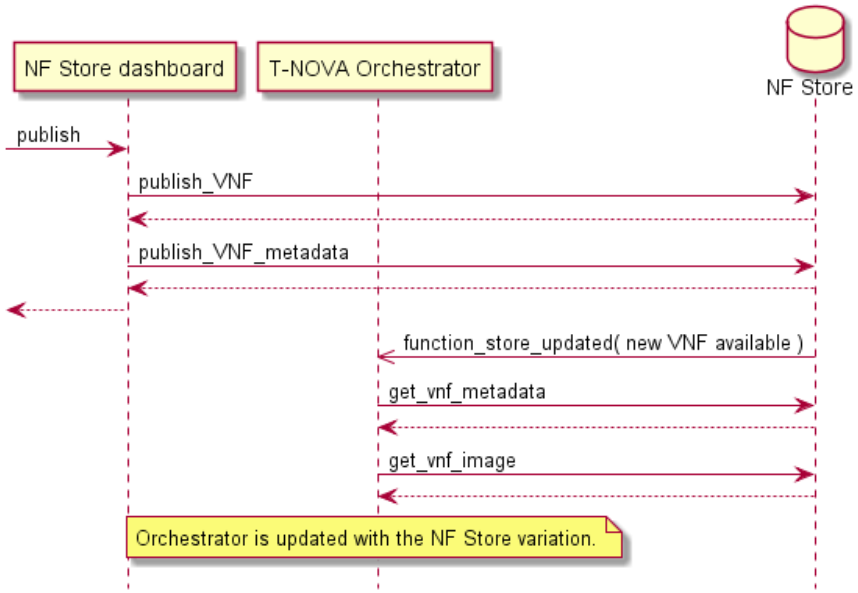

Figure 3-14. VNF publication in the NF Store

The workflow for withdrawing the VNF from the NF Store is slightly different. It is represented in figure 3-15.

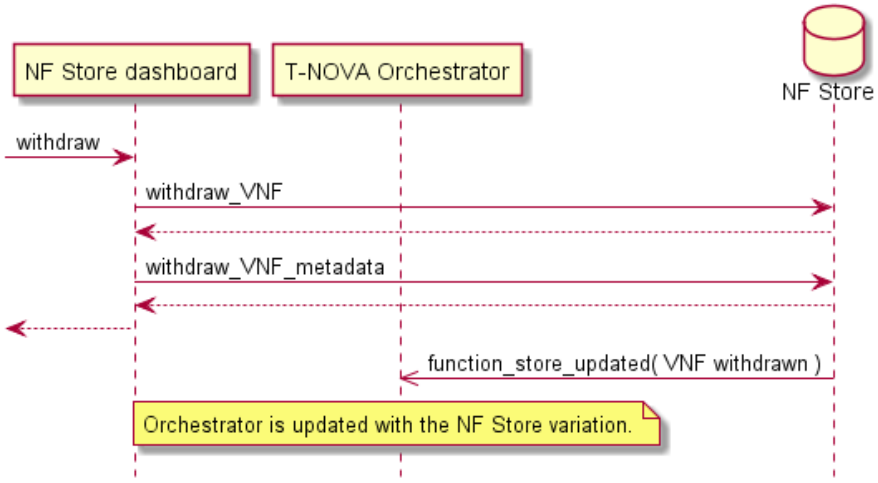

Figure 3-15. VNF withdrawal from the NF Store

3.3.4. Brokerage and Selection

These operations are performed by the marketplace. The metadata of the VNFs is accessed by the brokerage module in the marketplace to perform trading among Function Providers offering VNF with similar functionality. The final goal is to find the best price for the Service Provider, considering the VNFs description and offered SLA (see section 2.6.2.3.).

3.3.5. Deployment

Deploying a VNF means installing and initializing the VNF in the NFV infrastructure. Then the VNF is ready to be activated.

The deployment of a VNF consists of transferring the VNF VM image(s) containing the VNF from the NF Store to the NFV infrastructure.

The deployment phase requires interaction between T-NOVA Orchestrator and the VIM.

The workflow is fully described in Deliverable D2.31 related to Orchestration and Virtual Infrastructure specification [2].

3.3.6. Management

The management stage is dedicated to real-time management of the running VNF.

This is composed by a set of sub-stages such as:

- set-up
- start
- stop
- scaling
- monitoring

The management phase is controlled by the T-NOVA Orchestrator that directly interacts with the VNF and the VIM. There is also an indirect interaction between the VNF and the Virtual Infrastructure because of the execution of the virtual machines composing the VNF. This interaction is common to all the VMs in a cloud environment.

The VNF lifecycles events are described in a specific metadata file, the VNF Descriptor (VNFD), and are implemented as a set of operations or scripts the VNFM activates for configuring and controlling the VNF.

During the VNF's provisioning stage the VNFM interacts with the Init Configuration component that can be allocated in each VNFC composing the VNF, while the VNF lifecycle requires VNFM interacting with the VNF Controller that is present in only one VNFC of the VNF.

3.3.6.1. Set-up

The VNF set-up consists of the initialization phase of the VNF. The orchestrator asks the VIM to start the VNF. Then it can directly interact with the Init Configuration component of the VNF, for instance to configure the interfaces (e.g. NICs). This interaction happens through the T-NOVA API supported by the VNF. The bootstrap initialization results from an indirect interaction with the NFVI. The NIC configuration is a direct interaction with the orchestrator. The latter uses the VNF metadata information for learning about the VNF IP interfaces. Moreover, it needs to know the service description for configuring the service graph or forwarding graph in the right way. If needed, the ochestrator can also run network configuration tasks. In this case the T-NOVA Orchestrator asks the VIM to reconfigure the network.

The set-up phase in figure 3-16 represents a case of a VNF composed by a single VM.

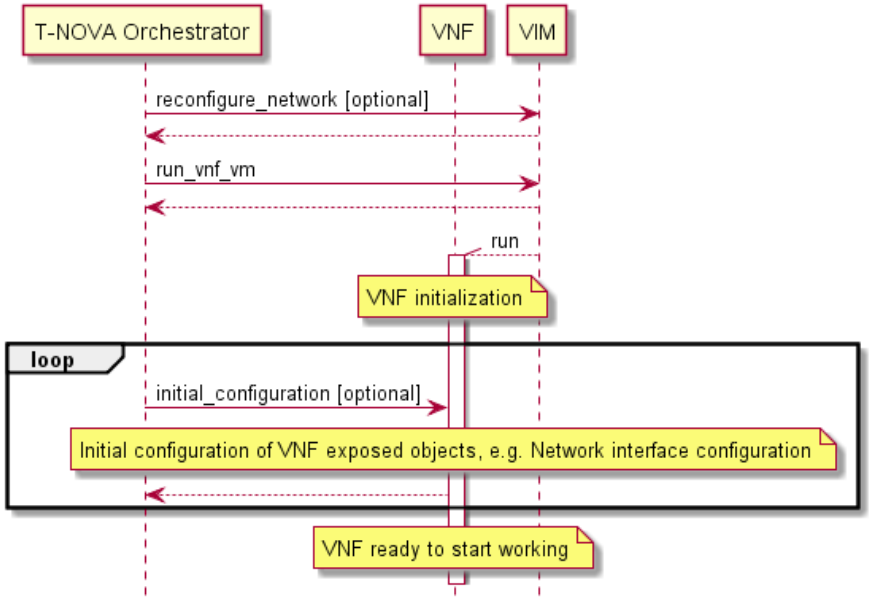

Figure 3-16. VNF set-up

In case the VNF is composed by several VNF Components the orchestrator shall repeat the procedure for all the VMs. In this case the NIC configuration could require more complex interactions with the VNF.

3.3.6.2. Start

The start command instructs the VNF to start providing its services.

The start phase in figure 3-17 represents a case of a VNF composed by a single VM.

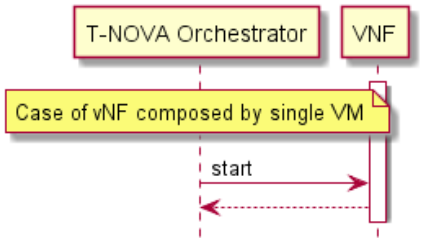

Figure 3-17. VNF start

3.3.6.3. Stop

The stop command instructs the VNF to stop providing its services. After a stop command, the VNF can receive a new start command. Otherwise, the lifecycle evolves to the termination stage.

We can distinguish between graceful and immediate stop. Graceful stop is a request to the VNF to stop accepting new service session request. However, active service sessions continue to be supported until their natural termination. Immediate stop request forces the termination of all active service session.

The stop phase in figure 3-18 represents a case of a VNF composed by a single VM.

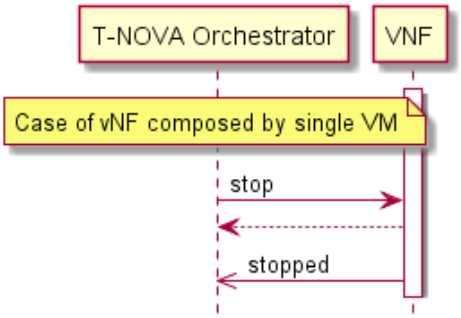

Figure 3-18. VNF stop

3.3.6.4. Scaling

Scaling exploits the elasticity property of a service based on a cloud environment. In other words, it is the ability to dynamically increase or reduce the amount of resources used for providing the service. According to ETSI NFV, scaling is the ability to dynamically extend or reduce resources granted to a VNF [38].

Scaling policies are known as scaling out/in, and scaling up/down. Scaling out/in means allocating or releasing resource instances, i.e. creating new VMs or terminating VMs. Scaling-out adds, while scaling-in removes resource instances. Scaling up/down means increasing or decreasing the resources allocated to an instance, i.e. increasing or decreasing memory or computation power to a VM. Scaling-up increases, while scaling-down decreases resources.

Scaling out/in must be supported by the VNF. It consists of accomodate or release a VNF component instance inside of the VNF. Whenever the orchestrator decides for scaling out/in, it triggers the VNFM that in turns ask the VNF to add or remove a VNF-C instance. Moreover the orchestrator is resposible to ask the VIM to allocate or remove the VM resources for the VNF-C involved in scaling.

Implementing scaling up/down needs additional features not widely available in present operating systems and hypervisors. This scaling policy consists in modifying the hardware configuration of a machine while it is running. Both the hypervisor and the operating system of the virtual machine shall support live adaptation of computation, memory, storage, and network resources. Moreover, these operations shall be coordinated between the hypervisor and the operating system. Also for this scaling policy it is up to the orchestrator to initiate and manage the process. In this case it shall require dedicated VIM capability while it seems that the VNF application is not particularly involved in the process. T-NOVA does not support scale up/down.

Independently from the chosen policy, scaling can be the strategy adopted by a Service Provider (SP) to match customer's expectation of an optimally performing system. More technically, scaling might be a follow-up of monitoring activity on SLA fulfillment. Monitoring data are collected from the system monitoring platform (in the orchestrator) and also from the VNF itself. Monitoring data will be compared against SLA threshold provided by the SLA management module in the marketplace. Whenever, some threshold is overcome, the orchestrator shall perform the most appropriate scaling procedure according with the description contained in the VNF metadata. In case the SLA is not fulfilled by scaling procedures, this would be taken into account by the SLA management module to take the appropriate actions tha will impact in billing (section 2.6.2.6.).

The diagram in figure 3-19 gives an example of scaling-out procedure. As soon as the demand for resources increases a new VNF-C instance is activated and configured for collaborating in efficiently providing the service.

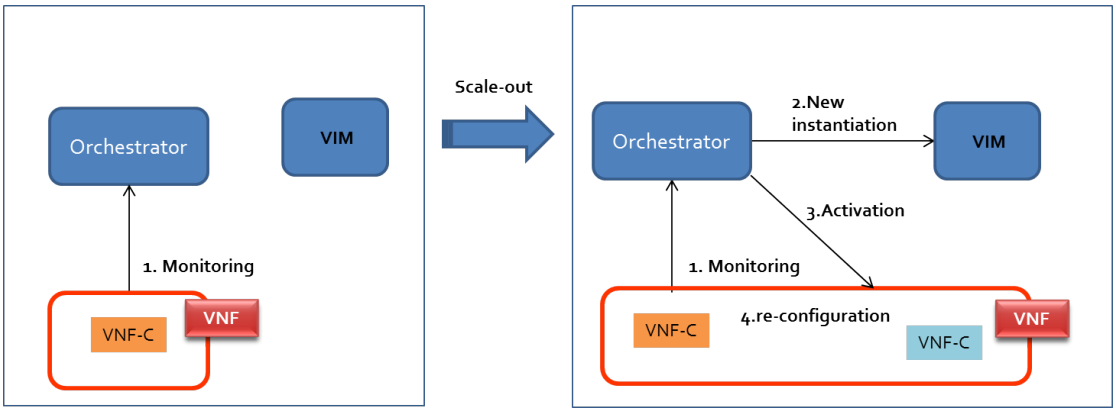

Figure 3-19. Scaling-out example

The VNF developer shall include in the metadata descriptor VNFD a section dedicated on scaling out/in. This information will be used by the orchestrator for activating the appropriate procedure when the need of scaling is detected.

More detailed description of scaling-out is provided in figure 3-20. When the orchestrator detects the need of scaling, it interacts with the VIM for allocating new VM and with the VNFM for activating the scaling procedure towards the VNF. The VNF receives the scaling command and the references of the new VNF-C and activates the most appropriate

procedure for accomodating the new VNF-C. In general it consists in the reconfiguration of a load balancing or scheduling funcion. For shake of clarity the detailed internal interaction between the orchestrator and the VNFM is not described here. Details are in [39].

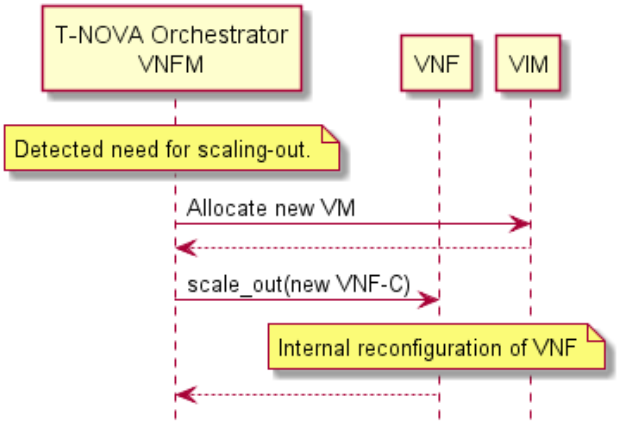

Figure 3-20. Scaling-out

The procedure for scaling-in is similar taking attention that now the VNF-C instance shall be terminated and released. Scaling-out is described in figure 3-21.

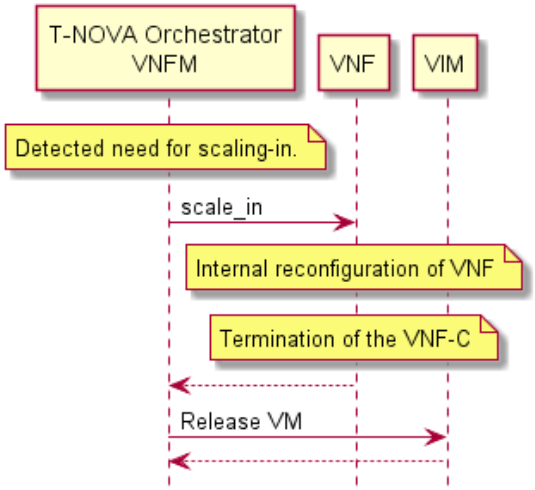

Figure 3-21. Scaling-in

We recongize that for many VNFs an internal front-end, scheduler, or load balancer function is needed in order to support scaling. However, it is not wise to generalize this practice imposing it in the design of any VNF.

3.3.6.5. Monitoring

The provision of an effective monitoring for checking the availability and the health status of the VNFs is mandatory. This will help in better understanding how the VNFs are using the resources, and whether the resources are correctly used. In T-NOVA, the monitoring mechanism will collect and display different types of data including memory, virtual storage and virtual network resource usage. To be more specific, the monitoring mechanism is part of the orchestrator and will be in charge of monitoring on the one hand the infrastructure through the VIM and on the other hand the VNFs that are part of the service. Here, the Function Provider (FP) needs to provide what to monitor at the VNF level, and the Service Provider (SP) needs to perform the same task at the composed network service.

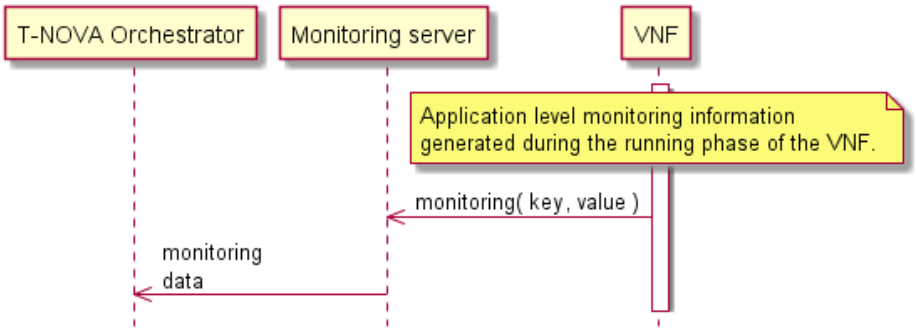

Figure 3-22. VNF monitoring

In Figure 3-22 the KPIs defined by the Function and Service providers as well as their corresponding values that are generated during the running phase of the VNFs are sent to the monitoring server that in turns delivers monitored information to the orchestrator. The latter will match the received information against the "rules" that are defined through the VNF metadata or through the SLAs. Based on this comparison, an action such as activating a scaling procedure might be taken. The KPIs that can be used for the monitoring mechanism include network related data (packet loss, bandwidth, Jitter, CPU threshold, etc.) and data related to the service provided by the VNF (in case of SBC could be number a SIP messages in a given time slot, number of concurrent active sessions, etc.).

3.3.7. Termination

This is the final phase of the VNF lifecycle that occurs at the end of the provisioning of the service implemented by the VNF.

Termination phase consists in the removal of VNF instance from the virtualized infrastructure. It can also require a network re-configuration.

The termination phase in figure 3-23 depicts a case of a VNF composed by a single VM.

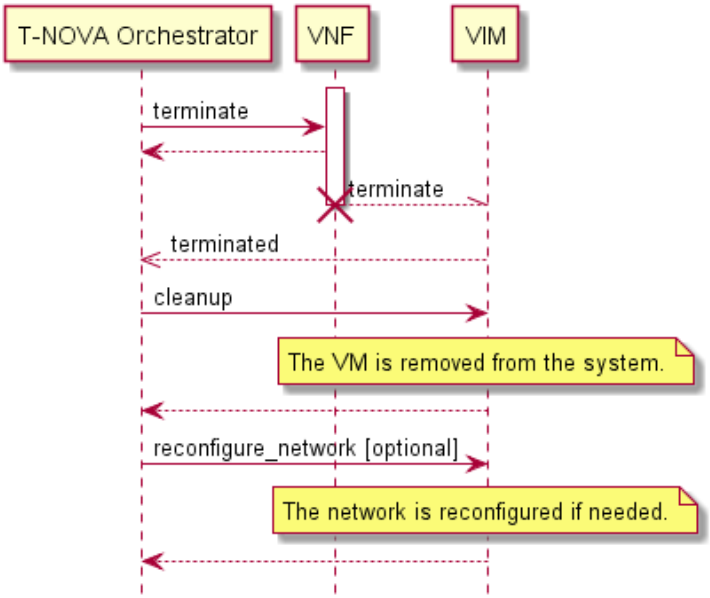

Figure 3-23. VNF termination or clean-up

In case the VNF is composed by more VMs or also more VNFCs the orchestrator shall repeat the procedure for all the VMs. In this case the NIC configuration could require more complex interactions with the VNF.

3.3.8. T-NOVA vs ETSI NFV lifecycle

In this paragraph, a comparison of the lifecycle defined in T-NOVA project with respect to the lifecycle defined in ETSI NFV will be carried out. The focus will be on the lifecycle management functions that are required to manage the instantiation, maintenance and termination of a VNF (or NS).

In ETSI NFV, the state transition diagram for a VNF is depicted in Figure 3-24.

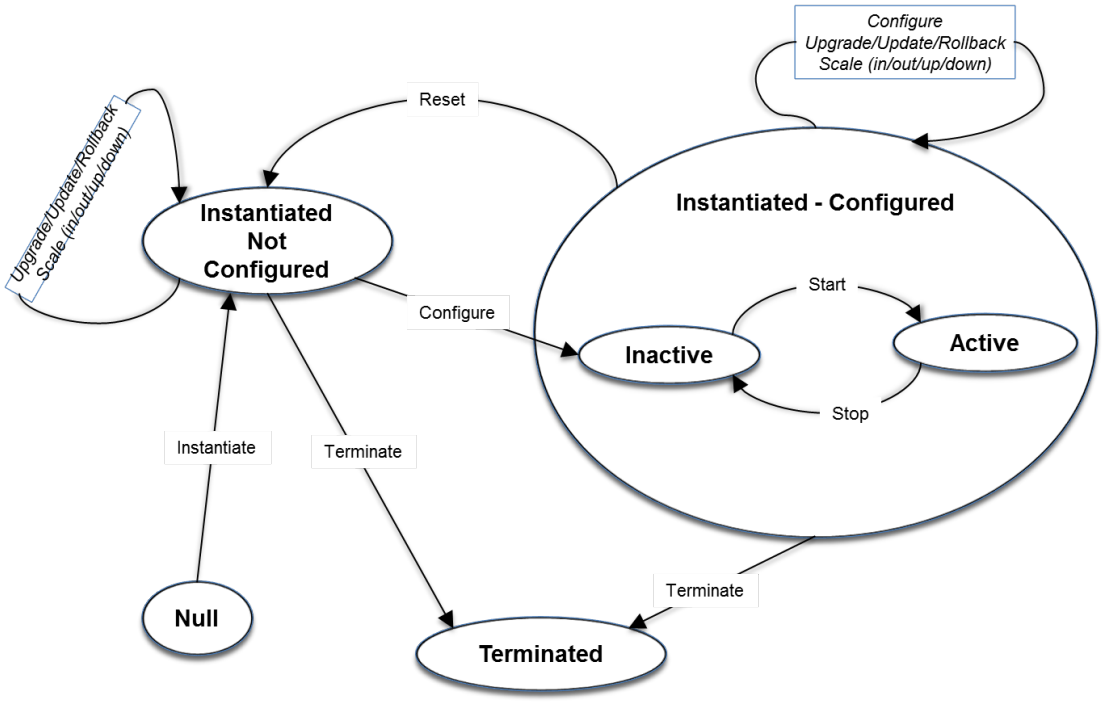

Figure 3-24. VNF instance state transitions

According to the diagram, a VNF can be found in the following states:

- Null: VNF Instance does not exist and is about to be created.
- Instantiated Not Configured: VNF Instance does exist but is not configured for service.
- Instantiated Configured – Inactive: VNF Instance is configured for service.
- Instantiated Configured – Active: VNF Instance that participates in service.
- Terminated: VNF Instance has ceased to exist.

Making a comparison between T-NOVA and ETSI VNF lifecycle is not a trivial task since the ETSI specification documents are still in draft status. The ETSI reference documents used take carry out this analysis are:

- Network Functions Virtualisation (NFV) - Virtual Network Functions Architecture [6] that describes the software architecture of the NFV framework;
- Network Functions Virtualisation Management and Orchestration [7] that describes the functions collectively provided by NFVO, VNFM, and VIM.

The VNF lifecycle is controlled by a set of operations necessary for a VNF to provide its expected functionality. The prerequisite for all lifecycle operations is the VNF on-boarding (process of registering the VNF with the NFVO and uploading the VNF data (VNFD, SW images etc).

In Table 3-2, an overview is given of T-NOVA lifecycle operations and how they map to (or differ from) their corresponding ETSI lifecycle operations.

Lifecycle operation		ETSI Reference
T-NOVA	**ETSI NFV**	
Development In T-NOVA, development is the process to build an NF application compatible with the T-NOVA environment.	- Development is not explicitly described in the ETSI framework.	-
Validation In T-NOVA, validation is an off-line process that has to be made prior to any VNF upload in the NF Store.	Validation In ETSI, a validation step is required during on-boarding by the NFVO.	[7].5.4.1
Publication In T-NOVA, a method is exposed by the T-NOVA Orchestrator for notification change(s) in the NF Store. After notification, the T-NOVA Orchestrator requests the new data.	On-boarding VNF on-boarding refers to the process of submitting a VNF Package to the NFV Orchestrator to be included in the catalogue. In ETSI, the sender will submit directly the VNF Package data to the Orchestrator.	VNF Package management [7].7.2.1 [7].B2.1
Modification	Update	
Withdrawal	Delete	
- - - Managed by T-NOVA Orchestrator.	Disable Enable Query	VNF Package management [7].7.2.1
Brokerage Operation performed in the Marketplace.	- Not in ETSI scope.	-
Selection Operation performed in the Marketplace.	- Not in ETSI scope.	-
Deployment	Instantiate Instantiate will include configuration if specified by the VNF deployment template. In ETSI, a feasibility check is (optionally) performed before instantiation.	[7].5.4.2 VNF lifecycle management [7].7.2.4 [7].B.3.1.2
Management::Set-up	Configure	VNF configuration

Lifecycle operation		ETSI Reference
T-NOVA	**ETSI NFV**	
In T-NOVA, Get, Create and Set operations will be supported.	Get In ETSI, the Configure interface will include the following operations: Get, Create, Set, Delete and Notify.	[7].7.2.6
Management::Start	Start	[6].6
Management::Stop	Stop	[6].6
Management::Scaling In T-NOVA, only in/out scaling will be supported. Sources of scaling will be either the VNF or the monitoring platform itself and the operation will be performed by the T-NOVA Orchestrator.	Scale In ETSI, scaling options are: in/out, up/down. Several sources of scaling have been considered (MANO.B4) that can be reduced to two categories: (a) automated scaling, where decision is done by or forwarded to the VNF Manager and (b) scaling based on management request, where the scaling request is coming from some sender (OSS/BSS or operator) and received by the NFV Orchestrator.	VNF lifecycle management [7].7.2.4 [6].6
Not supported in T-Nova.	In ETSI, the VNF lifecycle management interface will include the following oprations: Query Check Heal Update Modify Upgrade Reset	VNF lifecycle management [7].7.2.4 [6].6
Management::Monitoring In T-NOVA, a VNF monitoring method is exposed by the T-NOVA Orchestrator.	In ETSI, monitoring is related to VNF performance and VNF fault management. VNF information exchange is based on a notify/get mechanism.	VNF performance management [7].7.2.7 VNF fault management [7].7.2.8
Termination	Terminate	VNF lifecycle management [7].7.2.4 [7].B.6

Table 3-2 Comparison of T-NOVA with ETSI NFV lifecycle operations

In conclusion, 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.

Due to the fact that neither the specifications in ETSI nor the works in T-NOVA project are consolidated yet, some more investigation on this subject could be required in the future.

3.4. Network Function Framework APIs

The VNF framework described in this section interacts with the T-NOVA system through the interfaces represented in figure 3-1. The interfaces are exposed by the NF Store and also directly by the VNFs.

The NF Store exposes interfaces for interacting with the database of the VNF made available to the system.

Interface T-Da-Nfs is used by the dashboard for interacting with the NF Store for publishing, modifying, and withdrawing a VNF and its related metadata.

Interface T-Or-Nfs connects the NF Store with the orchestrator for providing information about the VNF metadata description and its executable image making the VNF available to the T-NOVA system.

Interface T-Br-Nfs is used by the Marketplace brokerage module to retrieve information about the VNFs available in the NF Store.

Interface T-Ve-Vnfm is exposed by each VNF to the orchestrator. During the active phase of a VNF, the interface T-Ve-Vnfm allows the orchestrator to directly interact with the VNF. This interaction takes place in the “Management” and “Termination” states of the VNF lifecycle. In particular, the whole active phase of a VNF (set-up, start/stop, scaling, monitoring) is controlled by means of direct communication between orchestrator and VNF. Such communication is performed through calls to the exposed functions of the VNFs API.

3.4.1. APIs high level logical description

For each interface exposed by the Network Function Framework we provide the description of the high level logical primitives or API functions. In-depth analysis and detailed specification of the API will be developed in workpackage WP3 and WP5.

3.4.2. Network Function Store APIs

These APIs allows interacting with the NF Store for managing the uploading, downloading, and deletion of the VNF image and the related metadata descriptor.

Interface T-Da-Nfs contains the following operations:

- publish_VNF, publish_VNF_metadata
  - Upload in the NF Store the VNF image and the VNF metadata descriptor.
- withdraw_VNF, withdraw_VNF_metadata
  - Delete from the NF Store the VNF image and the VNF metadata descriptor.

Interface T-Or-Nfs contains the following operations:

- get_VNF_metadata, get_VNF_image
  - Read from the NF Store the VNF image and the VNF metadata descriptor.

Interface T-Br-Nfs contains the following operations:

- get_VNF_metadata
  - Read from the NF Store the VNF metadata descriptor.

The complete definition of the operations over T-Da-Nfs, T-OR-Nfs is described in the technical specifications [40], [41].

3.4.3. VNF API

The API described in this chapter allows the VNF to be managed by the orchestrator in T-NOVA system. Any VNF must implement this API to be part of T-NOVA.

Interface T-Ve_Vnfm contains the following operations:

- initial_configuration
  - Provides the initial configuration parameters to each VNF-C composing the VNF. Typical usage is assigning the IP address of the VNF's network interfaces.
- start
  - Asks the VNF to start providing its services.
- stop
  - Asks the VNF to stop providing its services. Two types of stop are supported: hard and soft. A soft stop ensures a graceful cease of the VNF. In particular, asks to complete any in-progress operation before stopping the VNF. The hard stop asks to immediately cease all the operations of the VNF. The VNF remains active after the stop operation is completed, i.e. the MV is up and running; only the service provisioning is interrupted.
- terminate
  - Asks the VNF to terminate its execution. This operation shuts-down the VMs composing the VNF.
- scale_out, scale_in
  - Ask the VNF to perform scaling out/in procedures by reconfiguring the internal resources composing the VNF.

3.5. Network Function Store design

The NF Store is mainly a repository for the VNFs and their metadata. High level breakup of NF Store includes:

- NF repository
  - Archive of VNF images and related metadata. This data is versioned so that different versions of the same VNF can be present in the NF Store.
- NFS manager
  - Provides interfaces for interacting with the NF repository.

The NF repository is responsible for concurrent operations that are performed on the store (CRUD operations) on images and metadata.

The NFS Manager block implements the interfaces to the repositories. In addition, it delegates some traits of its responsability (like authentication and authorization access) to an external AA module. Security mechanisms can be adopted for data exchange over these interfaces.

The NF Store provides interfaces to the orchestrator (T-Or-Nfs) and marketplace's Dashboard (T-Da-Nfs) and Brokerage (T-Br-Nfs) as described in Figure 1-1. An additional interface (not shown in the picture) for console management is foreseen.

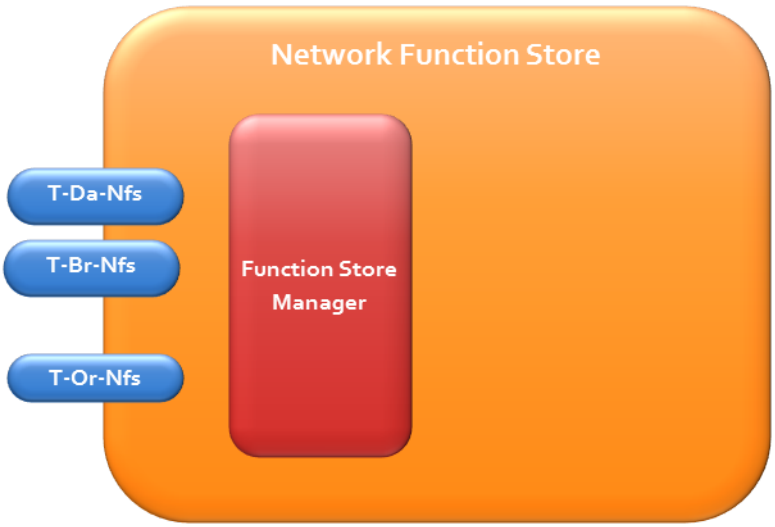

Figure 3-25. NF Store architecture

The requirements capture process for the Network Function Store has been performed according to the procedure explained in section 2.5 for the marketplace components. They are listed in Annex 6.1.2.

4. INTERACTION BETWEEN NETWORK FUNCTION FRAMEWORK AND THE MARKETPLACE

In this section we aim to summarize and to make clearer the relation between the Network Function Framework and the T-NOVA Marketplace according to what have been described in the previous sections.

Firstly, the way of purchasing VNFs it is summarized in section 4.1, and the interfaces between the NF Store and the marketplace are explained in section 4.2.

4.1. Purchase of VNFs

The purchase of a VNF through the T-NOVA marketplace should go through the following steps:

VNF storage

The T-NOVA Marketplace will advertise only validated Network Functions (NFs). After being tested and validated, the NFs will be uploaded into the NF Store. To prevent misuse of the NF Store, the Function Provider (FP) needs to authenticate itself and be authorized to upload the VNF. The same authentication procedure enables the FP to upload and upgraded version of its VNF. In addition to the VNF, the corresponding metadata information has to be made available.

VNF advertisement

The marketplace will advertise the VNFs stored in the NF Store based on the metadata that accompanies them and their availability provided by the orchestrator. In case a new version of the NF under consideration is uploaded, the NF Store will inform the orchestrator about this and consequently, the VNF advertisement on the marketplace will be updated.

VNF purchase

A customer willing to purchase a service from T-NOVA will need look for available services by selecting available VNFs or Network Services (NSs), and setting the start and end day and time of the service. The customer might also need to insert the information related to the ingress and egress routers that will form the end points of the path that will carry on the service traffic. The customer will also need to specify the SLA requirements he is interested in.

Once the customer selects a service, it will need afterwards to configure the related parameters. This will vary from one VNF to another. The parameters might include (content type, protocols to be used, traffic to be allowed or denied, prioritization rules if needed).

After the configuration, the service is ready to run, however, the customer needs first of all to select the means by which it will be charged.

4.2. Marketplace – Function Store interfaces

There are two interfaces between the marketplace and the Function Store:

1. T-Da-Nfs: The interface between the dashboard and the NF Store despicted in Figure 2-4 has the single functionality of providing the API for the Function Providers to insert their VNFs offerings and associated metadata in the Function Store.

Once the VNFs metadata are stored in the NF Store, the available VNFs to compose a service will be ready for the brokerage module by means of the orchestrator, which will update the availability in the Function Store. This is required to avoid that the marketplace could allow the SP to create a service starting from VNFs that although there are available at market level, they could be still not technically ready to be part of a service yet.

2. T-Br-Nfs: The brokerage module will use it to retrieve information about the available VNFs.

5. CONCLUSIONS

5.1. Summary

This document provides the specification and high level design of the Network Function Framework and T-NOVA Marketplace for later implementation. Analysing the state of the art, including main standardization activities, previous research projects and commercial solutions it has been gathered that at this stage a proper marketplace to deliver VNFs as a Service together with the Function Store as T-NOVA proposes does not exist.

The T-NOVA Marketplace has been designed to be used by three kinds of stakeholders according to the use cases analysis performed in T-NOVA [1], therefore a three views dashboard will be implemented as well as and access control module that will provide AA functionalities to control their different permissions. The T-NOVA Marketplace will allow Virtual Network Functions (VNFs) provided by a variety of software developers (Function Providers) to be published and traded by means of a brokerage module that will implement pricing mechanisms e.g. auctioning, when a new Network Service (NS) is going to be composed by a Service Provider (SP). T-NOVA Customers will be able to browse and select among the available network service offerings in the marketplace by means of a business service catalogue as well as negotiate the associated SLA and price. The billing procedure contemplates not only final customers of T-NOVA network services, but also the commercial relationship between the Service Provider and Function Providers.

In relation to the specification of the Network Function Framework two main tasks have been performed: the description and specification of the VNFs and 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 inactive and active part. In relation to the active one, the lifecycle states describe the VNF when it is up and running over a virtualised execution platform. On the other hand, the inactive lifecycle states span from the software development 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 interfaces with the rest of the T-NOVA system for making this information available for service orchestration.

5.2. Contributions to standards

T-NOVA Marketplace will be built considering the current NFV ETSI architecture and considering the on-going TMForum Best Practices for business services delivery under its frameworx (business process, information, application, and integration frameworks) but not restricted to it. For instance it has been decided to include a business catalogue to provide the marketplace offerings being aligned with TMForum proposes in its integration framework.

From the state of the art analysis performed, we gathered that the T-NOVA Marketplace concept is completely novel with regard to ETSI view [7]. T-NOVA introduces the marketplace aiming at opening the NFV market to third party developers for the provision of VNFs, a concept that currently falls outside the technical view of ETSI NFV.

On the other hand, supported by the TR228 TM Forum Gap Analysis related to MANO Work [11] it has been gathered that ETSI does not provide so far any more insight on the OSS/BSS (Operating Support System / Business Support System) of the operator apart from the definition of an interface; a detailed implementation model on how to manage operational and business support systems in a hybrid legacy and virtualized environment is something that ETSI is not addressing so far.

Though being out of the T-NOVA scope the interface between the MANO architecture and the existing OSS/BSS system of operators, T-NOVA aims to provide a first step on the direction of this research line by means of the implementation of the marketplace, which will implement some of the functionalities of a BSS system of an external operator, and what could be a first input for latest studies in the interoperability with OSS/BSS existing systems, which also TM Forum ZOOM intends to address in the future [8]. Other future work of TMForum that T-NOVA Marketplace is aligned to is the impact of the SLA Management in virtualization.

In relation to the NFV framework, we conclude 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.

Considering the evolution of ETSI standard for NFV and particularly the NFV Phase 2 the T-NOVA project will be actively participating to the ISG (Industry Study Group) activities and meetings providing different contributions. While ETSI NFV Phase 1 had the objective to provide informative requirements the focus of Phase 2 is on interoperability and also normative requirements with different level of detail will be provided. T-NOVA is active in the EVE WG (Evolution and Ecosystem) and in IFA WG (Interfaces and Architectures).

However, due to the fact that neither the specifications in ETSI nor the works in T-NOVA project are consolidated yet, some more collaboration with standards could be foreseen in the future.

The marketplace is a concept that doesn't require a full standardization because it is not necessary to implement the solution using components provided by different partners. In any case it is important that the marketplace, according to the VNFaaS and NFVIaaS use cases described in the ETSI GS NFV 001 (NFV Use cases) document, can be implemented using a standard Management and Orchestration framework. Therefore, we are working in order to identify the gaps that can hinder the implementation of this concept and we will provide contributions to the standards in order to overcome these limitations.

5.3. Future Work

The specification provided in this document has been built based on the requirements at T-NOVA system level described in previous deliverables together with some relevant parts of

the ETSI NFV work and TMForum best practices. The information assembled with this process has been the critical input into a two stage process: Stage 1 consisted on a research and design phase, where a system engineering approach was adopted to define the key functional components [3]. Stage 2 presented in this document has defined both the reference architecture and its functional entities and interfaces in a technology-agnostic manner to decouple the specifics of the implementations details. An additional third stage, which constitutes the activities within WP5 and W6, will address the details of the suitable technologies and their operation.

This report is the second and final version of the specification work reported in [12] that was delivered in September 2014. The current version includes slight refinements considering feedback from one year of analysis and implementation work in the abovementioned technical WPs.

Implementation work is still planned until end of December 2015 when the T-NOVA Marketplace and Function Store will be completely implemented. Year 2016 in T-NOVA project will be devoted to the system integration and testing of all its compoments, e.g. with the T-NOVA orchestrator and Virtualized Infrastructure Management. We do expect that system integration may detect some gaps or need of fine tuning the interface descriptions. Moreover, testing the system can identify some non-functional aspect that could suggest refining some part of this specification. For instance, it is difficult to figure-out performance and component interaction issues with the limited experience we have with actual NFV implementation in field.

Beyond T-NOVA, we have identified some 5G research and innovation projects towards T-NOVA Marketplace and Network Function framework can be a very good reference to build on. These are among others,

- 5GEx [42], that aims to build a sandbox to extend software networks in a multi-domain/operator environment. Although 5GEx is not expected to implement a full marketplace layer, it should specify a northbound API for end users to access the multi-domain service catalogue. T-NOVA marketplace can be a good reference to look at, for deriving functional and non-functional requirements on Business to Customer interface.
- SONATA [43], that focuses on the implementation of an enhanced modular orchestration platform and an SDK to facilitate service composition by service developers. T-NOVA GUI for Service Providers and its interface with T-NOVA orchestrator and T-NOVA Function Store can be seen as a relevant starting point for SONATA to build the SDK for service composition.

6. ANNEXES

6.1. Annex A - Requirements Specification

To specify requirements, the following template has been used, with the following fields:

Field	Meaning
Req. id	Requirement ID, of the form YYY_xx, in which YYY identifies the component and xx is numbered sequentially, starting from 01 por each different component. • Dashboard – D.x • Access Control – AA.x • SLA management module– SLA.x • Brokerage Module – B.x • Accounting module – A.x • Business Service catalogue store – BSC.x • Billing module – Bil.x • Interfaces marketplace - orchestrator – IMO.x • Network Function Store – FS.x
Use Case	Use case(s) in Deliverable D2.1 [1] from which the requirement is originated.
Domain	Technical domain to which the requirement belongs, selected out of the list: • Management and orchestration • Security • Service continuity • Operations • Market / Commercial operability
Requirement Name	Short requirement name
Requirement Description	Full requirement description. It usually corresponds to a sentence including the word “shall” (for mandatory requirements), ou “should” (for optional requirements).
Justification of Requirement	Rationale behind the requirement

Every requirement has an implicit severity level, which is indicated by the verb used to express it, in accordance to IETF RFC 2119 [1]:

- SHALL corresponds to an absolute requirement, something that must be supported by the implementation.
- SHOULD corresponds to a recommended, but optional, requirement – paraphrasing RFC 2119, this means that “there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course”.

Generally speaking, the criterion for assessing the severity level of each T-NOVA requirement was basically whether or not that specific requirement is indispensable for the system to deliver its basic function.

6.1.1. Detailed marketplace components requirements specification

Dashboard requirements for the three views:

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
D.1	UC1, UC2	Security - AA	Authentication and access control	The dashboard SHALL provide a “login in“ page for the different stakeholders to be authenticated	Stakeholders interacting with the T-NOVA system should be authenticated and authorised in order to be able to browse the Business Service Catalogue, issue SLA requests, or upload NFVs
D.2	UC1, UC2	Security - AA	Authentication and access control	The “login” page in the dashboard SHOULD offer to the different stakeholders means to use (username, password, OpenID, Google API) for authentication	Stakeholders interacting with the T-NOVA system should be able to use different authentication techniques to access T-NOVA
D.3	UC1, UC2	Security - AA	Authentication and access control	The “login” page in the dashboard SHOULD offer to the different stakeholders means to remember credentials when login on	Stakeholders interacting with the T-NOVA system should not be obliged to insert credentials when accessing again the system
D.4	UC1, UC2	Management & Orchestration	Web access	The Dashboard SHALL be accessible to authorized users via the Internet	The Dashboard will provide the necessary interface in order to be viewed over the Internet
D.5	UC1 UC 2	Management & Orchestration	Parallel Access	The Dashboard SHOULD provide multiple users login and no less than 10	The Parallel access will provide the necessary tools for every user to be able to provide his content
D.6	UC1, UC2	Management & Orchestration	Availability	The Dashboard SHOULD be available 24/7 365 days per year	The Dashboard must be always on in order to control every part of the T-NOVA infrastructure.

Dashboard requirements for the Customer view:

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
D.7	UC 2.2	Service continuity + Market/commercial operability - brokerage	Service offerings selection	The dashboard SHALL be able to allow a customer to search and watch offerings, selecting one or several of them	Service, SLA and price information need to be visualized by the customer to allows him to perform a selection
D.8	UC 2.2	Service continuity + Market/commercial operability – SLA management	SLA selection	The dashboard SHALL be able to allow a customer to select /negotiate among different SLA levels for a specific service	Service, SLA and price information need to be visualized by the customer to allows him to perform a selection
D.9			Service location	The dashboard SHALL enable the customer to configure certain parameters of the service, at least the location	
D.10	UC 5.1	Service continuity-SLA management	SLA visualization by customer	The dashboard SHALL be able to allow a customer to visualized SLA fulfilment information when he asks for it	Customer has to be able to visualize SLA fulfilment information when he asks for it
D.11	UC6	Market / commercial operability-billing	Bills visualization by customer	The dashboard SHALL be able to allow a customer to visualize his billing information when he asks for it	Customer has to be able to visualize his bills when he asks for it
D.12	UC2	Service Continuity-orchestrator	Service termination by the customer	The dashboard SHALL allow the customer to request service termination	The duration of the NS will be specified in the SLA, when the NS is no longer needed the system should de-compose the NS and cancel the SLA. Alternatively the SLA can be terminated by the customer on-demand.
D.13	UC3	Management & Orchestration– orchestrator (service configuration)	Customer service portal	The dashboard SHALL allow the customer to provide the means to configure the VNFs of a specific NS.	Any parameters required to configure the VNF (e.g. IP prefixes, traffic classes, etc.) must be accessible by the customer through the service portal.
D.14	UC3, UC4	Management & Orchestration– orchestrator (service monitoring)	Customer notification - VNF starts / fails to start	If the VNF starts correctly, the T-NOVA system SHALL be able to notify the customer about this event. The customer service portal shall provide this information to the customer. If the service fails to start correctly, the T-NOVA SHALL be able to notify the customer about this event.	The customer must get feedback about success or failure of his/her service request
D.15	UC4	Management & Orchestration, Elasticity	Customer service portal - Scale In/Out	The dashboard SHOULD provide a means for a customer to request either a “scale out” or “scale in” of a deployed VNF Service. When the customer requests a VNF “scale out” or “scale in” the customer will have the option to reuse a	

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
				previous configuration or to specify a new configuration.	
D.16	UC4.2	Management & Orchestration, Elasticity	Customer/SP Scale Down VNF service	The Dashboard SHOULD be able to allow the customer to scale down an existing VNF service	Scale down is necessary to ensure the T-NOVA system can meet the changing needs of the customer
D.17	UC5_1 – UC7_4	Management & Orchestration-Brokerage	Service catalogue	The dashboard SHALL be able to provide all services that are active for the authorised customers.	Service catalogue is essential, because it provides to the authorized customers the possibility to identify the service that needs to be terminated
D.18		Dashboard, Service Configuration	Collect NS parameters	The dashboard SHALL collect the service's pre-defined parameters when a customer selects a specific NS	See **[1]**. Note that the **Customer** can only choose from a list of authorized NS.
D.19		Dashboard, Service Configuration	Display NS parameters	For the NS chosen by the customer, the Dashboard SHALL display the parameters for a selected NS by the customer, taking into account their properties (default values, enumerated values, read-only, inter-relationship, etc.)	Display should take into account related/inter-dependent parameters (see [1])
D.20		Dashboard, Service Monitorisation	Collect and display metrics	The dashboard SHALL display the service's pre-defined metrics	See [3].
D.21		Dashboard, Service Monitorisation	Manage configuration of service parameters (Customer)	The dashboard SHALL allow the customer to configure some service parameters	Includes not only the graphical configuration (e.g., pie-graph vs. bar-graph, graph title, etc.)
D.22		Dashboard, Service Monitorisation	Display configured metrics	The Dashboard SHALL display the configured metrics	

Dashboard requirements for the Function Provider view:

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
D.23	UC2.2	Service continuity + Market/commercial operability – Function Store	VNFs and SLAs description by the FPs	The dashboard SHALL be able to allow Function Providers to describe their VNFs and metadata (conditions, SLA, price).	VNFs and SLAs description need to be stored in the system to allow the customer to browse through this information
D.24	UC1	Security-AA	FP authentication, certification	The dashboard SHALL allow the FPs to be authorised by the system in order to advertise, upload and modify any VNF. The acceptance of a FP will be subject of bilateral discussions between the developer and the T-NOVA operator.	This is essential in order to control the access to the Brokerage and the Function Store and increase security.
D.25	UC1	Operational-FS	VNF Advertisement	The dashboard SHALL allow FPs to be able to advertise the VNF capabilities in the system.	The FP for each VNF that is uploaded or modified needs to notify the Orchestrator in order that the Service catalogue is updated. This action is called advertisement of the VNF. The advertisement request should contain the VNF metadata, including name, id, pricing information, requirements and VNF capabilities.
D.26	UC1	Operational-FS	VNF withdrawal	The dashboard SHALL allow FPs to be able to removed one of tis VNFs from the system.	The FP for each VNF that is withdrawn needs to notify the Orchestrator in order that the Service catalogue is updated.
D.27	UC1	Operational-FS	VNF Upload	The dashboard SHALL allow FPs to be able to upload the packaged VNF to the Functions Store.	The system should offer a method to the FP for uploading and storing the packaged VNF to the Function Store. Both VNF image and its metadata are uploaded. When a particular VNF is requested the Orchestrator will instantiate this VNF to the appropriate NFVI-PoP.

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
D.28	UC1	Security/Operational-FS	VNF validation	VNF validation SHOULD be checked.	The VNF shall be validated before being included into the Function Store. Validation is an off-line process. It is out-of scope of T-NOVA.
D.29	UC1	Operational- FS	FP VNF status monitoring	All the VNFs of the same developer SHALL be browsable in the developer dashboard, from where the developer is able to monitor the status and other statistical data (popularity, rating, comments etc.).	This requirement covers the need for supporting the monitoring of each VNF by the FP in terms of availability, popularity, malfunctions and alerting.

Dashboard requirements for the Service Provider view:

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
D.30	UC2.2	Service continuity + Market/commercial operability – service catalogue	Services and SLAs description by the SP	The dashboard SHALL be able to allow Service Providers to describe their services and conditions description (service (+ service description), SLA, price)	Services and SLAs description need to be stored in the system to later allow the customer to browse through this information
D.31	UC 3.1	Service continuity-SLA management	SLA visualization by the SP	The dashboard SHALL be able to allow a customer to visualize SLA fulfilment information on demand	Customer has to be able to visualized SLA fulfilment information when he asks for it

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
D.32	UC6, UC2	Market / commercial operability-billing	Bill cycle agreement	The dashboard SHALL allow the SP to be assigned a bill/billing mode cycle with the FPs	Billing procedure needs to know when a bill cycle finishes (for non pay- as-you-go services).
D.33	UC6	Market / commercial operability-billing	Bills visualization by SP	The dashboard SHALL be able to allow a SP to visualize his billing information on demand	Customer has to be able to visualized his bills when he asks for it
D.34	UC2, UC3, UC4	Operational, Service Continuity, Management & Orchestration-orchestrator	NS Composition by the SP	The dashboard SHALL allow the SP to compose a NS from atomic VNF instances available at the NF Store and define the logical topology among the several components (see [2])	The creation of a NS from the combination of atomic/simple VNF is important in order to simplify the process provision of NS to the customers and avoid complex path calculations
D.35	UC3, UC4, UC5	Management & Orchestration-Orchestrator	Resource monitoring	The dashboard SHALL allow the SP to monitor and collect information about consumption and availability of resources (computational, storage, network) on a real time basis, including the resources consumed by each specific VNF instance.	Monitoring is essential to ensure that the deployment of VNF's onto hosting infrastructure is performed adequately. Monitoring is also r provides essential metrics required by operations such as rescaling, billing, etc.
D.36	UC4	Management & Orchestration, Elasticity-orchestrator	SP service portal - Scale Up/Down	The dashboard SHALL provide the means for a SP to request either to up or down scale the resources allocated to a deployed VNF Service based on the SLA evaluation.	The T-NOVA system must provide the ability for customers to request additional resources or the removal of resources from a deployed VNF service.
D.37	UC4	Management & Orchestration, Elasticity-orchestrator	SP notification - VNF is removed	A notification SHALL be shown to the SP if the VNF and its host VMs are successfully removed from the T-NOVA system.	The customer must get feedback about the success or failure of their service request
D.38	UC4	Management & Orchestration; Elasticity-orchestrator	SP notification - VNF rescale	The dashboard SHALL notify the SP when their request to rescale a VNF has been successfully completed.	The customer must get feedback about the success or failure of their service request
D.39	UC4.2	Management & Orchestration, Elasticity-orchestrator	SP Scale Out VNF	The dashboard SHALL allow the SP to choose to scale in/out an existing VNF.	Customers will request increases in VNF services to meet business needs
D.40		Dashboard, Service Composition	Browse available VNFs	The Dashboard SHALL allow the Service Provider to browse the available (and authorized) VNFs	The list of available VNFs is a combination of the list provided by the Orchestrator and the list of authorized services for the Service Provider (in a multi-Service Provider

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
					scenario).
D.41		Dashboard, Service Composition	Select and configure one or more VNFs	The dashboard SHALL allow the SP to compose a Network Service from a set of available VNFs [2].	Note that more than one instance of the same VNF can be used in the composition of a NS. This configuration includes the definition of the selected VNFs' connection graph. This connection graph is, in the simplest case, the connection of the output of a VNF instance to the input of another VNF instance. In more complex cases there might be some form of processing the output (e.g., max/min/average, step, etc.) before delivering to the input.
D.42		Dashboard, Service Composition	Define composed service parameters	For a given composed service, the Dahsboard SHALL allow the SP to define its parameters and map those to the parameters of the VNFs used in the composition.	Includes the definition of related/inter-dependent parameters (for displaying purposes, see [2]) and mapping VNF's parameters to the composed service parameters, as well as the indication of default values, mandatory or read-only properties, etc.
D.43		Dashboard, Service Composition	Define composed service indicators	For a given composed service, the dashboard SHALL allow the SP to define its metrics, mapping from those available from each of the VNF used in the composition	Includes the definition of related/inter-dependent indicators (for displaying purposes, see [2]) and mapping VNF's metrics to the composed service metrics.
D.44		Dashboard, Service Composition	Manage life-cycle of composed services	The Dashboard SHALL allow the SP to manage the life-cycle of the composed services.	Note that if a composed service is being used, it can not be removed entirely bit can be dropped from the list of available services in the Marketplace
D.45		Dashboard, Service Composition	SP-FP trading	The Dashboard SHOULD allow the SP to interact with the FPs in order to trade various VNFs	Auction/Trade inside the Service Composition

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
D.46		Dashboard, Service Monitorisation	Collect and display metrics	The Dashboard SHALL display the service's pre-defined metrics	See [3].
D.47		Dashboard, Service Monitorisation	Manage configuration of metrics (SP)	The Dashboard SHALL allow the configuration of service parameters by the SP.	Includes not only the visibility of the metric and any other graphical configuration (e.g., pie-graph vs. bar-graph, graph title, etc.) but also eventually some form of combining two or more metrics. A subset of these metrics will likely be related to the agreed SLA.
D.48		Dashboard, Service Monitorisation	Display configured metrics	The Dashboard SHALL display the configured metrics	

Access Control

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
AA.1	UC2	Security	Authentication and access control	AA module SHALL support mechanisms for authentication and authorisation for the different stakeholders. The authorization is based on the associated roles and permissions	Stakeholders interacting with the T-NOVA system should be authenticated and authorised in order to perform tasks such as uploading a NF or purchase a VNF
AA.2	UC2	Security	Secure communication	AA framework SHOULD provide encryption.	Encryption should be used, in order to prevent eavesdropping.
AA.3	UC1	Security	FP authentication, authorization	AA SHALL authenticate and authorise the FPs in order to advertise, upload and modify VNFs	Each FP that interacts with the Brokerage and the Function Store needs to be authenticated and authorised by the system. This is essential in order to control the access to the Broker and the Function Store and increase security.

AA.4		Security	Stakeholders Authorization	Profiles are created according to T-NOVA roles and stakeholders are assigned roles based on their responsibilities and qualifications	Based on the different stakeholders responsibilities and qualifications and the tasks that they will undertake, roles will be created
AA.5		Security	Stakeholders Authorization	Roles can be reassigned or granted new permissions if needed. Which means they SHALL be updatable and revocable	As responsibilities might change or extended/reduced over time, roles SHALL be flexible enough to be reassigned, extended with new features, or revoked

Brokerage:

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
B.1	UC2.1	Operational	Trading/Bidding	The Brokerage SHALL be able to perform auctions among Service provider and Function providers	The Brokerage must be able to initiate auctions whenever it is required.
B.2	UC2.1	Operational	Trading/Bidding	The Brokerage SHOULD take into account the SLA parameters	The Brokerage must be able to initiate auctions by using the SLA parameters as a prerequisite
B.3	UC2.1	Operational	Trading/Bidding	The Brokerage SHOULD return the best possible Function to the SP	The brokerage must be able to auction and return the best possible results to the SP
B.4	UC2.1	Operational	Trading/Bidding	The Brokerage SHOULD be able to accommodate multiple FP to a SP	The brokerage must be able to auction among the available FPs
B.5	UC2.1	Operational	Trading/Bidding	The Brokerage SHOULD be able to take into consideration predefined Atomic Trading Units	The brokerage must be able to auction among the available Functions with predefined Parameters

Business Service Catalogue

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
SC.1	UC2.2	Service continuity + Market/commercial operability	Services and SLAs description	The service catalogue SHALL be able to store all the available NSs in the T-NOVA marketplace, specifying SLA level and price.	Services and SLAs description need to be stored in the system to allow the customer to browse through this information
SC.2	UC2.2	Service continuity + Market/commercial operability	Services and SLAs description	The service catalogue SHALL be browsable by the SP	The service catalogue must support advance searching capabilities, to support search with different criteria, such as price, SLA, etc.
SC.3	UC2.2	Service continuity + Market/commercial operability	Services and SLAs description	The service catalogue SHALL support operations of creating/updating /deleting NSs.	The service catalogue needs to support updating and deleting operations, regarding the NSs.

SLA management module

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
SLA.1	UC2.2	Service continuity	SLA information customer-SP storage	The SLA management module SHALL store all the SLA agreements between a customer and the SP for each service.	SLA agreements must be stored in order for service monitoring to determine if the SLA has been fulfilled or not
SLA.2	UC2.2	Service continuity	SLA information SP-FPs storage	The SLA management module SHALL store all the SLA agreements between the SP and the FPs for each VNF.	SLA agreements must be stored in order for a VNF monitoring to determine if the SLA has been fulfilled or not
SLA.3	UC4, UC5, UC5.1	Management & Orchestration, Operations, Service Continuity	SLA – orchestrator interface	The SLA management module SHALL be connected to the orchestrator to let it know about the agreed SLA for each service. (When the SLA is not fulfilled the orchestrator will have to initiate the applicable action, e.g. rescaling)	SLA management and monitoring is considered essential for the commercial applicability of the T-NOVA system. The T-NOVA system must determine when an SLA is in breach and trigger corrective actions.

SLA.4	UC 5.1, UC6	Market / commercial operability	SLA fulfilment information storage (from the orchestrator)	The SLA management module SHALL store all the information about SLA fulfilment for eventual compensations or penalties for later billing.	In order to compensate the customer economically for not achieving the SLA agreed, the billing system must have this information
SLA.5	UC 5.1, UC6	Market / commercial operability	SLA – accounting/billing interface	The SLA management module SHALL be connected the accounting/billing system to let it know about eventual compensations or penalties when the SLA has not been fulfilled for a specific service.	In order to compensate the customer economically for not achieving the SLA agreed, the billing system must have this information
SLA.6	UC 5.1	Service continuity	SLA visualisation by customer and SP	The SLA management module SHALL be connected to the Dashboard to allow a customer and SP to visualize SLA fulfilment information when requested.	Customers and SPs has to be able to visualize SLA fulfilment information since this is the real service level they are getting and paying for.
SLA.7	UC2.2	Service Continuity	SLA procedure mechanisms	The SLA management module SHALL provide mechanisms to get an agreement presented and agreed between the parties	The involved parties need to be informed about the SLA offer from the other party and be able to agree or disagree

Accounting module

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
Ac.1	UC3	Management & Orchestration	Accounting notification - VNF starts	The accounting system SHALL know if a VNF or network service starts correctly.	For billing purposes, the accounting system has to be notified about the start of the VNF service instance
Ac.2	UC 5	Market / commercial operability	Resources usage for billing	The accounting system SHALL store all the information about resources usage by each service for later billing purposes.	Billing procedure needs to know the services that have taken place and when they start and end.
Ac.3	UC 6	Market / commercial operability	Price information for billing	The accounting system SHALL store the information about prices agreed by each customer for later billing purposes.	Billing procedure needs to know the price to be applied for service
Ac.4	UC 5.1, UC6	Market / commercial operability	SLA billable items	The accounting system SHALL be aware of the information about SLA fulfilment for billing compensations or penalties.	In order to compensate the customer economically for not achieving the SLA agreed, the billing system must have this information
Ac.5	UC6, UC2	Market / commercial operability	Bill cycle	The accounting module SHALL be able to provide the billing related information for any given period for each customer, and SP.	Billing procedure needs to generate the bills for customised periods of time.

			Components relationships	The accounting system SHALL store the necessary information of the service and function instances,, agreements, providers and customers.	All the elements involved in the issue of a bill must be linked and related.

Billing module

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
Bil.1	UC 6	Market / commercial operability	Price information for customer billing	The billing module SHALL receive the information about prices agreed by each customer for each service.	Billing procedure needs to know the price to be applied for service
Bil.2	UC 6	Market / commercial operability	Price information for SP billing	The billing module SHALL receive the information about prices agreed by each SP for each VNF.	Billing procedure needs to know the price to be applied for service
Bil.3	UC 6	Market / commercial operability	Bill issuing	The billing module SHALL issue bills when the customer's bill cycle finishes or service pay-as-you-go finishes and stores them within the customer profile.	Billing procedure needs to know when a pay-as-you-go service starts and finishes.
Bil.4	UC6		Billing-accounting interface	The billing SHOULD receive all the information needed for billing from the accounting module.	
Bil.5	UC6		Billing-User management interface	The billing module SHALL get the specific user related information along with the information received from the accounting system for billing purposes.	
Bil.6	UC6		Billing-Dashboard interface	The billing module SHOULD provide information (statistics and graphs) to the dashboard.	It's desirable by the customer to have statistics that represent what they are paying for in an easy-to-understand way.

Marketplace – orchestrator interface

Req. id	Domain	Requirement Name	Requirement Description	Justification of Requirement

IMO.1	Orchestrator, Marketplace	Provision a new network service	The Marketplace SHALL use this interface to inform the Orchestrator to provision the network service, after the Customer has selected and configured the network service. The Orchestrator SHALL read the SLA and the date/time to start the new network service. Each NS can be composed of one or more VNFs.	The date/time of start/end the service are part of the SLA.
IMO.2	Orchestrator, Marketplace	Change configuration of a deployed network service	The Marketplace SHALL use this interface to change the configuration of an already provisioned network service on the Orchestrator.	
IMO.3	Orchestrator, Marketplace	Provide network service state transitions	The Marketplace SHALL use this interface to know about the state transitions of a given network service, e.g. to allow start and stop billing for the service.	It is assumed that each NS has a pre-defined state-diagram, like 'Ready to run', 'Running', 'Stopped', etc., that is also known to the Marketplace.
IMO.4	Orchestrator, Marketplace	Provide network service monitoring data	The Marketplace SHALL use this interface to show the Customer how the subscribed network service is behaving, how it compares to the agreed SLA and bill the service usage.	This interface will very likely have to support very high volume traffic.
IMO.5	Orchestrator, Marketplace	Terminate a provisioned NS	The Marketplace SHALL use this interface to request the Orchestrator to terminate provisioned NSs	It is assumed that the impact on the dependent modules like billing, are taken care by the Marketplace (see NFVO.04). SLA Management is part of the Marketplace. Either after a customer's request or by the pre-defined ending date had been attained, the SLA Management notifies the Orchestrator of the end of the SLA.
IMO.6	Orchestrator, Marketplace	Secure communication	Interfaces between the Marketplace and the Orchestrator should be secured.	Encryption should be used, in order to ensure security against eavesdropping. Even between the Marketplace and the Orchestrator, since the Marketplace is really a set of distributed apps.

Marketplace – Function Store

IFM.1	Function Store Marketplace	Provide available VNFs	The Marketplace SHALL use this interface with the Orchestrator to provide the Service Provider with a list of the VNFs, so that it can select and parameterise them, or use them in the composition of a new network service.	It is assumed that this VNF metadata includes a URL/repository name from which to fetch the actual VNF software and install it on the previously allocated infrastructure (see NFVO.10 below). Note that, although this information will most certainly have to be cached on the Orchestrator's side for performance reasons, the available VNFs will be dynamic, so updates to this cached information will be rather frequent.
IFM.2	Function Store Marketplace	Provide available VNFs	The Marketplace SHALL use this interface to allow the FPs to upload VNFs to the Function Store	

6.1.2. Detailed Network Function Store requirements specification

Req. id	Use Case	Domain	Requirement Name	Requirement Description	Justification of Requirement
FS.1	UC1	Operational	VNF Upload (or Publish)	The Function Store SHALL be able to store all the packaged VNFs and their associated metadata.	The system shall offer a method to the FP for uploading and storing the packaged VNF to the Function Store. When a particular VNF is requested the Orchestrator will instantiate this VNF to the appropriate NFVI-PoP.
FS.2	UC1	Security/ Operational	VNF validation	VNF validation SHOULD be checked.	The submitted VNF is validated by the T-NOVA Function Store in order to increase security and integrity of the VNF package. This is out of the scope of T-NOVA
FS.3	UC1	Security/ Operational	VNF Identification	The Function Store SHALL provide a unique identification ID to each certified, advertised VNF.	The VNF ID will be the reference name used by the system for monitoring purposes.
FS.4	UC1	Operational	VNF Modification/ Withdrawal	The Function Store SHALL allow modification/ withdrawal of the packaged VNFs and their associated metadata.	The system will offer a method to the FP for modification and/or withdrawal of the packaged VNF in the Function Store.
FS.5		Operational	VNF Download	The Function Store SHALL allow downloading of the packaged VNFs and their associated metadata.	The system will offer a method for downloading information associated to VNFs from the repository.

FS.6		Operational	VNF Directory	The Function Store SHALL allow listing of packaged VNFs and their associated metadata.	The system will offer a method to list the content of the data stored in the repository.
FS.7		Operational	VNF Notification	The Function Store SHALL be able to notify changes in its status.	The system will provide a mechanism to notify an update or change in the content of the repository. This is basically used to make the Orchestrator databases in sync with the Function Store.
FS.8		Security/ Operational	VNF Authentication	The Function Store SHALL allow access only to authenticated users.	The system (AA module) will grant operation and control of the Function Store upon authentication of FPs.
FS.9		Operational/ Management	VNF User Privilege	The Function Store SHALL provide different levels of privileges to users (e.g. user and root levels).	FP will have user access to the repository. In addition, the Function Store will offer a powerful management interface to administrators.
FS.10		Operational/ Performance	VNF User Concurrency	The Function Store SHOULD provide multi-user capability.	Concurrent requests should be managed by the Function Store repository.
FS.11		Operational/ Security	VNF Data Protection	The Function Store SHOULD guarantee security mechanisms for transmission of data.	The data should be protected (e.g. encrypted) when transferred to/from the Function Store repository.
FS.12		Operational/ Security	VNF Data Oblivion	The Function Store SHOULD guarantee that cancelled data will be completely disregarded.	The FP should be able to delete his/her data from the repository in a definite manner.
FS.13		Operational/ Performance	VNF Service Continuity	The Function Store SHOULD be available without interruption regardless of time or day (24/7/365).	The Function Store service should be offered as a nonstop service.

6.2. Annex B. Dashboard Mock-up

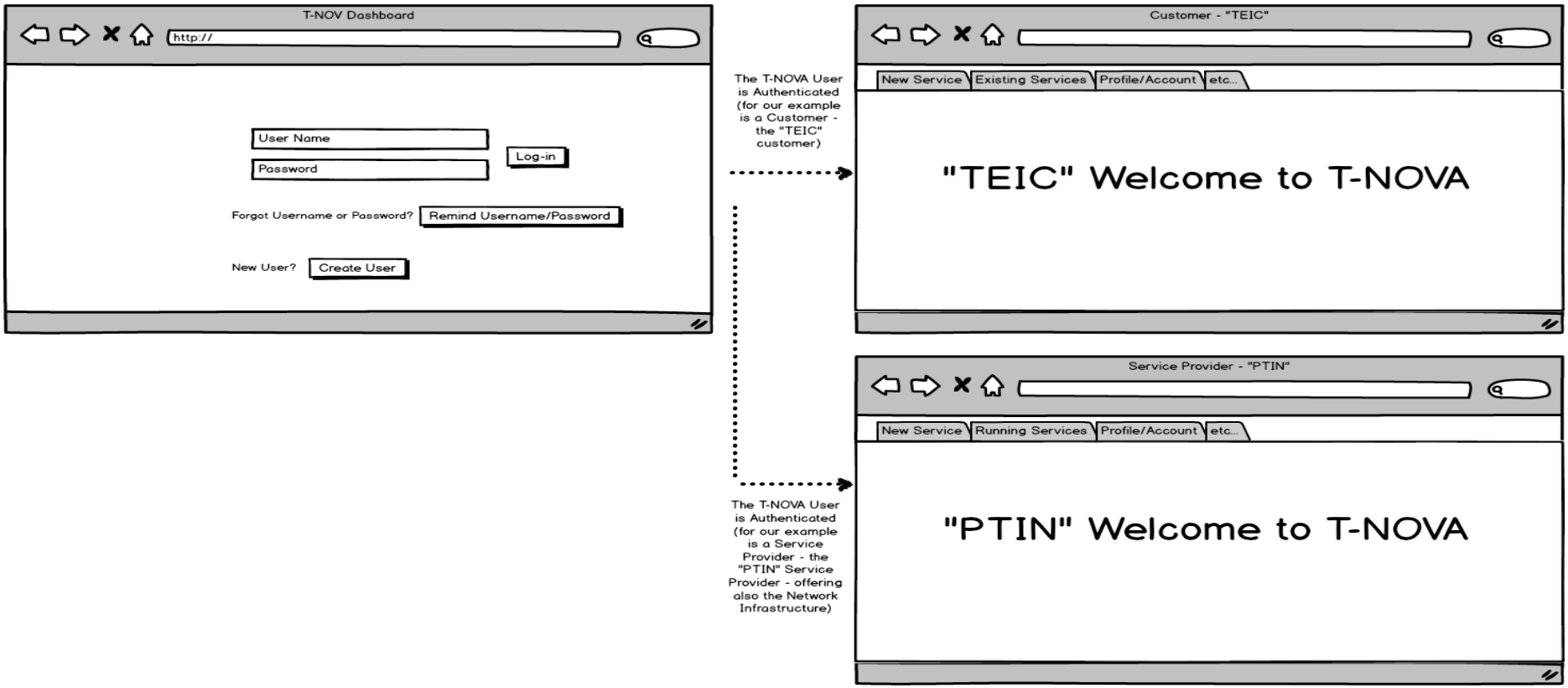

Figure 6-1 Dashboard Login Screen

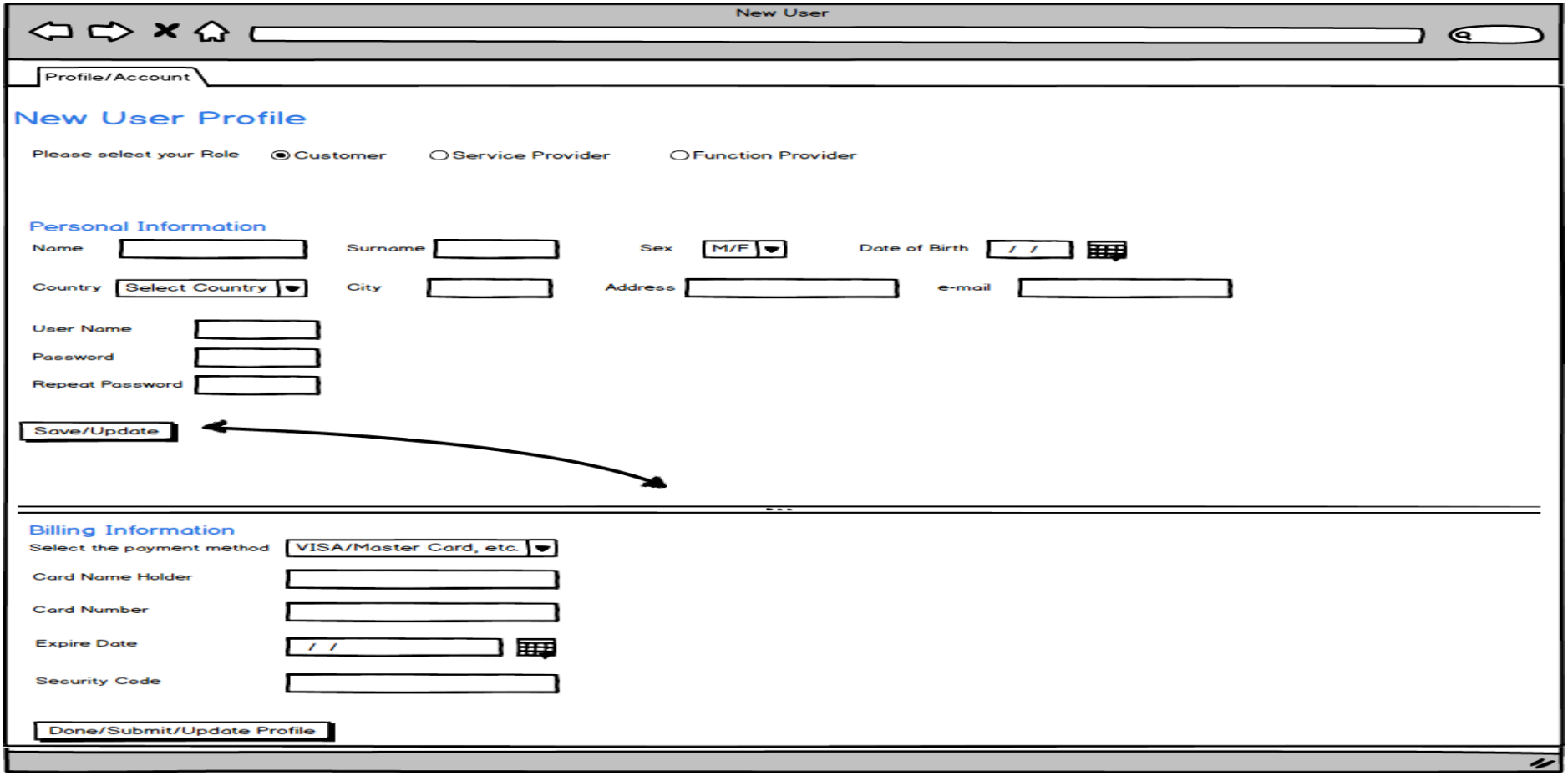

Figure 6-2 New User Profile

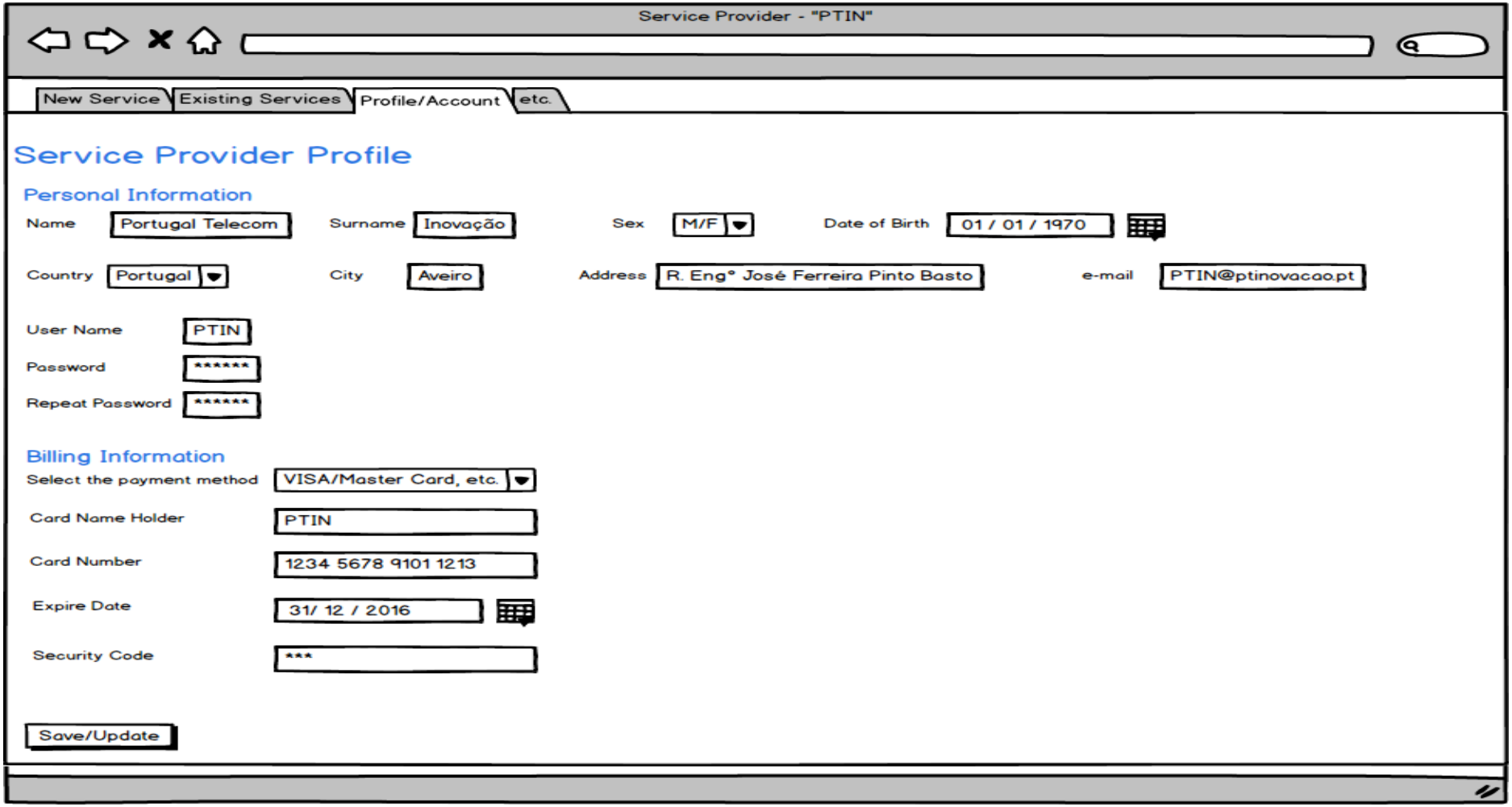

Figure 6-3 Service Provider Profile Screen

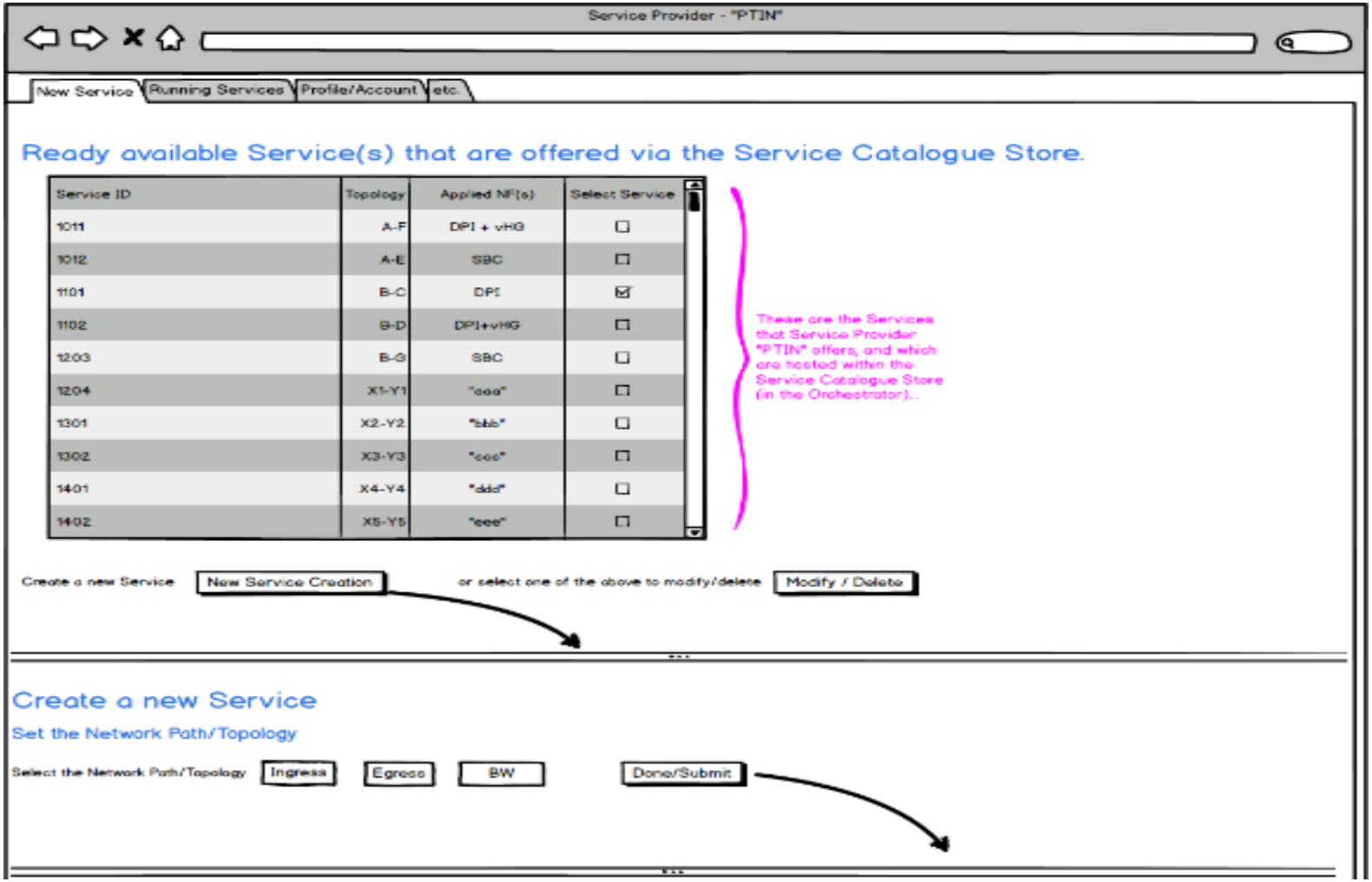

Figure 6-4 Service Provider New Service 1/2

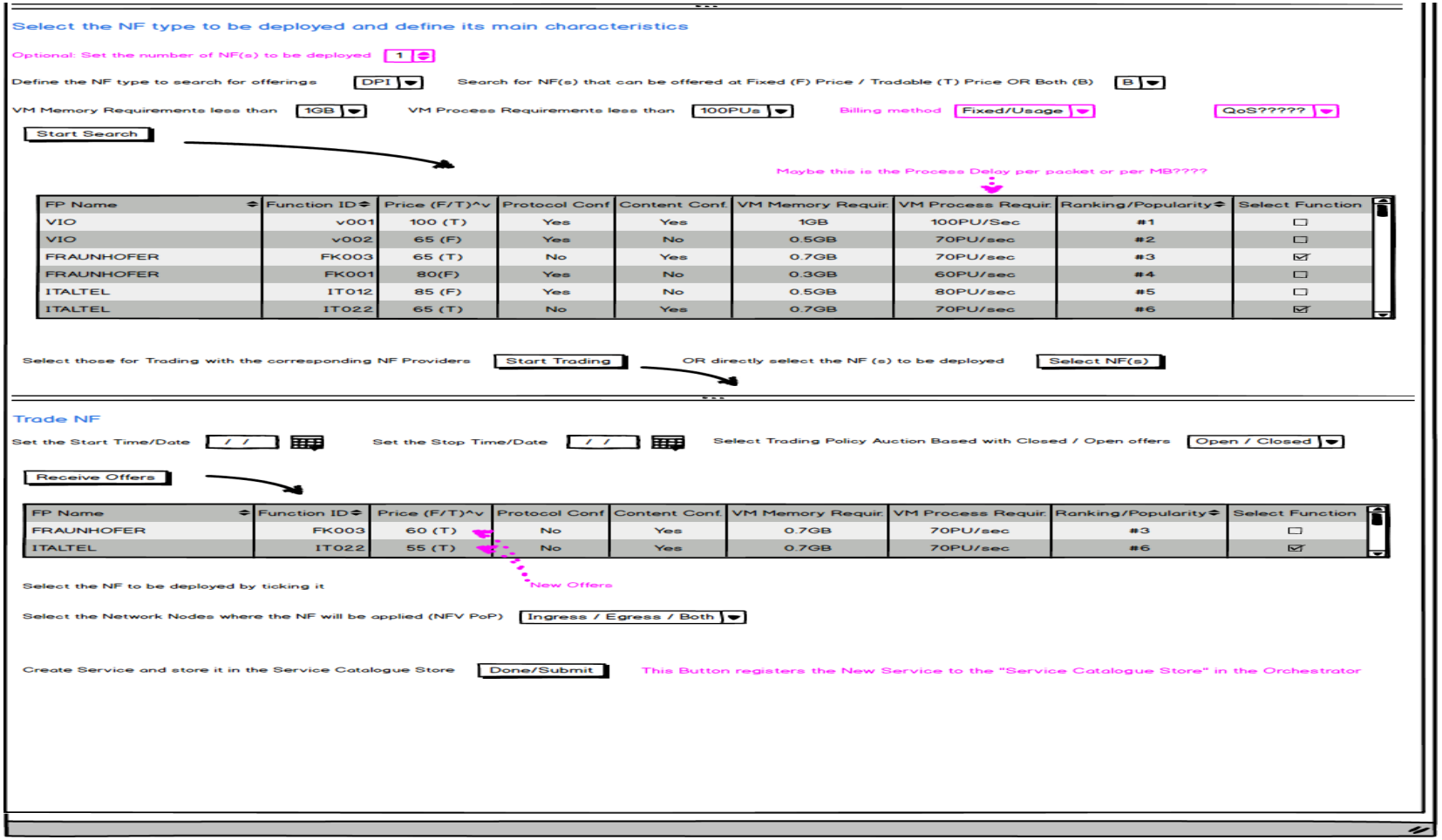

Figure 6-5 Service Provider New Service 2/2

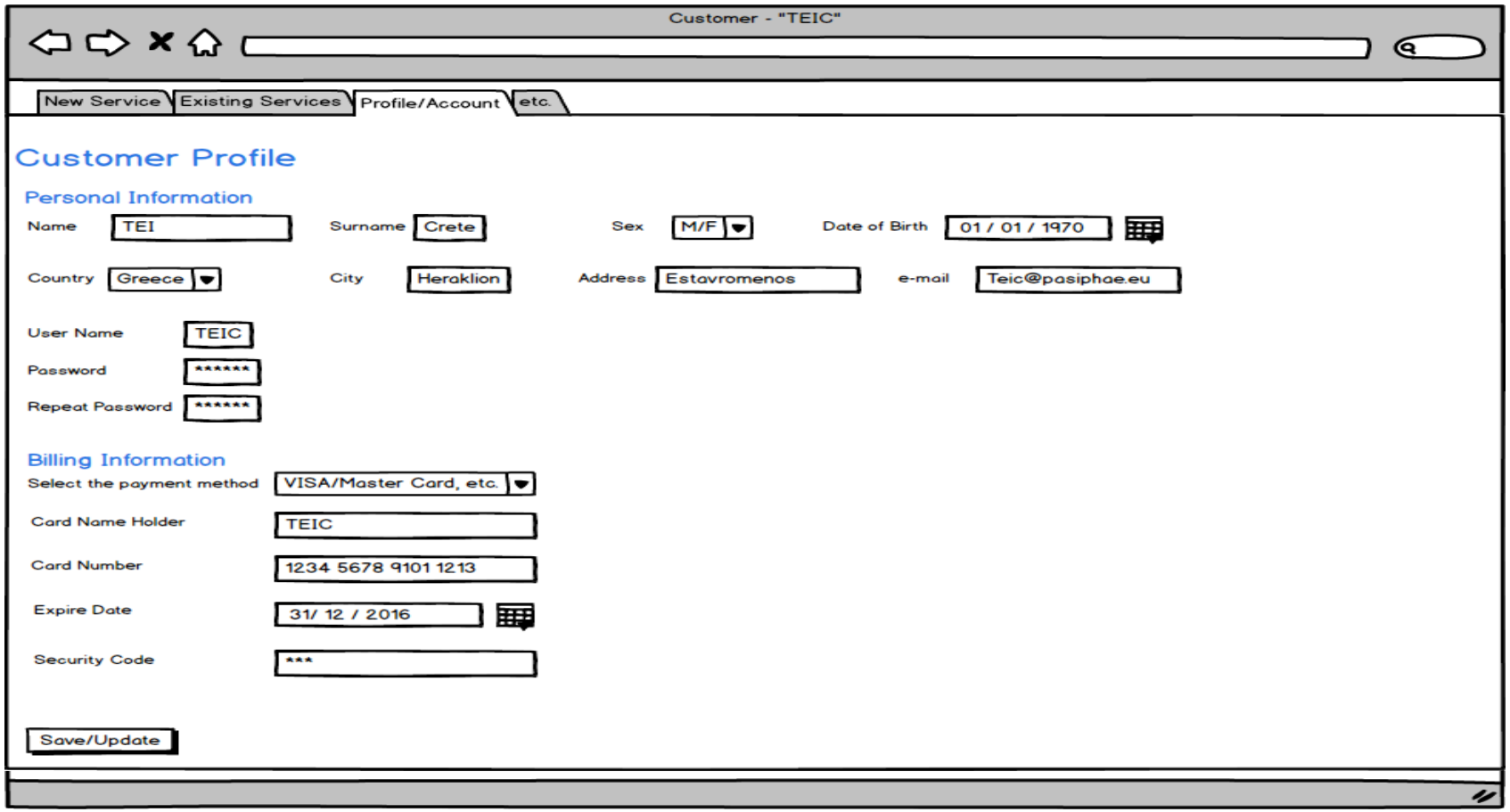

Figure 6-6 Customer Screen

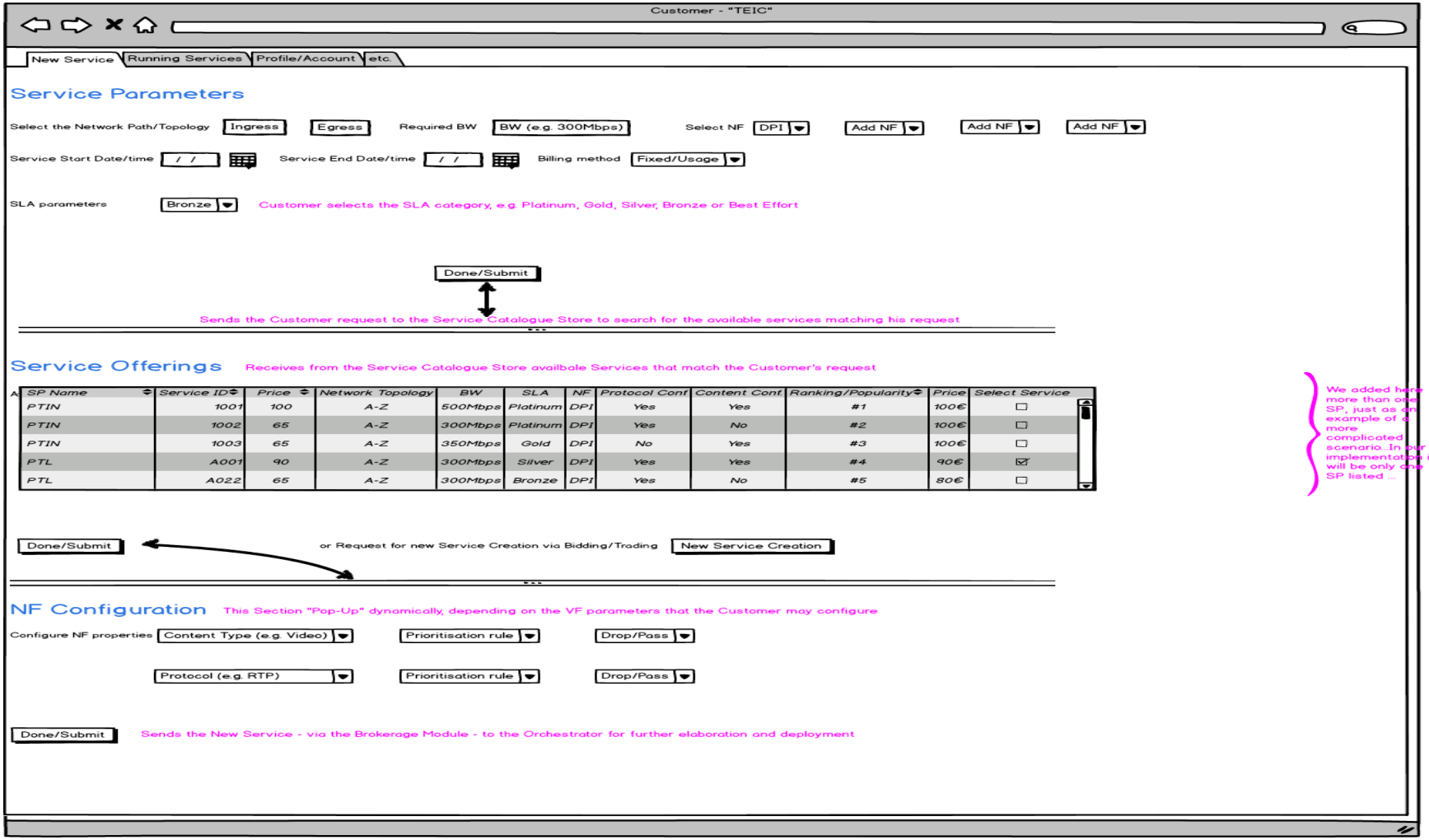

Figure 6-7 Custumer New Service


Customer - "TEIC"
New Service
Existing Services
Profile/Account
etc.
Existing Services
Displays the Services that the Customer has already acquired
SP Name | Service ID | Topology | BW | SLA | Network Util (%) | Applied NF(s) | VMs Util (%) | Start Date/Time | Stop Date/Time | Service Status | Select Service
PTIN | 1011 | A-F | 1000Mbps | Platinum | 80 | DPI + vHG | 80 | 01/01/2014 | 31/12/2016 | Running
PTIN | 1012 | A-E | 100Mbps | Best Effort | 0 | SBC | 0 | 01/02/2014 | 31/12/2016 | Paused
PTL | A001 | A-Z | 300Mbps | Silver | 60 | DPI | 90 | 19/06/2014 | 31/12/2016 | Running
Select a Service to start monitoring it
Done/Submit
Enables the Customer to start Monitoring a Service or Modify it
Service Monitoring
Service Description
SP Name | Service ID | Topology | Network Util (%) | Applied NF(s) | VMs Util (%) | Date/Time Started | Service Status | Debit (€)
PTL | A001 | A-Z | 60 | DPI | 90 | 19/01/2014 | Running | 1500
Start/Stop Service
Release Service
Network Utilisation
Data from the Network Monitoring Modules (e.g. via OpenDaylight)
View Log File
VMs Utilisation
Data from the Cloud Monitoring Modules (e.g. via OpenStack
Average
View Log File
Critical Events
This could be the SLA monitoring (fullfilment)
View Log File
NF Utilisation
View Log File
Flows Dropped/Passed
View Log File
Billing/Charging
Data from the Billing Modules
View Log File

Figure 6-8 Customer Existing Services

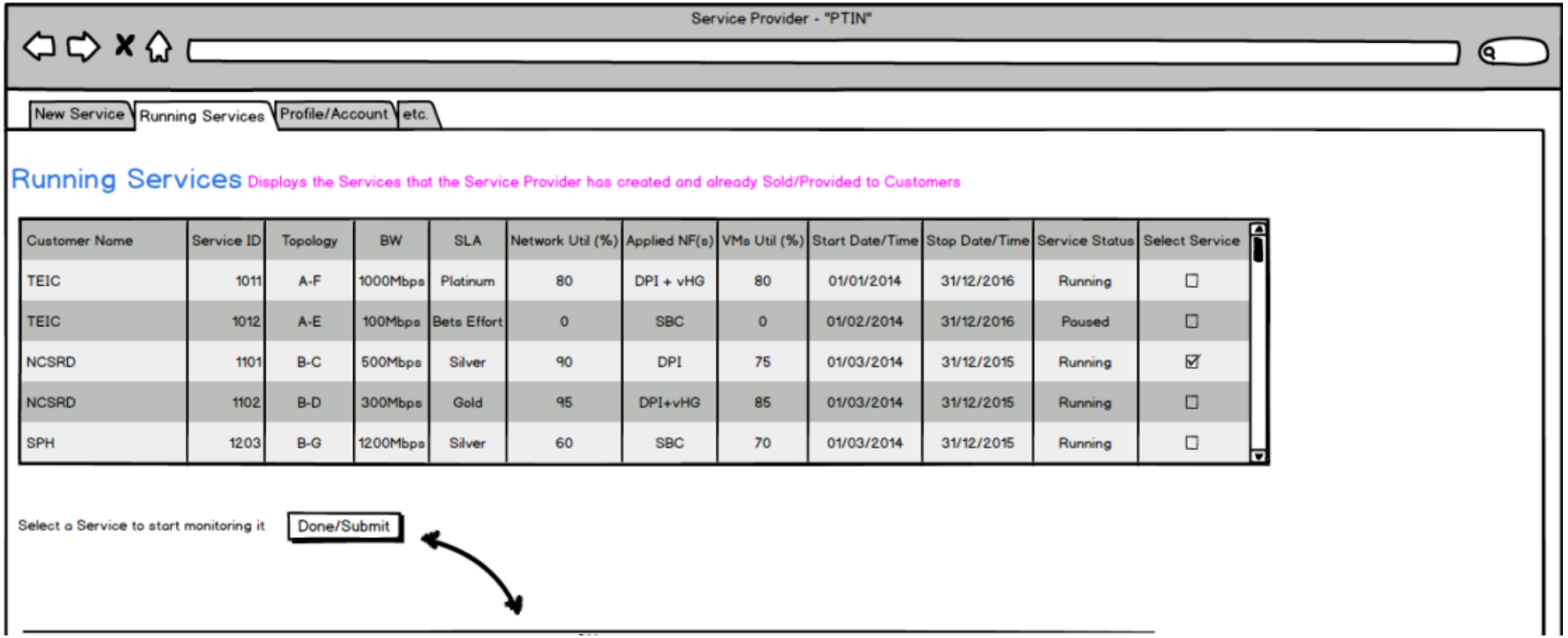

Figure 6-9 Service Provider- Running Services 1/2

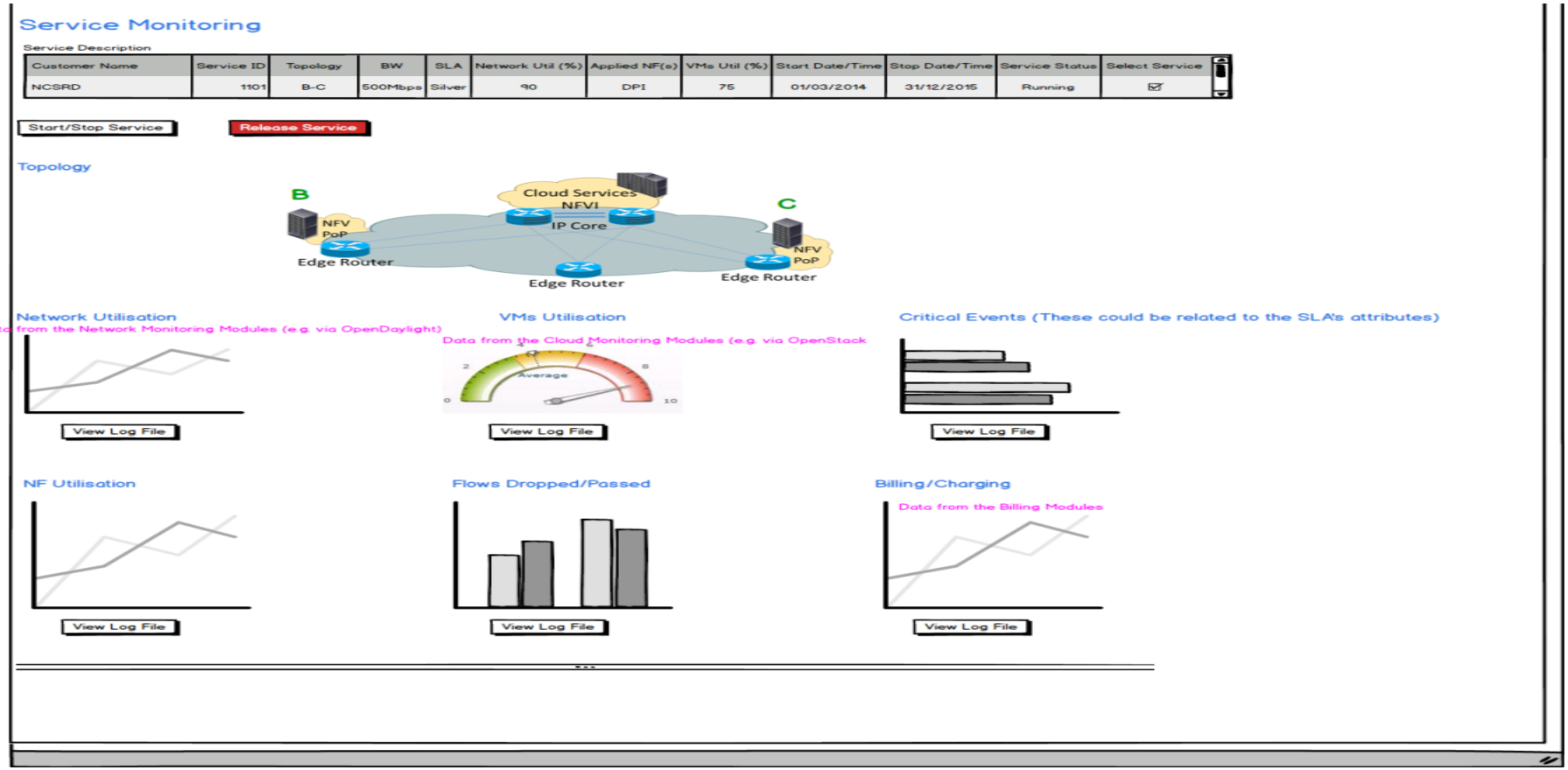

Figure 6-10 Service Provider- Running Services 1/2

7. REFERENCES

[1] Deliverable D2.1 – System Use Cases and Requirements. T- NOVA project.

[2] Deliverable D2.31 - Specification of the Infrastructure Virtualisation, Management and Orchestration - T-NOVA project.

[3] Deliverable D2.22 - Overall System Architecture and Interfaces - Final - T-NOVA project.

[4] Deliverable D6.01 - Interim Report on T-NOVA Marketplace implementation, T-NOVA project (Dec' 2014).

[5] Deliverable D6.3 - Users dashboard. T-NOVA project.

[6] ETSI GS NFV-SWA 001" Network Functions Virtualisation (NFV)- Virtual Network Functions Architecture.

[7] ETSI NFV ISG, "NFV-MAN 001 NFV Management and Orchestration," July 2014. [Online]. Available: http://docbox.etsi.org/ISG/NFV/Open/Latest_Drafts/NFV-MAN001v061%20management%20and%20orchestration.pdf.

[8] TM Forum, "TM Forum WebSite," http://www.tmforum.org.

[9] TMF ZOOM, «TMF ZOOM,» [En línea]. Available: http://www.tmforum.org/ZOOM/16042/home.html. [Último acceso: 2014].

[10] TMF "TR227 TM Forum Specifications relevant for MANO Work V1.0.1", 2014.

[11] TMF, TR228 TM Forum Gap Analysis related to MANO Work V1.0.1, 2014.

[12] Deliverable D2.41 - Interim Report on Specification of Network Function Framework and T-NOVA Marketplace. T-NOVA project (Sep'2014).

[13] TMForum: Virtualization: When will NFV cross the chasm? February 2015.

[14] TMForum Catalyst - https://www.tmforum.org/collaboration/catalyst-program/current-catalysts/.

[15] TMF Catalyst Project: Service Bundling in a B2B2X Marketplace. Available: [Accessed 2014] http://www.tmforumlive.org/service-bundling-in-a-b2b2x/.

[16] TMForum Catalyst project: How to maximize profitability with NFV orchestration - June 2015.

[17] TMForum NFV Management Ecosystem - http://tmflive2014bak.wpengine.com/nfv-management-ecosystem/.

[18] TMForum G1123 NFV Readiness: Packaging Virtualized Network Services for Procurerment and Operations R14.5.1 - September 2014.

[19] TM Forum Information Framework Enhancements to Support ZOOM R15.0.0 - May 2015.

[20] Deliverable D6.1 Service Description Framework - T-NOVA project.

[21] TMForum IG1120 Virtualization Impact on SLA Management - April 2015.

[22] TMForum IG1127 End-to-end Virtualization Management: Impact on E2E Service Assurance and SLA Management for Hybrid Networks R15.0.0 - May 2015.

[23] Deliverable D6.4 - SLAs and Billing - T-NOVA project.

[24] Project XIFI (FI-PPP), “Official Web Site,” [Online]. Available: [Accessed 2014].https://www.fi-xifi.eu/home.html.

[25] MCN project Mobile Cloud Networking - http://www.mobile-cloud-networking.eu/site/.

[26] Unify project - Unifying cloud and carrier networks - https://www.fp7-unify.eu/.

[27] NETIDE project - An integrated development environment - www.netide.eu.

[28] CENX Ethernet Lifecycle Manager (ELM) Available: [Accessed 2014] http://www.cenx.com/.

[29] Equinix Platform – Marketplace. Available: [Accessed 2014] http://www.equinix.com/.

[30] HP OpenNFV Partner Ecosystem - http://www8.hp.com/us/en/cloud/nfv-partners-labs.html.

[31] SDN App Store - http://blogs.hp.com/marketwatch/2014/09/27/hp-launches-industry%e2%80%99s-first-sdn-app-store-unleashing-new-wave-of-networking-innovations/.

[32] Cloud28+ - http://www.cloud28plus.eu/.

[33] Deliverable D8.1 - Market Assesment. T-NOVA project.

[34] V. Krishna, Auction Theory, 2010.

[35] Cyclops - https://icclab.github.io/cyclops/.

[36] Amazon AWS CloudFormation. Available online. [Accessed 2014]. http://aws.amazon.com/es/cloudformation/.

[37] OpenStackHot. Available [Accessed 2014] http://docs.openstack.org/developer/heat/template_guide/hot_spec.html..

[38] ETSI GS NFV 003 "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[39] Deliverable D2.32 - Specification of the Infrastructure Virtualisation, Management and Orchestration - FInal - T-NOVA project.

[40] Deliverable D5.1 - Function Store - T-NOVA project.

[41] Deliverable D3.1 - Orchestrator Interfaces - T-NOVA project.

[42] 5GEx project - 5G Exchange - https://5g-ppp.eu/5gex/.

[43] SONATA project - Service Programing and Orchestration for Virtualized Software Networks - https://5g-ppp.eu/sonata/.

8. LIST OF ACRONYMS

Acronym	Explanation
AA	Authentication and Authorisation
AAA	Authentication, Authorisation, and Accounting
API	Application Programming Interface
BSS	Business Support System
CRUD	Create Read Update and Delete
CPU	Central Processing Unit
DoW	Description of Work
eTOM	Telecom Operations Map
GUI	Graphical User Interface
EMS	Element Management System
ETSI	European Telecommunication Standard Institute
EU	End User
FI	Future Internet
FP	Function Provider
ISG	Industry Specification Group
ISP	Internet Service Provider
IT	Information Technology
KPI	Key Performance Indicator
MANO	Management and Orchestration
NFaaS	Network Functions-as-a-Service
NF	Network Function
NFC	Network Function Component
NFV	Network Functions Virtualisation
NFVI	Network Function Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
NS	Network Service
OSS	Operational Support System
QoS	Quality of Service
RBCA	Role Based Access Control
RTT	Round trip time
SaaS	Software-as-a-Service

SBC	Session Border Controller
SDK	Software Development Kit
SDN	Software-Defined Networking
SDO	Standards Development Organisation
SI	Service Integrator
SID	Shared Information/Data model
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SP	Service Provider
TAM	Telecom Application Map
TIP	TM Forum Integration Program
UC	Use Case
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
VNFaaS	Virtual Network Function as a Service
VNFD	Virtual Network Function Descriptor
VNFM	Virtual Network Function Manager
VNI	Virtual Network Interface
VNPaaS	Virtual Network Platform as a Service
WP	Work Package