25-11-2020

Deliverable D6.6 Transforming Services with Orchestration and Automation

Deliverable D6.6

Contractual Date:	30-11-2020
Actual Date:	25-11-2020
Grant Agreement No.:	856726
Work Package	WP 6
Task Item:	Task 2
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	PSNC
Document ID:	GN4-3-20-38E5D9
Authors:	Daniel Lete (HEAnet), Eduardo Jacob (RedIRIS), Hamzeh Khalili (I2CAT/RedIRIS), Iacovos Ioannou (CyNET), Ivana Golub (PSNC), Jasone Astorga (RedIRIS), Jovana Vuleta-Radoičić (UoB/AMRES), Konstantinos Georgilakis (GRNET), Kostas Stamos (GRNET), Marcin Adamski (PSNC), Maria Isabel Gandia Carriedo (CSUC/RedIRIS), Martin Dunmore (Jisc), Michal Balcerkiewicz (PSNC), Piotr Grabowski (PSNC), Ramesh Baskaran (Jisc), Roman Łapacz (PSNC), Sonja Filiposka (UKIM/MARNet), Susanne Naegele Jackson (FAU/DFN), Tim Chown (Jisc), Yuri Demchenko (UvA/SURFnet)

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).

Abstract

The digital transformation process challenges the traditional approach to network services demanding agility and flexibility in all stages of service management. Using Orchestration, Automation and Virtualisation (OAV) as technology drivers, the activities undertaken by the *Network Services Evolution and Development* task (Task 2) of the *Network Technologies and Services Development* Work Package (WP6) of the GÉANT project are focusing on helping the community tackle service transformation in an interoperable way by proposing a high-level architectural blueprint. The work on the Service Provider Architecture platform showcases a solution that follows the proposed concepts and principles.

Table of Contents

Table of Figures

Executive Summary

Many NRENs are in the process of adopting Orchestration, Automation and Virtualisation (OAV) principles, with some more advanced than others. The survey of their OAV practices and plans conducted among European NRENS by the *Network Technologies and Services Development* work package (WP6) of the GÉANT 4-3 project in 2019 highlighted common rationales for adopting OAV (not least more agile deployment with consistent configurations), but it also showed that each NREN was on a different stage of their OAV journey, and that many had not started at all [D6.2].

Reacting to the survey findings and to support the NRENs' transformation of services using orchestration and automation, WP6 initiated a number of focus groups to explore the foundations of a potentially more collaborative approach, with the establishment of a common terminology and a common reference architecture, and promoting knowledge exchange between the NRENs through an OAV wiki and community portal.

This approach does not mandate that NRENs use specific technologies. Rather, it seeks to lay out a common framework through which NRENs can offer their services at a fast pace. At the same time, using a common vocabulary and reference architecture allows NRENs to share knowledge about the implementation of common functionalities within their own domains, which in turn fosters the potential for future interoperability for multi-domain services.

The WP6 *Network Services Evolution and Development* task (WP6 T2) team has worked diligently to support the initiatives for a common approach and to promote knowledge exchange. For NRENs at a more advanced stage, the team is seeking dialogue with those NRENs to explore how their current infrastructure maps to the reference architecture developed by WP6 T2. This document lays out a proposal for common concepts and principles that describe that reference architecture. This flexible reference model can be used as a high-level blueprint for NRENs, who can further define and adapt the design so that it fits their needs and purposes while remaining as interoperable as possible with other NRENs who are also compatible with the same reference model.

The Service Provider Architecture (SPA) platform developed by the WP6T2 team is a production implementation of the Open Digital Architecture (ODA) architecture and is under-the-hood of the GÉANT Connectivity Service (GCS). It implements the proposed reference model, common concepts and principles, and can be used by NRENs as a platform for further transformation of their services towards autonomous networking using orchestration and automation. Being based on ODA principles, all of its components are reusable, and some of them are being considered to be adopted by other additional services and projects.

1 Introduction

The process of digital transformation brings new challenges to research and education networks, such as a demand for self-service portals, and the availability of real-time information about the status and health of all services. These challenges can only be efficiently and effectively addressed through the use of Orchestration, Automation and Virtualisation (OAV) techniques.

For the GÉANT community, including GÉANT as an organisation, the NRENs and related parties, it is crucial to drive this network evolution forward with a common approach. This should not be too prescriptive, but allow the adoption of OAV for digital services in an interoperable and inclusive fashion, facilitating knowledge exchange between the NRENs and other organisations, and bringing all partners on board while accommodating already existing solutions.

By focusing on a common approach to the development of digital platforms, the community can take advantage of an interoperable architectural blueprint that outlines the minimum set of common goals and principles necessary to ensure agile collaboration and interoperability not just between the GÉANT community and its NRENs, but also with other providers. In this way, a foundation for knowledge exchange, support and learning can be created, starting with the definition of a common terminology. Enabling the NRENs to use the same language and discuss their architectures and approaches using a common reference architectural model allows both commonalities and diversity to be identified. Using these common concepts, each NREN can lay out their path for implementing OAV depending on their level of maturity and goals, while continuing to be aligned with the rest of the community at a higher level.

The *Network Services Evolution and Development* task (T2) team of the GÉANT 4-3 project's *Network Technologies and Services Development* work package (WP6) has undertaken an extensive analysis of existing solutions and approaches to be able to propose a lightweight common approach - a reference architecture or blueprint - that could be adopted by the GÉANT community without imposing any specific restrictions on individual NRENs.

This deliverable aims to lay out the concepts of a common approach that will help NRENs build their interoperable digital platforms and transformation towards autonomous networks and services. For NRENs that are more advanced on the OAV path, the team is exploring how their existing infrastructure maps to the reference architecture. For NRENs at an early stage of the journey, the team is exploring the provision of training "by the community, for the community" in key OAV concepts.

A production implementation of the Open Digital Architecture (ODA) architecture has been developed in the previous GÉANT project in the form of the Service Provider Architecture (SPA) platform [D8.5], which is a part of the GÉANT Connectivity Service (GCS). Further development and enhancement of the SPA platform is continued in GN4-3 WP6 by the T2 team. This modular, digital platform is based on discrete, re-usable, functional building blocks and applies standards, principles and best practices [Principles]. It also enables agile service management towards zero-touch integration, operation and partnering. The SPA platform can easily be adopted for new services that are designed from scratch. In addition, due to its modularity, flexibility, component decoupling, and support for business and operational agility, the SPA can easily be used in service transformation processes with its readily available components and interfaces. The platform has been used to manage the lifecycle of the GÉANT Connectivity Service since the beginning of the project. Through collaboration and regular discussion with WP7, SPA has been extensively used by the GÉANT NOC and tested on a daily basis, which has helped to further validate the approach and adapt it to the requirements of the end users.

This deliverable is structured as follows:

Section 2 outlines an approach to service management transformation , explaining the differing levels of adoption to date of OAV in the NRENs , and presenting the architectural requirements for a flexible blueprint that could be used as a reference for NRENs.

Section 3 presents the Service Provider Architecture (SPA), an architecture to manage network services, its compliance with ODA and the implementation, which is used in production for the GÉANT Connectivity Service (GCS).

Section 4 summarises the work conducted by WP6 T2 and offers conclusions on how NRENs and other organisations can benefit from this work to transform their service management.

2 Service Management Transformation

Networks and their associated services have become commodities to users, who now demand self-service environments where they can make changes at any time. The challenge for NRENs is to transform their existing service management solutions to quickly adapt to the new research and education network needs, while keeping the quality, high capacity and resiliency their users demand. Digital platforms, integrating business and operational supporting systems, can help in offering these services in an agile, automated and flexible way.

2.1 Requirements and Considerations

To evaluate the status of OAV in the community, and establish and maintain consensus on a direction for interoperable network service orchestration and automation, a phase of "consensus building" was conducted. WP6 Task 2 carried out a survey and held several discussions with the NRENs and the international community. The inputs gathered in this phase are summarised in Deliverable D6.2 *Automation and Orchestration of Services in the GÉANT Community* [D6.2].

Based on the results of the study and inputs from the NRENs, several focus groups were created to explore specific topics and their relevance to the community, to discuss and document best practices, and to help promote specific OAV technologies. These focus groups were created with a limited initial lifetime and with an agreed specific output. Focus Groups (FG) were established on OAV Terminology, OAV Architecture, OAV Public Wiki, Campus Network Management as a Service (CNaaS), Data Transfer Node (DTN) infrastructures and OAV Training.

As seen in Deliverable D6.2, the level of OAV adoption, as well as the knowledge around it, varies from NREN to NREN. There is a community need for knowledge sharing, information exchange, and references such as a common terminology and architecture framework, and examples of use cases.

To address the knowledge gap, foster better understanding of the work and raise the bar for OAV across organisations, the group has participated in several community events to spread the word on OAV [NEMMO], [GA], [10SN], [TECHEX], [19STF], [PMW], [GNA-G]. In addition, infoshares have been organised or are being organised at the time of writing, and an OAV training programme is being developed to help create a common ground for knowledge and understanding in the community [CNaaSinfo], [DTNinfo], [OAVinfo].

Another way to foster understanding among the NRENs, and thus help transform service management, was to facilitate the use of a common terminology related to OAV, clarifying terms, providing definitions and creating a knowledge base as a reference, as OAV terms were often being used in different ways within the community. The Terminology FG investigated existing terminology documents from many standardisation bodies and projects [Standards], and found that there was no single document containing all the necessary terms. It, therefore, identified a list of relevant OAV terms and acronyms, and published a Terminology Whitepaper, providing short definitions for these terms which, where possible, are based on standardisation body documents, or on the FG-provided consensus-based definitions resulting from internal discussions and surveys within the team [OAVterm]. The list of terms and abbreviations can also be found on the public OAV wiki [WikiTerm].

Campus Network Management as a Service (CNaaS) was identified as an emerging use case for OAV in several NRENs around Europe, and discussions during the Network Monitoring Workshop [NEMMO], attended by around 70 people, confirmed the high level of interest in the topic from the NRENs. Apart from the network connectivity services the NRENs have been providing to their end institutions for many years, the requirements are now extended to also include management of networking services within and for campuses. Addressing high user demands and, at the same time, providing such services to several campuses, requires not just a high level of organisational capabilities but also orchestration and automation techniques, to obtain the level of scalability, flexibility, agility and speed that cannot be achieved with traditional manual configuration and network management. The CNaaS FG, together with the Wiki FG, created a community portal to share relevant use cases from the community [ComPort]. Together with WP6 Task 3, the team has also produced a set of recommendations for creating service documents for NRENs offering CNaaS services, and has organised an infoshare around the topic with the participation of several NRENs [D6.4], [CNaaS-SD], [CNaaSinfo].

The team has also provided additional resources to the community in order to support and help with service management transformation, such as the OAV Public wiki, the DTN wiki (in collaboration with T1) and the investigation of OAV architectures, which are detailed further in Section 2.2 [OAVwiki], [DTNwiki].

2.2 OAV Architecture Requirements and Considerations

Service management transformation should enable easier integration of the individual digital platforms of different service providers, including NRENs and their partners. To facilitate that, a common reference architecture is needed to be able to better understand individual solutions. This by no means prescribes how a solution should be designed, nor does it refer to individual software components. Rather, it provides the possibility to compare and make different solutions interoperable by understanding their exact role and the set of functionalities they provide in service management. Therefore, the OAV architecture investigation was undertaken to determine common characteristics in the advanced architectural frameworks of different organisations and standards groups, to define a reference service architecture to be used as a blueprint for the creation of new services and workflows, and to help develop solutions that can easily be automated and orchestrated in order to pave the way to facilitate collaboration and the potential for future interoperable multi-domain services.

Services that used to be technology innovations are today considered commodities that are not just expected to be available everywhere at any time, while secure and resilient, but it is also highly desirable that they can be set up quickly with as little intervention from the operators, partners or users as possible. Therefore, the service management model has to transform from the manual intervention of service provider agents using separate operational and business supporting systems, towards a new model of zero-touch service setup and provisioning.

This transformation includes the business processes, service provider architecture and supporting systems. To evaluate the suitability of different existing architecture models as a reference architecture, the WP6 T2 team considered the required features and capabilities that should be present in that architecture.

The resulting list below is not exhaustive or prioritised in any way, but it provided a solid basis for the evaluation, and can potentially be re-used when studying any new emerging architectural approaches. The list of features includes:

- End-to-end service lifecycle management with a standardised approach to abstraction and functionality exposure.
- Open APIs.
- Flexible scalability (scale in/out and up/down).
- Flexible integration with other partners for inter-domain services.
- Allow for new services, resources, devices or demands from the scientific community.
- Modular design based on components, where each component is a building block with a well-defined set of functionalities and interfaces.
- Common information modelling across components.
- Support for all kinds of organisations and service catalogues.
- All components should be secure, especially in terms of interfaces and access to data.
- Able to deal with any type of resource, virtual or physical (resource types may include Internet of Things (IoT) devices, sensors, or science instruments (e.g. radio telescopes, gene sequencers, etc.), licensed mobile spectrum, etc.
- Service discovery.
- Support for multi-domain orchestration.
- Technology- and service-agnostic.
- Compatibility with the standard industry-adopted architectures and frameworks such as the ones promoted by TM Forum, ETSI, IETF, OASIS and also with cloud-based virtual resources.
- Multi-tenancy support.

In the search for an architecture that would satisfy as many of these requirements as possible, different literature was reviewed including:

- Available standards: TM Forum Frameworx [TMFF], MEF Lifecycle Service Orchestration [MSOS16], ETSI Zero touch network & Service Management (ZSM) [EZSM], ETSI Open Source MANO (OSM) [EOSM20], [ETSIMANOA], ETSI Generic Autonomic Networking Architecture (GANA) [GANA16]
- Best practice documents: TMF [IG14]

- Projects: Open Baton [OB20], European Open Science Cloud (EOSC) architecture [EOS20], SDN for End-to-end Networked Science at the Exascale (SENSE) [SEN20], Open Network Automation Platform (ONAP) [ONAP20]
- GÉANT service architectures including the General Virtualisation Model (GVM), SPA and GCS

A careful review of the available literature concluded that the identified requirements and design principles are closely aligned with the approach currently taken by the TM Forum Open Digital Architecture project (ODA) [ODAWP].

The main idea of ODA is to “offer an industry-agreed blueprint, language and set of key design principles to follow” [ODAWP]. The complete set of ODA Concepts and Principles is available in [ODACP].

The ODA principles structure OAV processing through re-usable, discrete functional blocks that use decoupling and integration with open APIs to allow for all blocks to work together in a common architecture. This “building block” approach allows NRENs to have a common understanding of individual system architectures and the digital platforms of other organisations, including NRENs, partners and other third parties. The ODA principles, the building-block approach and its decoupling and integration model allow NRENs to keep their own catalogue and inventory systems, and customer, resource and network services, and only share what they want to be shared through common, open APIs, while still retaining their proprietary processing within their own domain. This strategy of using functional building blocks and open APIs also facilitates a step-by-step adoption of the model, and NRENs, that are at different stages of implementation, may find it easier to progress through transformation by deploying one step at a time and to start by sharing only some select services.

The ODA architecture is aligned with other industry standards such as ETSI NFV and MEF LSO, providing a unique high-level reference view of different possibilities to implement an OAV architecture that follows a defined set of design principles.

With a reference architecture in place, the next step towards a multi-domain, multi-solution environment for the GÉANT community is to map existing solutions to the reference architecture in order to establish a common ground for mutual understanding of individual solutions. Therefore, as a proof-of-concept of this blueprint approach, several different network architectures (from standardisation bodies to projects and use cases) were mapped to the functional blocks of the ODA framework. The outcome of this exercise will be published in a white paper. A number of NRENs’ architectures are also being mapped to this blueprint, to facilitate common language, knowledge sharing (including experiences with deployed OAV solutions and technologies), comparison and to support collaboration with other organisations.

2.3 ODA Principles and Functional Blocks

The selected reference architecture TM Forum Open Digital Architecture (ODA) defines 24 principles in five categories [ODACP]. Some of these are:

- Business architecture is iterative and reusable
- Interoperability

- Catalogue-based architecture
- Information is accessible, quickly available, secure and analysable shared asset
- Technology-agnostic
- Ease of use
- Decoupled modules with capability exposure via APIs
- Stepwise evolution to target
- Control technical diversity

The ODA architecture is represented with six blocks as defined in the ODA Functional Architecture [ODAFA] and presented in Figure 2.1. This representation provides a high-level view of all enterprise functions, and helps understand the responsibilities of each functional block and identify the integration points that are necessary to implement full end-to-end workflows.

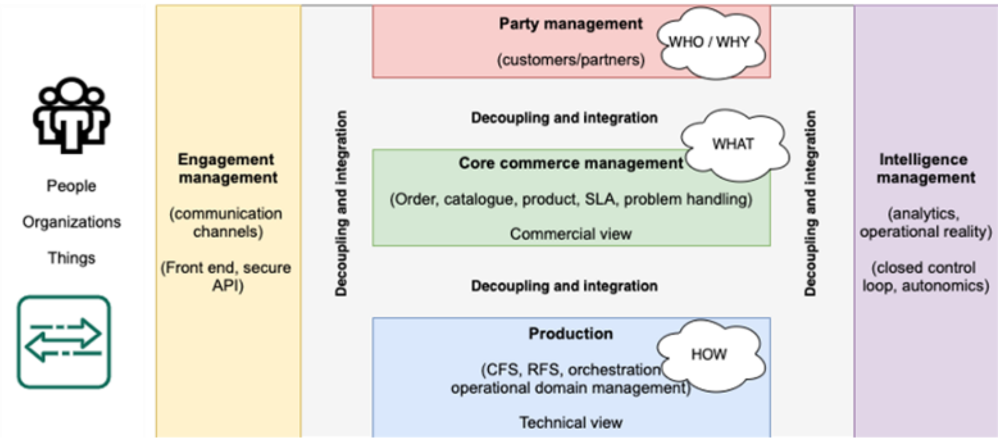

People
Organizations
Things
Engagement management
(communication channels)
(Front end, secure API)
Decoupling and integration
Party management
(customers/partners)
WHO / WHY
Decoupling and integration
WHAT
Core commerce management
(Order, catalogue, product, SLA, problem handling)
Commercial view
Decoupling and integration
Production
HOW
(CFS, RFS, orchestration, operational domain management)
Technical view
Decoupling and integration
Intelligence management
(analytics, operational reality)
(closed control loop, autonomics)

Figure 2.1: ODA Functional block grouping based on [ODAFA]

The **Engagement Management** functional block focuses on the interaction with all internal and external actors. These can be people or software agents, representing customers, employees, partners, third parties, etc. This interaction can be implemented via multiple communication channels. However, the main functionality that needs to be maintained is the omnichannel experience where a user who uses multiple communication channels to interact is always experiencing a consistent journey. The interactions can include providing information, or activating processes and functions that are implemented by components from different functional blocks using the corresponding APIs. It is important to understand that this functional block is a presentation layer only, it does not store any processes, functions or operational data, only technical functions that are needed to provide the right context to the user. In other words, the Engagement Management functional block is responsible for the front-end, authentication and authorisation of users, and management of the user-interaction journeys including content personalisation and filtering. Another important function of this block is the API HUB, which is responsible for exposing a standardised set of APIs to the partners or other external systems.

The **Party Management** functional block manages the internal and external actors of interest, such as persons or organisations (referred to as parties). It focuses on management of the information related to the parties, the party roles and rights, and all related marketing, sales and billing activities.

The **Core Commerce Management** functional block is responsible for all activities related to the provisioning of the main business processes by managing the catalogue of product offers and products, and handling the lifecycle of product orders, from order creation to all of the orchestration actions necessary to fulfil the complete order, including any assurance activities. The main concept of this functional block is that it is decoupled from the technical concerns of product delivery. Technical aspects are not the responsibility of this block. This block may also include product inventory management, problem handling, Service Level Agreement (SLA) management, and other related functions such as product testing or agreement management.

The **Production** functional block is in charge of the complete lifecycle management of the services including the Customer-Facing Services (CFS), and the Resource-Facing Services (RFS) for all technologies that are used in the organisation. Using the exposed APIs of this functional block, the Core Commerce Management can ask for the fulfilment of services that compose a given ordered product, or the Engagement Management block can forward a user request for changing an existing service. The main functionalities of the Production block include all activities related to the end-to-end service and resource lifecycle management, including, in the case of multi-domain service partnering, actions that may span across multiple organisations that are part of the ecosystem. The CFSs are used to describe the supported services in a technology-agnostic way so that they can be used as the basis to build product offers for customers. The RFSs are the technology specific implementation of CFSs, and the Production block manages the lifecycle of all RFSs no matter the technology (including any virtualisation, orchestration and/or automation-related technology) or origin (local or remote, provided by a partner, etc.). In addition to the delivery of services, the Production block is also responsible for service development, infrastructure deployment, operations management, usage and performance management, workforce management, resource provisioning, service and resources catalogues and inventories, etc.

The **Intelligence Management** functional block focuses on the processes related to data analytics using the operational data produced by other functional blocks. This functional block employs data analysis techniques, such as trend analysis, correlation and data aggregation needed for marketing and sales forecasting, but also for network performance evaluation. Because of the large amount of operational data, it would benefit from implementation of big data related capabilities. The insight management capability of this block enables the uncovering of new patterns and relations based on historical data, using popular approaches based on artificial intelligence and machine learning. Another functionality of this block is the Autonomic Manager, which focuses on the implementation of closed control-loops that can configure and then adapt in real time to the state of a resource or a network using cognitive algorithms. In other words, this functional block deals with knowledge management (including the production knowledge plane) activities on different time scales (fast and slow) to implement self-anything (organisation, healing, tuning, etc.) capabilities. Referring back to the set of design principles, it is evident that the Intelligence Management block includes the implementation of intent-based networking capabilities, and, therefore, should also include the specification of high-level policies and objectives. Based on the described characteristics and functionalities of the Production block it can be recognised that this block represents the heart of an NREN's network operations centre.

The **Decoupling and Integration** block manages the separation of the functional blocks so that the boundaries between the sets of related services represented with the functional blocks are respected. The integration part provides a combination of the services, in such a way that a high-level goal is achieved. This functional block ensures that the integration can be implemented in a flexible way without any predefined combination patterns. To achieve this, the functionalities that need to be offered by the decoupling and integration fabric include the management of an API catalogue and related documentation, as well as the ability to perform message routing and API mediation. This enhances the separated components' APIs with policies, security and control.

3 Service Provider Architecture and Platform Implementation

The Service Provider Architecture (SPA) is a representative architectural solution that is completely aligned with the ODA concepts and practices, and it adopts the models and concepts defined in the TM Forum Frameworx [TMFF]. Its design and development started during the GN4-2 project by designing and implementing an initial pilot platform [D8.5]. It is implemented as a platform used by the GÉANT Connectivity Service (GCS), run by the GN4-3 Network Core Infrastructure and Core Service Evolution and Operations work package (WP7). In addition, the SPA solution can be used as the basis for the development of a digital platform for NRENs. It can be used to manage the lifecycle of any type of service and is, therefore, applicable for all services provided in the GÉANT community, including networking services, and the transparent certificate or eduroam services.

SPA is described here both as a design and implementation, which currently manifests itself through the Self-Service Portal (SSP) and is used by the GÉANT NOC, but also as an example of the design and implementation of an ODA-compliant architecture.

Founded on ODA principles, SPA is built from a number of loosely coupled components, each of them providing a well-defined set of functionalities exposed via specific APIs. The functionalities and granularity of the components are based on the functional application description provided in the TM Forum Application Framework (TAM) [TTAM]. Similarly, the data model used in each component is based on the TM Forum Shared Information Data model (SID) [TSID]. To be able to implement service management processes, orchestration of the action is necessary. Orchestration in SPA is based on business processes which should be defined in line with the TM Forum Business Process Framework (BPF, also known as e-TOM) [TBPF]. In addition to the three pillars of Frameworx (BPF, SID and TAM), the APIs exposed by each component are based on the TM Forum Open APIs specification [OAPI].

The SPA architectural model follows an ever-evolving set of best practices and standards that are developed and supported by many major communication service providers around the world. The choice to use Open APIs fosters interoperability with other commercial providers, enabling the use of the same APIs for internal and external use, while building single and/or multi-domain services. Furthermore, the modular ODA building block principles adopted by SPA provide the freedom to NRENs to adopt only the pieces of SPA which are of immediate interest to them. In this way, SPA can be seen as an effort to also contribute to the process of NRENs transforming their traditional OSS/BSS environments towards digital business platforms.

3.1 Design

The SPA implementation is made using the concept of microservices, where the complete digital platform is seen as a collection of services [MSS]. Each architectural component is implemented as a separate self-contained microservice that has its own separate life cycle starting from deployment, to maintainability and testability.

Each of the SPA platform components exposes a RESTful API that can be used for the integration of the decoupled components. All REST APIs follow the same patterns that define the use of the verb commands, filtering, and similar actions in a common way, making it easier to integrate the components. A messaging queue/channel is used for the exchange of information between the components.

The choices for the implementation of the SPA platform components are based on maximum re-use. For each component the most suitable free, open-source implementation is chosen. The choice is made based on the suitability of the information model, the popularity and support of the solution, and its maintainability. Only when a suitable, free, open-source component cannot be found, a component is developed in house by the task development team. This approach not only expedites the platform implementation, but also fosters re-use of existing components among the NRENs, and guarantees stability and performance of the components given their reputation in the open source community. However, choosing an existing non-TMF compliant component requires the development of a wrapper that will expose the correct Open API for the component. In other words, an extra layer needs to be developed that will translate from/to the native component API and the TM Forum Open API defined for the desired functionality.

For orchestration, a free open-source business process management solution was used with an integrated business process engine. The architecture supports a design with one or multiple hierarchical orchestrators. For the initial implementation of SPA, a design with a single process engine is chosen. The processes and subprocesses are hierarchically defined to support high maintainability and extensibility.

3.2 Components and Interfaces

The current SPA platform implementation consists of a number of components and modules with functionalities that span all ODA functional blocks. The mapping of the SPA components to the ODA functional blocks is presented in Figure 3.1.

Engagement Management

There are two SPA components (the SPA Self-Service Portal and the eduGAIN service) that are focused on communication channels with the users and their authentication and, therefore, belong to the Engagement Management functional domain.

- The Self-Service Portal (SSP)

  The Self-Service Portal is the main access point for users. It is a user-friendly Graphical User Interface (GUI) that provides a one-stop-shop functionality for all users of the SPA platform, irrespective of whether they are end-users or administrators that have additional management rights. The SSP is a front-end, i.e. a type of communication channel that builds an environment for the interconnection between the users and the platform. The information that can be seen using the SSP is collected from a number of components that reside in the other functional domains. In other words, the SSP does not contain any logic that is related to the management of services. All user actions made in the SSP trigger API calls to specific components, such as the service catalogue, or activate business processes offered by the orchestrator.

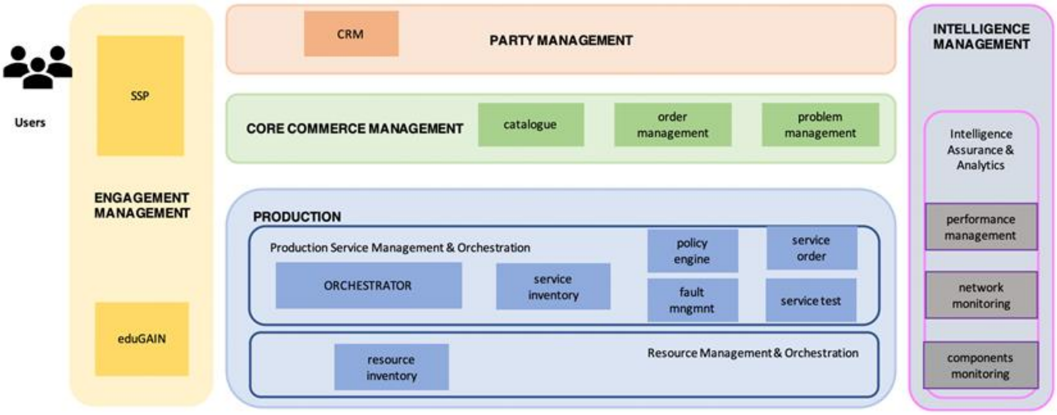
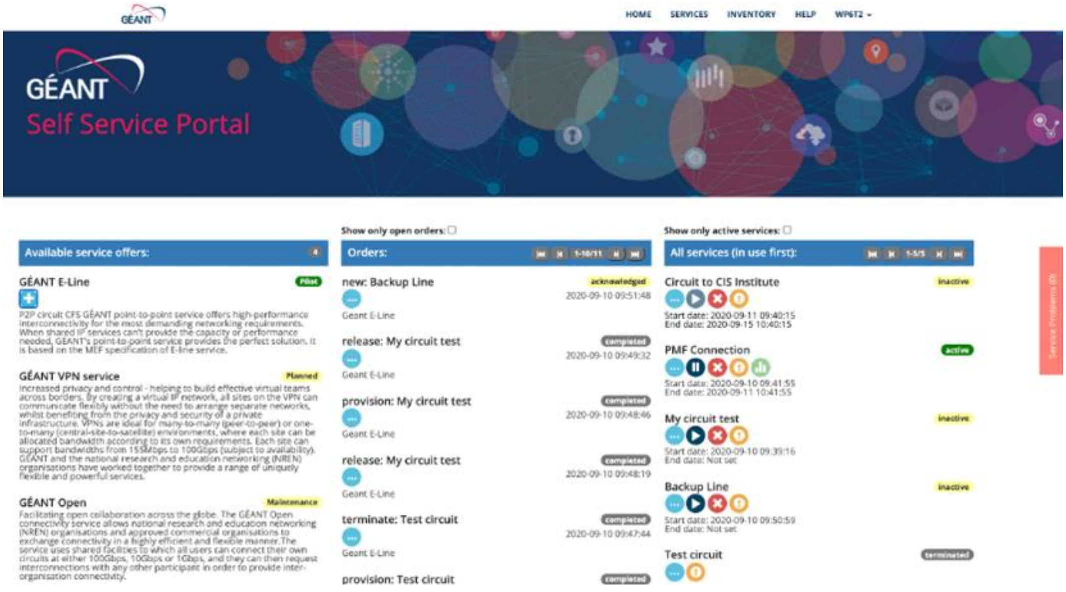


Figure 3.1: Mapping of SPA components to the ODA functional blocks

Currently, there are two different views provided by the SSP, depending on the role of the user that logs in (roles are maintained in Party Management): the user view and the admin view. The user view provides an end-user-oriented view of services that can be ordered and managed on-demand, including tracking of all orders (actions related to services) and information on the status of the service instances and related problems. Figure 3.2 presents a sample screenshot of the user view.



Available service offers: 4

GÉANT E-Line Pilot

P2P circuit CFS GÉANT point-to-point service offers high-performance interconnectivity for the most demanding networking requirements. When shared IP services can't provide the capacity or performance needed, GÉANT's point-to-point service provides the perfect solution. It is based on the MEF specification of E-line service.

GÉANT VPN service Planned

Increased privacy and control - helping to build effective virtual teams across borders. By creating a virtual IP network, all sites on the VPN can communicate flexibly without the need to arrange separate networks, whilst benefiting from the privacy and security of a private infrastructure. VPNs are ideal for many-to-many (peer-to-peer) or one-to-many (central-site-to-satellite) environments, where each site can be allocated bandwidth according to its own requirements. Each site can support bandwidths from 155Mbps to 100Gbps (subject to availability). GÉANT and the national research and education networking (NREN) organisations have worked together to provide a range of uniquely flexible and powerful services.

GÉANT Open Maintenance

Facilitating open collaboration across the globe. The GÉANT Open connectivity service allows national research and education networking (NREN) organisations and approved commercial organisations to exchange connectivity in a highly efficient and flexible manner.The service uses shared facilities to which all users can connect their own circuits at either 100Gbps, 10Gbps or 1Gbps, and they can then request interconnections with any other participant in order to provide inter-organisation connectivity.

Show only open orders: □

Orders: 1-10/11

Order	Service	Status	Time
new: Backup Line	Geant E-Line	acknowledged	2020-09-10 09:51:48
release: My circuit test	Geant E-Line	completed	2020-09-10 09:49:32
provision: My circuit test	Geant E-Line	completed	2020-09-10 09:48:46
release: My circuit test	Geant E-Line	completed	2020-09-10 09:48:19
terminate: Test circuit	Geant E-Line	completed	2020-09-10 09:47:44
provision: Test circuit		completed	

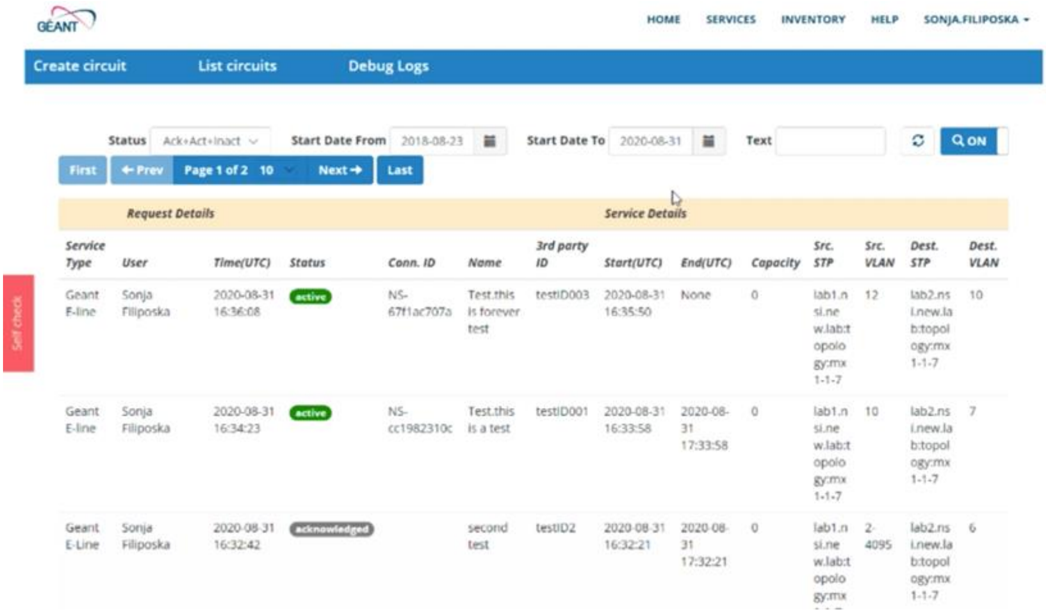
Show only active services: □

All services (in use first): 1-5/5

Service	Status	Start date	End date
Circuit to CIS Institute	inactive	2020-09-11 09:40:15	2020-09-15 10:40:15
PMF Connection	active	2020-09-10 09:41:55	2020-09-11 10:41:55
My circuit test	inactive	2020-09-10 09:39:16	Not set
Backup Line	inactive	2020-09-10 09:50:59	Not set
Test circuit	terminated		

Figure 3.2: Sample screenshot of the SPA user view

The SSP admin view provides an administrative view of the SSP functionalities that is oriented towards service administrators. Figure 3.3 presents a sample screenshot of the SSP admin view.



Status: Ack+Act+Inact
Start Date From: 2018-08-23
Start Date To: 2020-08-31
Text:

Request Details						Service Details							
Service Type	User	Time(UTC)	Status	Conn. ID	Name	3rd party ID	Start(UTC)	End(UTC)	Capacity	Src. STP	Src. VLAN	Dest. STP	Dest. VLAN
Geant E-line	Sonja Filiposka	2020-08-31 16:36:08	active	NS-67f1ac707a	Test.this is forever test	testID003	2020-08-31 16:35:50	None	0	lab1.nsi.new.lab:topology:mx 1-1-7	12	lab2.nsi.new.lab:topology:mx 1-1-7	10
Geant E-line	Sonja Filiposka	2020-08-31 16:34:23	active	NS-cc1982310c	Test.this is a test	testID001	2020-08-31 16:33:58	2020-08-31 17:33:58	0	lab1.nsi.new.lab:topology:mx 1-1-7	10	lab2.nsi.new.lab:topology:mx 1-1-7	7
Geant E-Line	Sonja Filiposka	2020-08-31 16:32:42	acknowledged		second test	testID2	2020-08-31 16:32:21	2020-08-31 17:32:21	0	lab1.nsi.new.lab:topology:mx	2-4095	lab2.nsi.new.lab:topology:mx 1-1-7	6

Figure 3.3: Sample screenshot of the SPA admin view

- eduGAIN service

  User authentication for the SSP is done using eduGAIN [EDUG]. A user trying to access the SSP web address is automatically redirected to the eduGAIN authentication pages, where the user can authenticate using their institutional credentials. The credentials are then compared to the information in the SPA CRM component, and if the user is successfully authenticated, they are then authorised with privileges for the SSP view that corresponds to the user role defined in the Party Management domain.

Party Management

The Party Management functional domain implemented in the SPA platform focuses on the management of the information about different actors related to the use and management of provided services. It is a single source of information about customers, owners of services and resources, partner organisations, admin accounts and similar information, including their roles and privileges. Other typical functionalities of this functional domain such as billing are out of scope for the SPA platform implementation.

- Customer Relationship Management (CRM) component

  The SPA Customer Relationship Management (CRM) component is located within the Party Management functional block. Initially a repository for customer information, this component has been enhanced and now serves multiple purposes. It contains information about all the customers and their organisations, as well as any relationships between the organisations. In addition to customers, it also stores information about any other types of parties, such as admin users. In other words, the component is a repository for all types of party information and their inter-relationships. Additionally, the component also stores the roles and permissions of all parties, which are then used to determine the SSP view and the functionalities of the SPA platform that can be used by the given party.

  The implementation of the CRM component in the SPA platform is done using the free open-source package SuiteCRM [SCRM]. The native API provided by SuiteCRM is wrapped with the TM Forum Open APIs for customer management and party management.

Core Commerce Management

The Core Commerce Management functional domain is treated as the central point for the catalogue, ordering and problem management in the SPA platform implementation. This is the main information that is presented to customers regarding their services provided by the SPA platform. Each of these functionalities is implemented as a separate component.

- Catalogue

  The catalogue component in the SPA platform includes the product and service catalogue. It is implemented using the ITSM extension module of OTRS [OTRS] with some small customisations regarding the categories and relationships between objects. In the catalogue, a one-to-one relationship is defined between a product and a customer facing service (CFS). Thus, each CFS is provided as a potential product to the customer. One CFS, which is the technology-agnostic representation of a service, can be implemented using one or more resource-facing services (RFSs) that are a technology-specific implementation of a service element. The catalogue stores the relationship between CFSs and RFSs. Each of the services is described using a set of service parameters for which a schema is defined and then used correspondingly in the service inventory.

- Order management component

  The order management component is implemented in the form of a ticket tracking system, using the OTRS standard ticketing functionality. Each user action or request regarding the lifecycle of a service is stored in OTRS as a ticket (or a hierarchical set of tickets). During the implementation of the action, which is done using the orchestrator and the activation of a corresponding business process, the status of the ticket is continuously updated, which enables the SSP to show updated information to the customer. Once the action is completed, the ticket is closed and any related tickets, such as a parent ticket, are updated. In the case of unsuccessful actions, the ticket also stores the error messages of the system.

- Problem management component

  The problem management component is used to centrally manage service problems related to customer experience. Similar to the order management component, the problems are stored as tickets in OTRS, in a separate service problems queue. Service problems are raised by the customer using the SSP's functions for reporting a problem with a service. The created ticket can then be reviewed by the service admins and a corresponding action may be taken. The ticket can also be correlated with any tickets with service technical problems that may be raised by the service's monitoring systems. The customer can track the status of the ticket and have updated information on how their problem is being handled by the service provider.

Production Management

The main functional domain of the architecture is the Production Management functional domain, which deals with all functions related to the network service management lifecycle. This domain contains a set of service-agnostic components that deal with the network services in an abstract, technology-agnostic way. The implementation of the corresponding actions on the network element is performed via the TMF-compliant communication interface of the SPA platform with one or more technical domains. Each of the production management functionalities is implemented as a separate component.

- Service inventory

  The service inventory component is a single point of truth about the statuses of all service instances, including CFSs and RFSs. The service parameters are stored for each service in the inventory including status, related order, customer, relationship to underlying resources, and a set of service-specific parameters, i.e. service characteristics. Due to the lack of a free open-source solution that has the flexibility to store the fixed and changing service characteristics in an easily manageable way, the service inventory component is implemented in-house by the development team as a PostgreSQL [PSQL] DB with a logical layer and an Open TMF-based API.

- Resource inventory

  The resource inventory stores information about all resources available in the technical domain managed by the SPA platform. This includes all types of resources, both physical and virtual, as well as any potential abstraction layers over these resources, and their hierarchical relationships. This component is implemented as an in-house developed PostgreSQL database with a logical layer and a corresponding TMF-based Open API.

- Service ordering component

  The service ordering component is implemented together with the order management component from the Core Commerce functional domain. While the main customer actions create tickets that track the implementation of the action, for each RFS related to the CFS in question, a separate child ticket is created and tracked independently. The main order ticket status is changed based on the status change of all related child service orders for the RFS services in question.

- Policy management component

  The main goal of the policy management component is to provide information for each service or each action under which circumstances the service can be used and/or the action executed. The component is implemented as a rule engine with a set of rules defined for a service and/or action. The rules can be based on information that resides in any of the SPA components. An example set of rules could be: an end-user cannot have more than three active instances of a given service unless the user is the manager for a specific NREN. The policy engine is implemented using the free open-source Drools rule engine [DRE].

- Fault management component

  The fault management component is responsible for tracking information about a technical problem with a service that has been reported by service monitoring or a similar system. The information is stored as a special type of ticket in OTRS. If such a ticket exists for a given user service, then the user can be informed via the SSP. The ticket state can be updated via the monitoring systems or the network admins that have resolved the problem. These tickets can be correlated with the problem management tickets if necessary.

- Service testing component

  The service testing component allows any provided service and its related management functionalities to be tested. Using this component, regularly scheduled tests can be run for a given service. Depending on the settings, the results of the tests can be sent asynchronously to the user via another communication channel such as email or Slack messaging. In the case of a failed test, the scheduled tests are, as required by the GÉANT NOC, automatically suspended until the problem is solved. The component also allows test reports for a specified time period to be provided. The implementation of this component is in the form of Selenium automated tests [SEL], and additional logic for the activation and storing of reported data.

Intelligence Management

The Intelligence Management functional domain focuses on big data analytics and any AI-related functions that can help further automation and implementation of autonomous networks that perform self-organisation and self-healing functions. The SPA platform interfaces with external components focusing on network service monitoring and service performance management. It has a separate monitoring implementation for its internal components based on Nagios [NAG] and Telegraf [TEL].

Decoupling and Integration

The decoupled components are accessible via APIs that expose their capabilities. In other words, all functions that are implemented by the SPA components can be activated using API calls.

All exposed APIs are REST based, meaning that the component capabilities are implemented using HTTP common verbs such as GET, POST, PATCH, DELETE that provide create, read, update and delete (CRUD) functions. To maintain a common way of addressing more complex API calls that include filtering, sorting or pagination, common API design patterns are used across all components based on the open REST API design guidelines [APID]. Figure 3.4 presents the TMF Open APIs that are exposed by the SPA platform components. As can be seen in the figure, almost all the components from the internal functional domains are exposed via Open APIs. When the same component is used to expose multiple capabilities, a separate Open API is used for each of the component capabilities. Such an example is the OTRS ticketing being used for ordering and problem management.

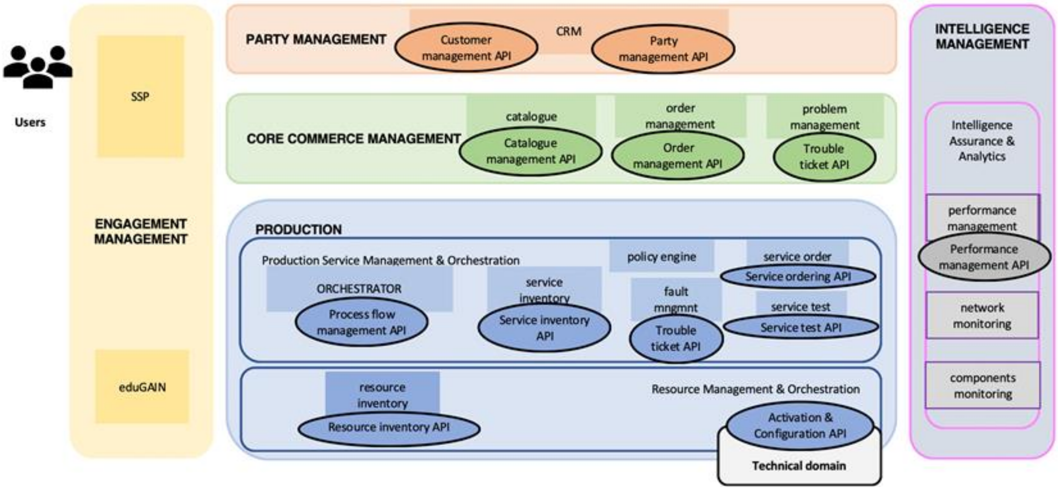


Figure 3.4: TM Forum Open APIs exposed by corresponding SPA components

Only the in-house developed component capabilities are exposed using Open APIs out of the box. The exposure of the rest of the component interfaces with Open APIs is done by wrapping the native component API with an Open API. For these purposes, the SPA platform uses Camel connectors [ACM]. Each Camel connector is a logical layer that provides the mapping between a component native API and the desired Open API. In this case, all Open API end-points are actually Camel connector end-points that are then transformed and routed as native API calls to the components.

3.2.2 Automation and Orchestration in SPA

Orchestration in the SPA platform is done at the business process level. The SPA platform implements an open source business process management suite called Activiti [ABPM]. The orchestration action that spans multiple components is developed as a business process within Activiti. The Activiti process engine is tasked to drive the defined processes triggered with an API call. Processes orchestrated in the SPA platform implementation are new service setup, existing service change and a service termination based on the e-TOM end to end customer-centric process flows for Request to Answer, Order to Payment, Request to Change and Termination to Confirmation. An example ordering process definition is presented in Figure 3.5. All business processes are designed in a hierarchical fashion, with high-level processes comprising a number of sub-processes. This allows for easier maintenance and focused changes in the low-level processes without disturbing the higher order logic of events.

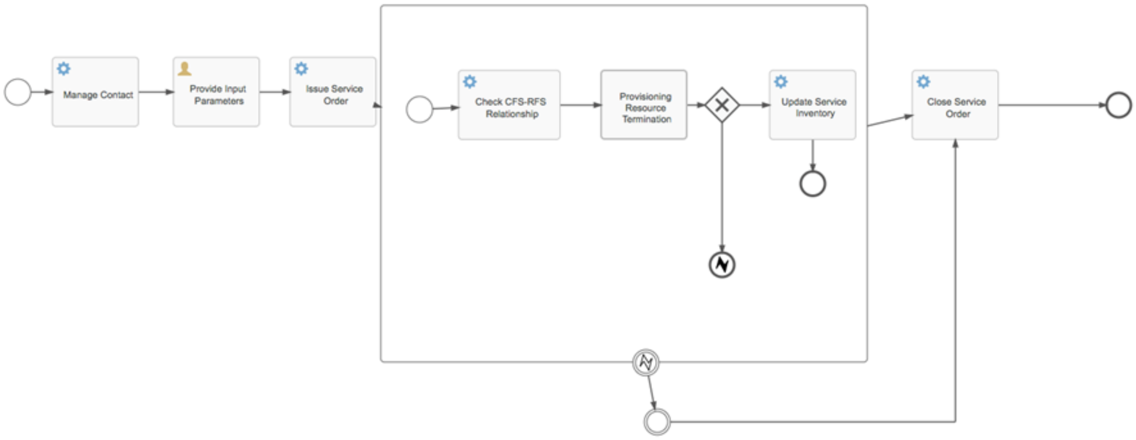


Figure 3.5: Graphical representation of the definition of the service termination business process

The complete SPA platform is designed as a fully automated solution. All processes currently defined in Activiti are automated and require no manual intervention. However, if necessary, the orchestrator also supports definition of semi-automated processes that require a human intervention in some phase of the process. The manual action can be a simple review and acknowledgement, or a more complex decision on the next phase that should be executed.

The Flow open API is proposed as the API that exposes the capabilities of the orchestrator. By exposing this API to the external world, the SPA platform can provide its capabilities to other processes or machines that do not need to use the SSP to be able to trigger a service management action.

3.3 Network Service Transformation Using SPA Platform Components

In contrast to the traditional approaches to network service development where network services and related processes were developed in a siloed fashion, one of the main principles of the SPA architecture and the platform is that the components are service-agnostic. They can be reused and configured for building new service management solutions addressing emerging requirements.

Currently, the SPA platform implementation is mainly used as a service management platform for the GÉANT Connection Service (GCS). WP6 and WP7 collaborate closely in terms of agreeing and defining additional features for implementation. The development team is designing, analysing and developing additional specifications and capabilities that will enable the use of SPA platform components for GÉANT IP service configuration automation and, potentially, for additional services. Some of the components have raised interest for re-use outside of GÉANT, which is the case with the SyMEC project that focuses on edge computing management [SYMEC].

3.3.1 GÉANT Connection Service

The SPA platform implementation is used by the GÉANT NOC to manage the service lifecycle of the GCS. The SSP admin view has been specifically developed according to the NOC requirements and specification which enables NOC engineers to have a real-time view of the status of all service instances and all actions related to the service instances.

The integration with the technical implementation of GCS, which is based on OpenNSA agents and a central management console that logs the OpenNSA-related events, is done via the Activation and Configuration API that translates the actions triggered in the SPA platform to configuration changes that need to be pushed to the network via OpenNSA. All feedback from the technical domain is propagated back to the SPA components and presented to the NOC admins. Thus, the service inventory stores information about GCS instances, the resource inventory stores information about all OpenNSA service termination points and their capabilities, while the order management component tracks the success of all actions triggered in the SSP that activate the corresponding processes in the orchestrator.

3.3.2 MD-VPN

Due to the reusability, standardised APIs and information model of the implementation of the SPA service and resource inventory functionalities, it was decided to collaborate with WP7 to explore the possibility of using this implementation as the main inventory of the MD-VPN service or as a supporting component for the already existing one, i.e. the MD-VPN Service Inventory (SI). A strong argument for using the SPA inventory implementation was the ability to gradually add in future SPA components and transform the MD-VPN service into an automated and business-process-based solution.

A comparative analysis has been done between the MD-VPN SI software developed in the earlier phases of the GÉANT project and the SPA platform components in order to determine if it would be possible to store the MD-VPN links information in the SPA inventory.

The analysis has resulted in the extension of the SPA inventory component with the service and resource characteristics specific for MD-VPN, so that it can be used for storing MD-VPN information. Some of the data stored in the MD-VPN SI has been identified as party (stakeholder) information data and location data, and for the purposes of storing this information in the SPA platform, an extension to the SPA CRM component is being created.

3.3.3 GÉANT IP

The GÉANT IP service provides general purpose IP transit for the NRENs and related organisations, enabling the flow of IP traffic for the R&E community. The GÉANT organisation is looking into automating the provisioning of this service and implementing the feature that can help streamline the process of service management. To this end, WP7 T1 is collaborating with the SPA development team on using the SPA service and resource inventory components to describe the GÉANT IP service and its underlying resources. The information stored in the inventories will be used to fill in the templates for network device configurations built by SaltStack, which serves as an infrastructure automation engine [STS].

3.3.4 SyMEC

PSNC is looking to use the SPA platform components within a Polish research project called SyMEC, which aims to design and implement a Multi-Access Edge Computing (MEC) system composed of three main elements: a specialised MEC server hardware, a MEC service orchestrator and a MEC service/application repository [SYMEC]. In general, the system will allow new services to be offered and existing services to be improved in the 3G/4G/5G telecommunication network by moving the services from the cloud to the edge of the network near the end user.

The activities are directed towards integrating SPA platform components with the SyMEC system with the use of TM Forum Open APIs. The WP6 T2 team is providing support to the SyMEC partners in the tasks that are required to deploy, adjust and configure the SPA platform to manage MEC services offered within the SyMEC system. This includes the preparation of a customised self-service portal and assistance in integration tests with the MEC service orchestrator and the MEC technical domains via the TM Forum Activation and Configuration API.

4 Conclusions

Service providers worldwide are transforming the management of their services to meet the growing demands from their users. Users consider telecommunication services to be a commodity rather than a technological advancement, and expect simple-to-use self-service portals and apps that provide fast, agile, relevant and customised responses to their demands. This presents substantial challenges to service providers, and their initial response has been in the adaptation of their operational and business supporting systems, transforming them into digital platforms.

Aware of these transformations, which can also be observed in the GÉANT community, the *Network Services Evolution and Development* task (T2) of the GN 4-3 *Network Technologies and Services Development* work package (WP6) has engaged with GÉANT NRENs to understand their current status, activities and plans in the areas of Orchestration, Automation and Virtualisation (OAV), which are recognised as the main drivers of the transformation. This consensus-building engagement led to the creation of several focus groups through which relevant follow-up actions are being undertaken, and support is offered to NRENs through reference documents, practical implementations, consultancy and expertise.

As a part of this work, a reference OAV Terminology document has been published, comprising a collection of OAV terms from a selection of standards, best practice documents and other sources, which is published as a wiki page and a stand-alone white paper.

Aiming to promote a common approach to the implementation of new-generation digital platforms by the NRENs and other GÉANT community organisations, the OAV architecture focus group has analysed a variety of different available architectures. This analysis identified the TM Forum's Open Digital Architecture as the most appropriate reference architecture for promoting a common high-level set of concepts and principles to help guide the community through its process of digital transformation through OAV implementation, while promoting collaboration between NRENs and the potential of future interoperable multi-domain services.

Automation and orchestration have been closely followed in individual use cases in the GÉANT community, particularly those related to the emerging strong interest in Campus Network Management as a Service (CNaaS) that is now being provided by several NRENs. Such services are an excellent example of how NRENs have been motivated to speed up their service management transformation towards zero-wait, zero-touch, self-configuring agile and flexible services. Support to NRENs has been provided through the organisation of several knowledge sharing events as well as a community portal at the OAV wiki where sources of various NREN use cases and implementation examples have been gathered to be used and adapted as appropriate.

In parallel, the T2 software development team has continued the development of the Service Provider Architecture (SPA) initially started in GN4-2, and, in collaboration with WP7 and the GÉANT NOC, created a Self-Service Portal (SSP) built on SPA that is used to manage the production GCS. The SPA platform implements the vision, principles, guidance and best practices promoted by ODA. Thus, SPA not only promotes the adoption of ODA as the common reference architecture but can be drawn upon by NRENs and any other organisation to help and support their transition towards digital platforms and services.

References

[10SN] https://wiki.geant.org/display/SIGNOC/10th+SIG-NOC+meeting

[19STF] https://wiki.geant.org/pages/viewpage.action?spaceKey=APM&title=19th+STF+-+Zagreb%2C+March+2020 (accessible to project participants only)

[ABPM] https://www.activiti.org/

[ACM] https://camel.apache.org/

[APID] https://www.tmforum.org/resources/how-to-guide/tmf630-api-design-guidelines-4-0/

[CNaaS-SD] https://www.geant.org/Resources/Documents/Campus_Network_Management_Service_Definition_Template_Checklist.pdf

[CNaaSinfo] https://events.geant.org/e/cnaas/

[ComPort] https://wiki.geant.org/display/OAV/OAV+Community+Portal

[D6.2] https://www.geant.org/Projects/GEANT_Project_GN4-3/GN43_deliverables/D6-2_Automation-and-Orchestration-of-Services-in-the-GEANT-Community.pdf

[D6.4] https://www.geant.org/Projects/GEANT_Project_GN4-3/GN43_deliverables/D6.4_Campus_Network_Management.pdf

[D8.5] https://www.geant.org/Projects/GEANT_Project_GN4/deliverables/D8.5-SPA_Pilot_Service.pdf

[DRE] https://www.drools.org/

[DTNinfo] https://events.geant.org/e/DTN/

[DTNwiki] https://wiki.geant.org/display/DTN/

[EDUG] https://edugain.org/

[EOS20] https://ec.europa.eu/research/openscience/index.cfm?pg=open-science-cloud

[EOSM20] https://osm.etsi.org/

[ETSIMANOA] https://www.whitestack.com/products/whitenfv/

[EZSM] https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf

[GA] https://wiki.geant.org/pages/viewpage.action?pageId=123775545 (accessible to project participants only)

[GANA16] https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp16_gana_Ed1_20161011.pdf

[GNA-G] https://video.nordu.net/media/Orchestration%2C+Automation+and+Virtualisation+in+G%C3%89ANT+++-+Pre-recording+for+the+Q3+2020+Community+VCs/0_lytwfx52

[IG14] https://books.google.co.uk/books/about/Information_Governance.html?id=1hURAwAAQBAJ&redir_esc=y

[MSOS16] https://www.mef.net/wp-content/uploads/2016/03/MEF-55.pdf

[MSS] S. Newman, "*Building Microservices: Designing Fine-Grained Systems*" O'Reilly Media, Inc., 2 Feb 2015

[NAG] https://www.nagios.org/

[NEMMO] https://wiki.geant.org/display/PUB/Workshop+on+Network+Management+and+Monitoring

[OAPI] https://www.tmforum.org/open-apis/

[OAVinfo] https://events.geant.org/e/OAV/

[OAVterm] https://wiki.geant.org/download/attachments/138747522/Orchestration%2C%20Automation%20and%20Virtualisation%20Terminology.pdf?version=1&modificationDate=1587389752575&api=v2

[OAVwiki] https://wiki.geant.org/display/OAV/

[OB20] https://openbaton.github.io/

[ODACP] https://www.tmforum.org/resources/reference/gb998-open-digital-architecture-oda-concepts-principles-r19-0-0/

[ODAFA] https://www.tmforum.org/resources/exploratory-report/ig1167-oda-functional-architecture-v5-0/

[ODAWP] https://www.tmforum.org/resources/whitepapers/open-digital-architecture/

[ONAP20] https://www.onap.org/

[OTRS] https://otrs.com/

[PMW] https://wiki.geant.org/display/PUB/Performance+Management+Workshop

[PSQL] https://www.postgresql.org/

[Principles] https://www.tmforum.org/resources/reference/gb998-open-digital-architecture-oda-concepts-principles-r19-0-0/link

[SCRM] https://suitecrm.com/

[SEL] https://www.selenium.dev/

[SEN20] http://sense.es.net

[Standards] https://www.axelos.com/news/blogs/may-2019/itil-4-connecting-key-concepts-part-4
https://purplegriffon.com/downloads/resources/itil4-foundation-glossary-january-2019.pdf
https://www.tmforum.org/resources/reference/tmf071-open-digital-framework-terminology-v8-0-0/
https://indico.cern.ch/event/810635/contributions/3593247/attachments/1927128/3223214/HEPiX_Network_Virtualisation_Working_Group_Update_Oct_2019_HEPiX.pdf
https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf
https://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf

[STS] https://www.saltstack.com/

[SYMEC] https://www.symec.com.pl/pl/

[TBPF] https://www.tmforum.org/resources/standard/gb921-business-process-framework-processes-v20-0/

[TECHEX] https://meetings.internet2.edu/2019-technology-exchange/detail/10005496/

[TEL]	https://www.influxdata.com/time-series-platform/telegraf/
[TMFF]	https://www.tmforum.org/download-frameworx/
[TSID]	https://www.tmforum.org/resources/suite/gb922-information-framework-models-suite/
[TTAM]	https://www.tmforum.org/resources/standard/gb929d-application-framework-v20-0/
[WikiTerm]	https://wiki.geant.org/display/OAV/OAV+Terminology+and+Glossary

Glossary

API	Application Programming Interface
BPF	Business Process Framework
BPM	Business Process Management
CFS	Customer Facing Service
CNaaS	Campus Network as a Service
CRM	Customer Relationship Management
CRUD	Create, Read, Update, Delete
FG	Focus Group
GCS	GÉANT Connectivity Service
GTS	GÉANT Testbed Services
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
IoT	Internet of Things
ITSM	IT Service Management
MEC	Multi-Access Edge Computing
NREN	National Research and Education Network
NSI	Network Service Interface
OAV	Orchestration, Automation and Virtualisation
ODA	Open Digital Architecture
OTRS	Open Ticket Request System
REST	REpresentational State Transfer
RFS	Resource Facing Service
SID	Shared Information Data
SLA	Service Level Agreement
SPA	Service Provider Architecture
SSP	Self-Service Portal
SyMEC	System MEC
TAM	TM Forum Application Mapping