

17-01-2018

Deliverable D8.5

Service Provider Architecture Pilot v1.0

Deliverable D8.5

Contractual Date:	28-12-2017
Actual Date:	17-01-2018
Grant Agreement No.:	731122
Work Package/Activity:	8/JRA2
Task Item:	Task 2
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	MARnet/UKIM
Document ID:	GN4-2-17-25C29B
Authors:	Sonja Filiposka (MARnet/UKIM), Roman Lapacz (PSNC), Michal Balcerkiewicz (PSNC), Piotr Grabowski (PSNC), Marcin Adamski (PSNC), Jakub Gutkowski (PSNC), Afrodite Sevasti (GRnet), Jerry Sobieski (NORDUnet)

© GÉANT Limited on behalf of the GN4-2 project.

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

Abstract

This document describes the implementation and organization of the Service Provider Architecture pilot. The first part of the document describes the Service Provider Architecture in terms of components, interfaces and processes chosen, defined and implemented. The second part of the document focuses on the development pilot planning and execution scheduled to start beginning of January 2018.

Table of Contents

Executive Summary	1
1 Introduction	3
2 Background: Role of OSS/BSS	3
3 Service Provider Architecture	5
3.1 General Design Principles	5
3.2 Standardisation	6
4 Pilot SPA Implementation	7
4.1 Components and Mapping	9
4.2 Pilot Organisation	12
4.3 Pilot Outcomes	14
4.4 Pilot Extensions	15
5 Conclusions	16
References	17
Glossary	18

Table of Figures

Figure 2.1: Pluggable OSS/BSS architecture	5
Figure 4.1: Pilot SPA components v1.0	7
Figure 4.2: Service offer	11
Figure 4.3 Service ordering	12
Figure 4.4: Service termination	12

Table of Tables

Table 4.1: TAM components mapping to current operations and SPA	9
Table 4.2: TMF OpenAPIs used in SPA	10
Table 4.3: eTOM business processes mapping to SPA	11

Executive Summary

The Service Provider Architecture (SPA) pilot, version 1.0, will be run to test the high-level blueprint design made during the first period of the GN4-2 project. Formulation of the SPA blueprint has been based on GÉANT operations requirements, best practices, and standards published by the TMForum. It aims to provide a new generation OSS/BSS solution that is lean, flexible, easily extended, and able to support a multiple-user-facing service in a consolidated fashion. The SPA design and development work presented in this report has been made within GN4-2's Network Services Development Joint Research Activity (JRA2) SPA Task (Task 2).

The first part of this document provides an overview of the high-level design of SPA based on the combination of several software architecture paradigms. This includes the service-oriented architecture, especially microservices implementation, event-driven architecture and responses to external events in the network environment, in combination with business process management and standardised service bus messages.

Based on the high-level design of the pilot description (provided in the second part of the document), the task team has chosen an initial set of components and APIs for implementation, as well as the main business processes that are to be used to orchestrate the distributed components.

This report provides the overall architecture of the pilot, the two phases in which the pilot is divided (migration and implementation) and the objectives that the pilot aims to achieve.

As a development pilot with up to ten alpha users, it aims to evaluate the effort necessary to migrate a service to SPA. The first phase of the pilot will migrate the existing consolidated connection service (CCS) software suite to function within the SPA environment. The first phase pilot objective is to evaluate and document the scope of effort to port a typical service into the SPA environment.

Once the first phase of the pilot is finished, a second phase will ensue that will help determine where the service will be offered, testing the functionalities provided via a business-process driven, self-service portal on a selected user group and including:

- Browsing the service catalogue and obtaining information about the service.
- Ordering a new service instance.
- Tracking the order completion in real-time, obtaining information about all past or current active services.
- Terminating an active service that is no longer needed.
- The users will test and check the functionalities of the pilot based on pre-defined test scenarios and user stories.

The second aim of pilot is to analyse the performances of the designed system in terms of defined OSS/BSS metrics, focusing on user experience and overall system performance metrics relating to software quality.

The outcomes of deploying and running the initial SPA pilot will be published to update this document, and will be used as input for the next design and development phase of the future versions of SPA. The results will feed back into the SPA roadmap to revise the current roadmap for SPA development and define the possible necessity for solution refactoring or changes in the cornerstones of the architecture. It is important that the performances of this initial pilot are thoroughly examined so that any possible setbacks are addressed as early as possible.

1 Introduction

Section 2 of this document includes background details of an OSS/BSS solution. Section 3 provides an overview of the high-level design of SPA based on the combination of several software architecture paradigms. This includes the service oriented architecture, especially microservices implementation, event-driven architecture and responses to external events in the network environment, in combination with business process management and standardised service bus messages.

Based on the high-level design of the pilot description (provided in Section 4 of the document), the task team has chosen an initial set of components and APIs for implementation, as well as the main business processes that are to be used to orchestrate the distributed components.

Finally, the conclusions section sets out the next steps and follow-up of the pilot.

2 Background: Role of OSS/BSS

With the implementation of disruptive technologies in networking, such as network function virtualisation and software defined networks, a holistic organisational and operational transformation is needed to address new approaches to strategy and innovation, user centricity, digital operations and agility, and the ability to engage in new, multi-sided and platform-based business models [[CCA](#)].

The IT system solutions, typically referred to as Operations and Business Support Systems (OSS/BSS), are key to implementing and taking advantage of new opportunities at scale. The process of implementation and/or transformation of these systems is not easy, and as more service providers try to cope with the challenges of overhauling their OSS/BSS systems, a new approach is needed to design the new generation of the OSS/BSS systems. This new generation of solutions should be based on state of the art IT technology, such as Service Oriented Architectures (SOA) and microservice-based approaches, open source components, virtualisation and open APIs.

Operations support systems (OSS) [[BOSSF](#)] as a whole includes the systems used to support the daily operations of the service provider. These incorporate business support systems (BSS), such as billing and customer management and service operations (service provisioning and management), situated on the network-wide view. OSS/BSS is built on top of the different, usually vendor locked, element and network management systems (EMSs and NMSs) consolidating the information from these systems into single points of access, such as a resource inventory. In this layered management approach, BSS corresponds to business management and OSS corresponds to service management, while the other management layers include network management and element management. The major goals of the layered approach are to increase the overall system performance, availability and productivity, but to also reduce the time needed to develop new services. In this way, the complex

management of operations activities (including service activation, service assurance, customer services, service monitoring and others) can be reduced into an automated set of steps managed by different system components.

The user access to the overall OSS/BSS system is through a self-service portal [[GMOSS](#)] that will act as a one-stop-shop for browsing, ordering, managing and terminating services. In this way, full two-way communication can be realised between the user and the network, where all user requests are automatically translated into processes and workflows that trigger the necessary network changes, and, reversely, all user (service) relevant network events are presented to the user so that s/he is continuously informed about the status of his orders and services.

The Service Provider Architecture (SPA) developed in the JRA2 Network Services Development, Task 2, is striving to define a state-of-the art software architecture for an OSS/BSS solution to be used in the R&E environment by GÉANT and the NRENs. The main idea of the SPA efforts is to setup the basis needed to support agile development of services based on new technologies and to foster the creation of an integrated information framework that provides the common service blocks and data for all services using standardised APIs.

The SPA pilot implementation closely follows the best practices and toolkits (information data model and API specifications) provided by TM Forum using a microservices approach based on free, open source components. It should be noted that incorporating free and/or open source components carries the need to determine the quality and ongoing support mechanisms for those components, which should be a concern for production services environments (but outside of the scope of this document).

The major steps involved in the work carried out in Task 2 at the beginning of the project was focused on analysis of the current status of operations and gathering requirements from the GÉANT network operations centre. Based on this input, the initial blueprint for a customised OSS/BSS solution and its basic major components and their interconnections was defined, together with a roadmap that describes the steps of development and implementation of the solution. The published roadmap milestone document [[RM](#)] describes the initial analysis and requirements and provides an overview of the SPA roadmap.

The work done in this task is also related to the Systems and Processes Architecture for e-Infrastructure Integration developed by JRA1 Task 3, which focuses on the implementation of a multi-domain service orchestrator. In this integration scenario, the orchestrator should interact with a given network domain by contacting the developed SPA-based OSS/BSS that serves the domain taking the role of a partner as opposed to a typical end-user client. This type of interaction should be developed in collaboration with JRA1 T3 and will be further explored in the next versions of the SPA.

The SPA pilot that starts in January 2018 will focus on the migration efforts needed to transition a service into SPA, as well as on the challenges that need to be tackled to ensure reliable and efficient OSS/BSS environment when running the migrated service using a set of alpha users. The pilot is planned to run for four months. At the end of the pilot, a report will be published that will include the pilot results and lessons learned about the operational changes required when using the new generation OSS/BSS approach to service-oriented network architecture.

3 Service Provider Architecture

3.1 General Design Principles

The main body of research in the future architecture of agile OSS/BSS solutions (ex. [MSB]) focuses on distributed, loosely coupled solutions that closely follow the Service Oriented Architecture (SOA) paradigm, in combination with the Event Driven Architecture (EDA). By leveraging EDA and SOA, an OSS/BSS solution can be developed as a model-driven, agile “sense and respond” system that can (in near real-time) detect external events and trigger processes to enforce end-to-end management.

The SPA design follows a lightweight approach using microservices accessible via a standardised, TMF-based, REST-based API, as shown in Figure 3.1. A business process management (BPM) engine is in charge of components orchestration and response to external events. An enterprise service bus (ESB) is used to implement connectors, i.e. wrappers, for the native APIs of the components, in order to ensure the use of standardised APIs that are independent of the specific choice of the component. Another function provided by the lightweight ESB implementation is the message queuing that enables buffering and scheduled delivery of messages between components.

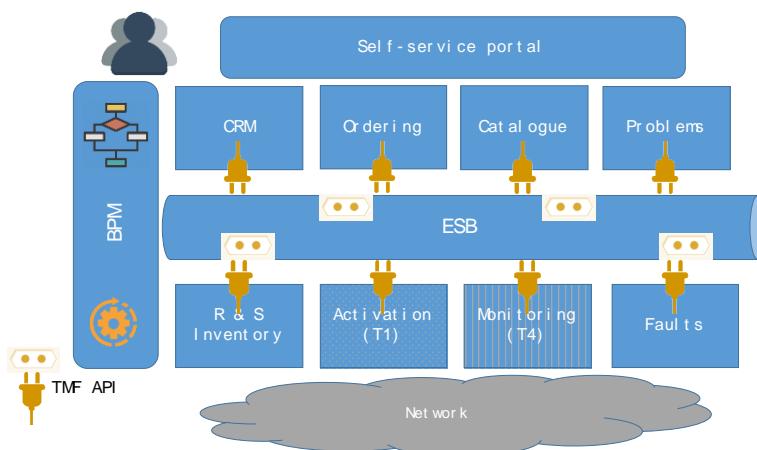


Figure 3.1: Pluggable OSS/BSS architecture

The high-level SPA design and its relationship to existing standards and efforts has been published in two research papers that were presented on international IEEE conferences:

- “Customer-Centric Service Provider Architecture for the R&E Community”. Published in the proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2016 [CCA].
- “Transforming silos to next-generation services”. Published in the proceedings of the 17th IEEE EUROCON International Conference on Smart Technologies, 2017 [TS].

3.2 Standardisation

The current best practices in the OSS/BSS design standardisation [[ODL](#)] for communication service providers (CSPs) are provided in the TM Forum Framework documentation [[TMFF](#)].

The Framework approach to the next generation OSS/BSS solution is to adopt the lean operator model turning towards reduced operational costs and flexible infrastructure. Reduced cost is achieved with high level of automation. The TMF Framework proposes a distributed environment as advocated in SOA, where interfaces are published and there is a seamless flow of information between modules, which is to be achieved with standard information model and well-defined interfaces for interaction.

The TMF Framework comprises three separate views of the OSS/BSS solution:

- Business process framework (eTOM) – provides a frame of reference for business processes, detailing generic business processes in the different domains.
- Application framework (TAM) – describes the ecosystem of OSS/BSS applications (components). All components are presented as groups of functionalities on different levels of details.
- Information framework (SID) – the last building block that provides common information data vocabulary to minimise duplication and overlap.

To ensure a standard approach and interoperability, the service bus connectors implement TMF-compliant, open API. During the last year, the TMForum has published a number of OpenAPI definitions [[TMFAPI](#)] that are based on the microservices approach and can thus be readily used as the basis for defining the necessary connectors in the SPA solutions.

In this way, each chosen component in the SPA corresponds to a component as defined in TAM, and the TMF OpenAPI developed connector is following the specification that is based on the SID data model. In addition, within the BPM engine all defined processes are based on the eTOM documentation.

With all messages across the service bus being SID aligned it is ensured that if components change the effort required to integrate the new component into the environment is reduced enabling faster introduction of new components.

Most end to end business processes need to orchestrate multiple components. By insulating these business processes from the functions relating to the various end point components using a standard Process Framework such as eTOM, they are independent of the components [[BRD](#)].

4 Pilot SPA Implementation

The initial pilot SPA implementation focuses on the customer centric processes and activities. These include service catalogue browsing, and service fulfilment (ordering, activating, using and terminating services). The next versions of the implementation will focus on adding features concerning service feasibility and assurance.

Figure 4.1 represents the components and the main information flow for version 1.0 of the SPA pilot, together with the main TMF OpenAPIs that are used and the corresponding component choices.

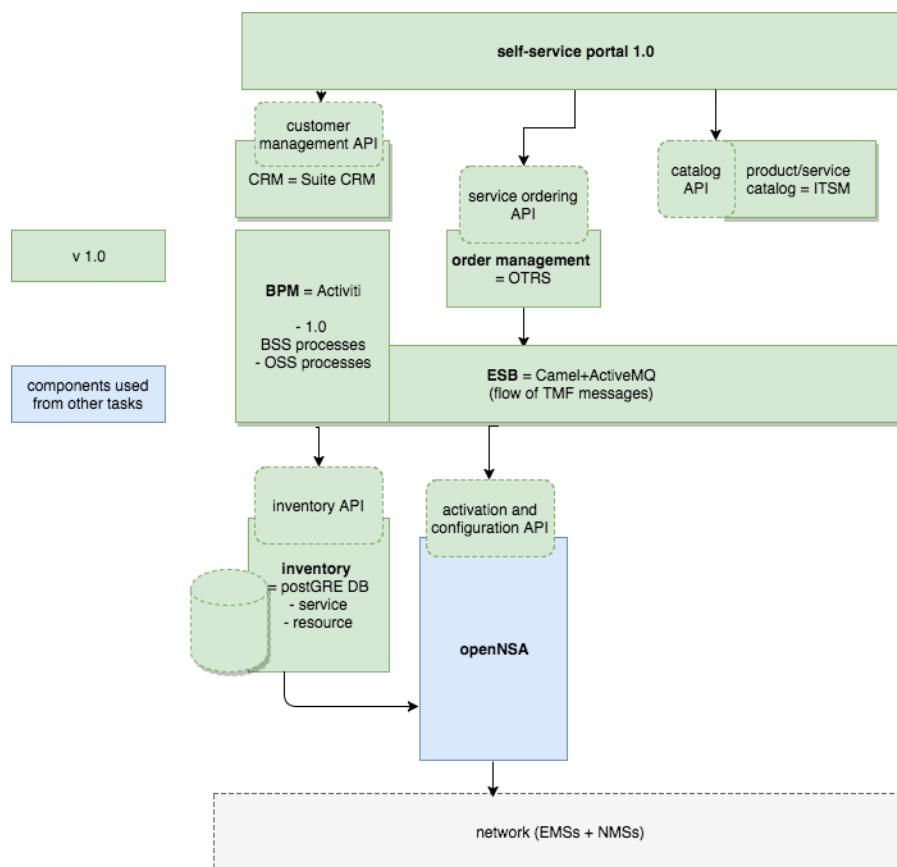


Figure 4.1: Pilot SPA components v1.0

The functionalities of the components for the initial pilot are:

- Self-service portal: A web GUI interface for the customer providing the following actions:
 - Browse available services provided in the catalogue.
 - Order a service and receive feedback upon automated provisioning.
 - Terminate active service.
 - Browse ordering history.
- CRM: Customer database that stores essential customer info.

- Order management: Order tracking system implemented using service ordering tickets connected in a parent-child fashion. The service ordered by the customer creates the initial parent service ordering ticket. During the process of provisioning the order is decomposed in the underlying RFSs needed for the implementation, where each RFS provisioning is tracked using a newly created child ticket attached to the parent customer facing services (CFS) order ticket.
- Service catalogue: Stores the information about all available services (customer facing services – CFS and resource facing services – RFS) and their status, the service specification (service specific parameters) and inter-service relationship description (interdependency between CFSs and RFSs). Based on the service specification defined in JSON, the self-service portal provides auto generated forms when the user orders a chosen CFS. Only CFS services with active status are available for ordering.
- BPM: Engine that drives the defined and implemented business processes, and acts as the main logic of the system as a whole. Each process can be triggered by an external event (from the self-service portal, or other entity that calls the engine API). Each process defines all necessary steps needed to implement a complex request. The list of processes that are going to be implemented in the initial pilot is provided in Table 4.2 .
- ESB: Lightweight implementation of an Enterprise Service Bus that is used only for message transformation in order to make sure that the messages that are passed between the SPA components are TMF compliant
- Inventory: Database that serves as service inventory and logical resource inventory.
- Activation & configuration: Service-specific module that should activate the service after which the service will be officially provided for use to the customer.

All components used in the pilot SPA are free and open source. Each choice is based on a separate investigation of available and suitable solutions for each component separately. In addition to being free and open source other component criteria included: available support, active development community, performances, possibility to extend the underlying data model, and compatibility with current GÉANT choices for components.

Whenever necessary an extension is developed for the original open source component so that additional information can be stored about the items. All extensions in place are made according to the compliance with the TMF SID model. Examples of these extensions include adding additional types of relationships between services in the catalogue, adding additional fields to tickets that will interact with GEANT's OTRS system, extending the customer data model in SuiteCRM, etc. All extensions are documents and installation scripts are designed so that new fresh deployments of the customised components can be made upon demand.

The service and resource inventory is the only component that was developed in-house. The original intent was to use the newly developed TMF compliant inventory provided by GEANT, but since the inventory project was put on hold after the initial design stage, the SPA task team developed a restricted implementation of the inventory that is in full compliance with the GÉANT inventory design that is to be used for the SPA pilot purposes until the complete GÉANT inventory component is ready to be used. Based on the pluggable framework design the switch from the pilot inventory to the production GÉANT inventory will be seamless for the rest of the architecture.

Admins can also use the backend interfaces of the components for data maintenance such as: adding new customers or services, updating information and alike. Also, the BPM GUI is available for overview of the current status of the active processes and the process history. The inventory can be directly accessed for bulk data input via the DB admin GUI interface.

4.1 Components and Mapping

The following tables provide an overview of the mapping between the TMF defined OSS/BSS components (based on TAM documentation) and the chosen component in the initial SPA pilot implementation. Information on component functionality and related data stored within the component is also included.

TAM component (TMF Framework)	SPA component	Functionality and data stored
Customer Self-Management	Dynamic self-service portal = DJANGO CMS	User GUI to interact with SPA (browse, order, track services), calls BPM
Customer Information Management	CRM = SuiteCRM	User information (organisation and individual related)
Customer Order Management	Order management = OTRS	Order general information, order items and related parameters, status of orders
Service/Product Catalogue Management	Catalogue component = ITSM OTRS	Services offered, their relationship to underlying services needed to support them including each individual service specification Provides the separation between CFS and RFS.
Service Inventory Management	In-house SID compliant Service Inventory = PostgreSQL DB	All service instances and their parameters and relationship to the resources used for provisioning
Resource Inventory Management	In-house SID compliant Resource Inventory = PostgreSQL DB	Available resources and their current status
Service Activation Management	Service dependent component (OpenNSA chosen for pilot)	Activates the RFS based on the input parameters

Table 4.1: TAM components mapping to current operations and SPA

In addition to the standard TAM components, the SPA pilot implementation includes a lightweight version of a service bus, based on Camel and ActiveMQ used for message queuing and reliability, and for developing connectors that define the Open APIs for the components. The Camel and ActiveMQ combination has been chosen such that if it is decided to move to a more heavyweight complete ESB implementation it is easy to extend the components and use Service Mix [SM], which is a well-known ESB based on these two components.

The second additional component is the business process manager, Activiti, used for the definition and implementation of all business processes for orchestration of the components. Activiti is an open source workflow engine written in Java that can execute business processes described in BPMN 2.0. Activiti is also the foundation for Alfresco's Alfresco Process Services [APS], which makes it easier to combine this component with the work done in other activities in the project, such as Service Activity 2, where Alfresco for Document Management is used as part of the overall digital business platform.

All components are used by invoking a TMF-compliant, open API that provides the functionalities of each component. All APIs are REST based, using POST to create new or modify existing information, and GET to retrieve the stored information. The following table provides an overview of the APIs developed for the SPA pilot.

TMF OpenAPI	Related SPA component	Functionality description
Activation and Configuration API	Connector developed for OpenNSA [NSI]	Allows the user to retrieve, create, update, delete services. The same API can be used to Activate and Configure Services or Resources.
Customer Management API	Connector developed for SuiteCRM	Provides a standardised mechanism for customer and customer account management, such as creation, update, retrieval, deletion and notification of events.
Resource Inventory Management API	Native API for accessing the in-house developed resource inventory DB	The intent of this API is to provide a consistent/standardised mechanism to query and manipulate the Resource inventory.
Service Catalogue API	Connector developed for the ITSM module for OTRS	The Service Catalogue Management API allows the management of the entire lifecycle of the service catalogue elements.
Service Inventory Management API	Native API for accessing the in-house developed service inventory DB	The intent of this API is to provide a consistent/standardised mechanism to query and manipulate the Service inventory.
Service Ordering Management API	Connector developed for OTRS	A standardised mechanism for placing a service order with all of the necessary order parameters. It allows users to create, update & retrieve Service Orders and manages related notifications.

Table 4.2: TMF OpenAPIs used in SPA

The business processes to be implemented in the first version of the pilot correspond to the customer domain, end-to-end business processes, as defined in TMF Business Processes Framework. The following table lists all processes that are included in the pilot and their corresponding name in the business process engine that is part of the SPA implementation. The SPA implementation is carried out using multiple processes that are connected using subprocess calls. The main idea is to separate the BSS orchestration from the OSS orchestration, and provide a logical division of business processes that are easy to change.

eTOM Business Process	SPA Business Processes	Functionality Description
Request to Answer	Service Offer	Get list of available services in catalogue and obtain the necessary CFS parameters for a chosen service
Order to Payment	Service Order, Handle RFS Order, Provisioning	Record an order for activating a service and track its status, decide on the underlying RFSs that need to be activated based on catalogue info and handle each activation separately
Request to Change	CFS Change, RFS Change	Record a request for changing the parameters of an active service, check all underlying RFSs that need to be changes, handle individual changes and track progress
Termination to Confirmation	CFS Termination, RFS Termination	Record a request for terminating an active service, track progress while terminating all individual underlying RFSs

Table 4.3: eTOM business processes mapping to SPA

The highest-level process models that are called by the self-service portal, based on user actions are provided in the following figures. All tasks within one process are either automatically implemented (note the small wheel icon in the upper left corner) or are waiting on additional input from the user (note the small user icon in the upper left corner). Wherever necessary, exception handling is also implemented within the process, enabling the engine to handle the problems that might occur whenever automated tasks are not behaving as expected.

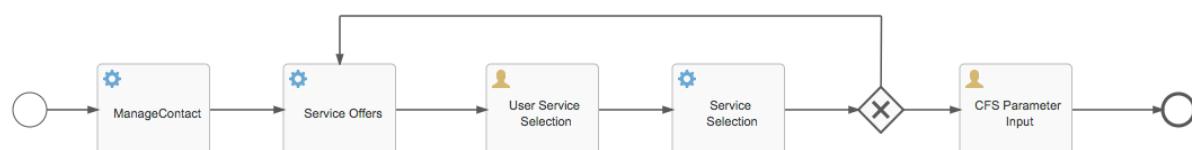


Figure 4.2: Service offer

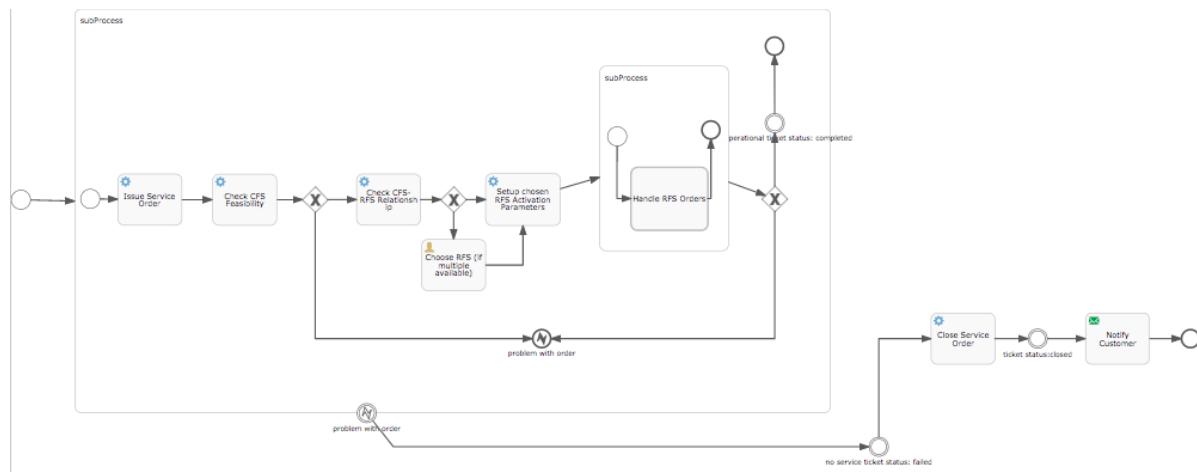


Figure 4.3 Service ordering

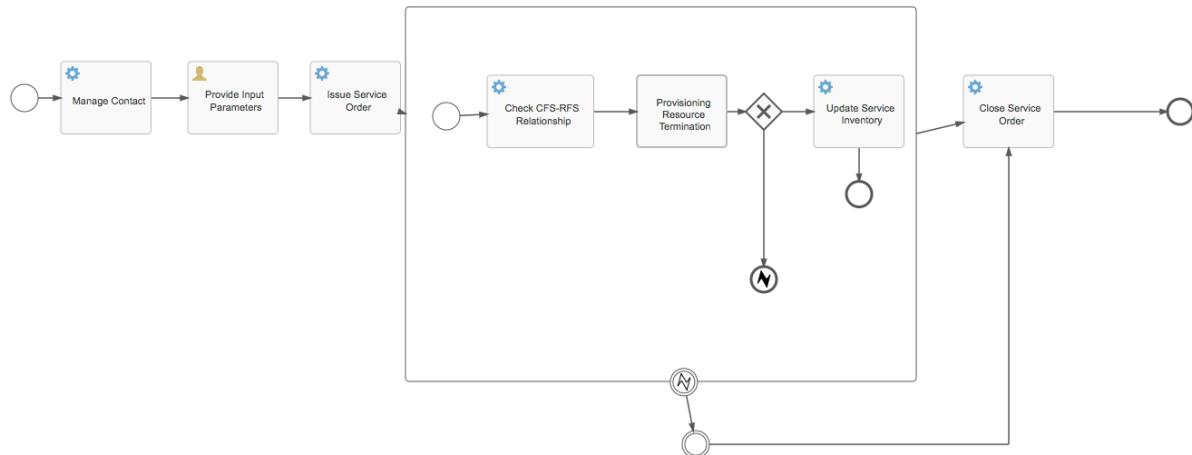


Figure 4.4: Service termination

4.2 Pilot Organisation

The initial pilot implementation for SPA is envisioned as a development pilot that is deployed and tested in order to: obtain the additional information necessary to revisit and check the high-level SPA design decisions, identify possible weak spots in the architecture, test the response in a loaded environment, and identify and address any other flaws in the implementation or deployment.

Since almost all of the current SPA development has been done in manner that is service agnostic, the pilot implementation will be comprised of two phases:

- A migration phase where a sample service is migrated from its current service context to be accessible from the SPA business process workflow.
- A test phase where the service will be offered to a restricted set of alpha users from several NRENs and the GÉANT service management team aiming to gather the necessary information

about testing the service related actions, identify shortcomings, and determine the corrective actions to fix or improve the process.

The total duration of both phases is four months, which dedicates two months per phase. The official start of the first phase is January 2018.

This initial SPA pilot does not aim to transition into a production environment. There are many essential SPA components remaining to be implemented and many questions need to be evaluated as to the scope of effort of transitioning services to SPA or their operational complexity. The results of this pilot will be published, and the JRA2T2 SPA task will fold the findings into the next phase of SPA and E-Line development.

4.2.1 Phase 1: Porting the Consolidated Connection Services (CCS) to the SPA-Based Design

The initial migratory service will be the JRA2-T1 Consolidated Connection Services (CCS). This effort of migrating CCS to SPA is referred to as the E-Line project – the name taken from the target CFS it provides and Ethernet Private Line as defined by Metro Ethernet Forum (MEF) [[EPL](#)].

With the components, APIs and processes in place, the service specifics need to be implemented in the overall architecture. The activities that will take place during this phase include:

- Analysis of the E-Line service definition.
- Defining corresponding customer-facing and resource-facing, TMF compliant service specifications.
- Representing their relationships in the service catalogue.
- Mapping the service specific information with corresponding attributes added to the service inventory.
- Populating the resource inventory with all information necessary to represent the CCS network infrastructure over which the pilot is going to run.
- Customising the activation and configuration module to the service definition.

The outcome of this phase is an SPA implementation that supports the creation, activation and termination of E-Line service instances accompanied with a clear understanding of the general and service specific parts of the OSS/BSS solution and an analysis of the effort necessary to port/add a new service to the existing architecture.

For the purposes of the SPA pilot, JRA2-T1 CCS will develop an OpenNSA virtual production network environment that corresponds to the OpenNSA deployment in the GÉANT core network. In this way, the pilot will run in an environment that reflects the GÉANT production environment.

At the end of this phase, the SPA components suite will be deployed in a pre-production environment within a GÉANT Testbeds Service (GTS) testbed. All SPA components are deployed across several VMs in order to foster the distributed nature of the solution, and the pilot will run in a similar environment so that the results can provide insight to the possible problems and issues that might arise due to the distributed hosting of the components. An automated deployment process is developed for the creation of a GTS testbed and installation and configuration of all SPA components.

4.2.2 Phase 2: Using E-Line to Analyse SPA Implementation

The E-Line service will be enabled for use via the self-service portal to a selected subset of alpha users to analyse the correctness and effectiveness of the solution. The chosen alpha users will be test representatives of different NRENs and GÉANT operations that will play the roles of end users for the E-Line service.

The main goal of this phase is to analyse whether the implementation of each component or process provides the necessary functionality for successful provisioning of the service and related self-service actions.

This phase will last at least 60 days, during which several, specific real-life scenarios will be purposefully put into effect to stress the flexibility and robustness of the solution.

The self-service portal will be analysed for perceived ease-of-use (aka user acceptance) and functionality. The feedback from the alpha users will be used to make changes in the GUI layout during the subsequent design phase.

Examples of the functionality testing scenarios that are going to be developed include:

- Service requests in a multitenant environment.
- Dealing with incorrect and stale input.
- Correct serialisation of simultaneous requests to avoid deadlock or conflicts.
- Automated use of new information added to or changed in the inventory.
- Correct, real-time feedback in the self-service portal.
- Error message handling and propagation.

All scenarios are going to be defined in separate test scenario documents created during Phase 1. This pilot integration testing documentation will include the test scenario description with end user stories, the steps executed by the end user, the desired outcome of the scenario, and the obtained outcome for all scenarios where end users have observed a problem. Any minor issues discovered will be solved and retested during this phase.

Based on the outcomes of the test scenarios, the pilot, and the captured user experience, the business processes implementation will be refined and optimised if necessary. Other major problems or performance issues will be recorded and analysed. The necessary changes in the design and/or implementation will be made in the next version.

4.3 Pilot Outcomes

An additional Deliverable will be written at the end of the pilot that will describe the successes and/or challenges encountered or revealed during the migration and trials in the pilot.

The first part of this report will specify the necessary steps and scope of effort estimations of the process of porting a service to SPA, outlining the outcomes of the first phase of the pilot.

The second part will provide an in-depth analysis of the functional and performance results of the implementation of SPA, focusing on the outcomes of the second phase of the pilot. All activities of the service users will be evaluated individually and categorised based on common actions. This will enable the Task 2 team to detect the potentially weak links in the current design of the architecture, as well as spot potential bottlenecks or inconsistencies in the data model.

The information gathered from the pilot will be used as input into the design phase of the next version of SPA, where a solution will be devised to address the identified problems. The pilot documentation will provide a qualitative and quantitative comparison of the effectiveness of the SPA solution to the current workings of the partner portal using existing GÉANT operations KPIs wherever possible, in combination with the TMF defined set of OSS/BSS metrics [[KPI](#)].

4.4 Pilot Extensions

Future SPA work will build on the pilot experience and add additional features to the solution.

The first components to be added to the pilot are service verification and performance monitoring. The Task 2 team is in close cooperation with JRA2 Task 4 Performance Monitoring and Verification team, which is developing these components in-house. Based on the guidelines provided from Task 2, corresponding TMF Open APIs will be used to contact these components and connect them into the rest of SPA. The business processes will be extended to include verification and start of routine monitoring after activation before the service is presented as 'ready for use' to the end user. The end user will then be able to retrieve monitoring information using the self-service portal.

Other extensions that are included in the roadmap for Task 2 include checking service feasibility using defined business rules before trying to activate it, as well as providing the possibility to change the service parameters of an already activated service.

While the pilot is running, the Task 2 team will continue development in parallel, therefore it is expected that by the end of the pilot, new features will be added to as the work progresses.

5 Conclusions

The SPA efforts described in this pilot review document represent the start of building GÉANT's future OSS/BSS solutions. The design and development outcomes obtained in this phase of the project will provide valuable input for the future work on SPA. They will help identify weak links in the architecture, possible performance or scalability issues, or problems with data storing and synchronisation across the layers.

Ongoing SPA work will build on the pilot outcomes revisiting the design and any identified implementation problems in the next version. Once all issues are addressed, the SPA team will continue with roadmap progress, adding features to the solution.

It must be emphasised that the work on development of OSS/BSS components is a permanent ongoing process in the organisation that aims to continually improve the efficiency of operations and provide new functionalities to the end users. The work done on SPA does not aim to turn the architecture into a production service, but provide a clear pathway that will lead to this long-term goal. By the end of GN4-2, the SPA team will have carried out the base SPA framework pilot that includes setting up the most essential OSS/BSS components exposed using standardised APIs and implementing the main customer-centric business processes that fall into the fulfilment and assurance operational process areas according to the TMF. This work will be carried out incrementally, with two more extensions to this pilot focusing on assurance and then feasibility, ending with SPA pilot v1.2.

References

[APS]	Alfresco Digital Business Platform
[BOSSF]	Jithesh Sathyan, Fundamentals of EMS, NMS, and OSS/BSS, CRC Press, 2010
[BRD]	GB921 Business Process Framework and ITIL Together R14.5.1 Standard
[CCA]	Customer-Centric Service Provider Architecture for the R&E Community
[CCV]	White Paper OSS Systems Need to Evolve From Network Management to Customer Centric Service Management, Cisco, 2012
[EDA]	Programming without a call stack – Event-driven architectures
[EPL]	E-Line: Ethernet Private Line defined by MEF
[GMOSS]	The Guide to Modern OSS, OSS Line, 2016
[ITUM]	M.3100 : Generic network information model
[KPI]	GB988 TM Forum Metrics Definitions R16.5.1 Standard
[MSB]	Transforming BSS/ OSS systems to Microservices Architecture
[NSI]	NSI Connection Services v2.0 – Open Grid Forum
[ODE]	Open Digital Ecosystem, TM Forum
[ODL]	Open Daylight https://www.opendaylight.org/
[RM]	M8.2 Service Provider Architecture Roadmap
[SM]	Apache Service Mix
[SOA]	SOA Blueprint – SOA Practitioners Guide
[TMFAPI]	TMForum Open API Table (accessible by members only)
[TMFF]	TM Forum Framework
[TS]	Transforming Silos to Next-Generation Services

Glossary

API	Application Programming Interface
BPF	TMForum Business Process Framework
BPM	Business Process Management
BSS	Business Support Systems
CFS	Customer Facing Service
CRM	Customer Relationship Management
CSP	Communications Service Provider
DB	Database
EDA	Event Driven Architecture
EMS	Element Management System
ESB	Enterprise Service Bus
eTOM	TMForum Business Process Framework old naming
GUI	Graphical User Interface
ITSM	IT Service Management
ITU	International Telecommunications Union
KPI	Key Performance Indicator
MEF	Metro Ethernet Forum
NMS	Network Management System
NSI	Network Services Interface
OSS	Operations Support Systems
OTRS	Open-source Ticket Request System
RFS	Resource Facing Service
SID	TMForum Shared Information Data
SOA	Service Oriented Architecture
SPA	Service Provider Architecture
TAM	TMForum Application Mapping
TMF	TeleManagement Forum