20-06-2018

Deliverable D5.5
Service Transitions and Validations

Deliverable D5.5

Contractual Date:	31-05-2018
Actual Date:	20-06-2018
Grant Agreement No.:	731122
Work Package/Activity:	5/SA2
Task Item:	T1
Nature of Deliverable:	R
Dissemination Level:	PU
Lead Partner:	PSNC
Document ID:	GN4-2-18-299781
Authors:	Marina Adomeit (AMRES), Michael Baierlein (LRZ), Pawel Berus (PSNC), Gerard Frankowski (PSNC), Ivana Golub (PSNC), Szymon Kupiński (PSNC), Nikola Mancic (AMRES), Aleksandra Radulovic (MREN), Marko Stanec (CARNET), Bartosz Walter (PSNC/PUT), Marcin Wolski (PSNC)

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 reports on the service validations and transitions to SA2 production performed by SA2 T1 in GN4-2 from May 2016 to March 2018. The validation and test process, including the test strategy is presented. An overview of the service, test phases, test team composition and scope of the testing is provided for each transition and validation completed during the reporting period.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

The Service Transition and Software Management Task (T1) of the Trust & Identity and Multi-Domain Services Activity (SA2) is responsible for carrying out a comprehensive assessment of candidate services to determine their readiness for production. This is carried out through a number of tests, such as pre-production testing, security testing and validation of service design and support documentation, among others. This report covers the activities of the SA2 T1 test team, including service transitions to production as well as customised quality assurance requests, in the first 24 months of the GN4-2 project, from the start of May 2016 to the end of April 2018.

During the reporting period, SA2 T1 successfully supported the transition to production of eduroam CAT GEANTLink Installer and CAT managed IdP and the pre-production evaluation of the Multi-Domain VPN Service Inventory (MD-VPN-SI), as well as the customised quality assurance assessments of three other services – perfSONAR, Firewall on Demand and GTS. In each of these cases, SA2 executed a number of tests to verify whether the service complied with the defined quality and security parameters and other criteria. These criteria are set out either in the form of quality statement for transitions to SA2 production, or as acceptance criteria for customised requests.

In addition to carrying out its regular transition to production assessments, the team worked on improvements to the service validation and testing process inherited from the GN4-1 project. This involved the formalisation of standards and best practices, and the creation of a knowledge database, including a set of relevant templates for documenting tests results, guidelines for requesting the service, etc.

Work started on defining baselines and policies for service validation and testing aspects, such as the adoption of a consistent test strategy based on existing industrial models and standardised approaches. In collaboration with the Service Optimization Task (SA2 T4), the team is also actively looking at potential continual service improvements to the testing process output, i.e. test reports with incidents, problems and error records.

Summary reports of the performed transitions and quality assurance requests are presented here and include the scope of the testing process, a technical description of the test environment, the tools used for testing and the test teams engaged.

All tested services have reported tangible benefits in terms of improved insights into the quality of products and stability of operations. This was all achieved through a strong collaboration between team members within the Activity, as well as with other relevant GÉANT project research, service and networking activities.

1 Introduction

The Trust & Identity and Multi-Domain Services Activity (SA2) in GN4-2 is dedicated to the efficient and effective operation of Trust & Identity and Multi-domain services in production, following the relevant operational procedures, in line with GÉANT's commitment to provide high standards of quality and levels of availability of service to its partner NRENs and the R&E community at large.

SA2 service operational teams intervene as needed, once the GÉANT services that are developed by teams in other GN4-2 project activities reach production, to streamline the design of technical components in terms of the operational models used and help transition the services to the production environment, making sure that they are fit for purpose and fit for use. To this end, SA2 defines and carries out the service transition process for the services, coordinating all sub-processes and the various teams involved.

SA2 ensures that services in production are operated and supported by skilled experts, that the relevant procedures, processes and documentation are in place for their efficient operation, and that their operational health and usage are monitored and reported to stakeholders. The service validation and test process in SA2 is a key element of service transition. SA2's priority is to deliver high-quality and reliable services and every service entering the production phase must be validated against a minimum needed set of criteria. These criteria determine whether a service can be efficiently operated and is compliant both with internal policies and rules as well as with legislative requirements. Any potential issues are identified during this process, and possible solutions are applied before the service is launched in a production environment.

Outside of its transition to production process, SA2 also provides validation and testing for services completing a major development cycle, or that are being prepared to transition to production in other service activities. In such cases, SA2 offers a quality assurance check based on a customised subset of the service and validation testing process.

This document reports on the service transitions and validations performed in SA2, including a description of the service transition process employed for its production environment, as well as the quality assurance procedures and assessments carried out in the first 24 months of the project, from May 2016 to April 2018.

Section 2, sets out the SA2 service transition process, including the main actors involved and the interactions between them. Section 3 introduces the validation and test process and gives a detailed view of the test strategy and compliance to standards and best practices. Section 4 reports on the validation and test cases, providing on overall view of the tests carried out, their scope, the test environment and the SA2 T1 teams taking part in testing. Finally, some general considerations about testing and further enhancements to the transition process are presented in the conclusions.

2 Service Transition Process

SA2 has defined a detailed service transition to production process in line with the GÉANT services lifecycle defined by the Project Lifecycle Management [PLM]. The transition process relies on the PLM to confirm that a service has passed the development gate and can start its transition to production operations. A high-level BPMN diagram of the transition process employed in SA2 is shown in Figure 2.1. The different swim-lanes shown, correspond to teams that are responsible for carrying out specific transition process tasks, following the organisational structure of the GN4-2 project.

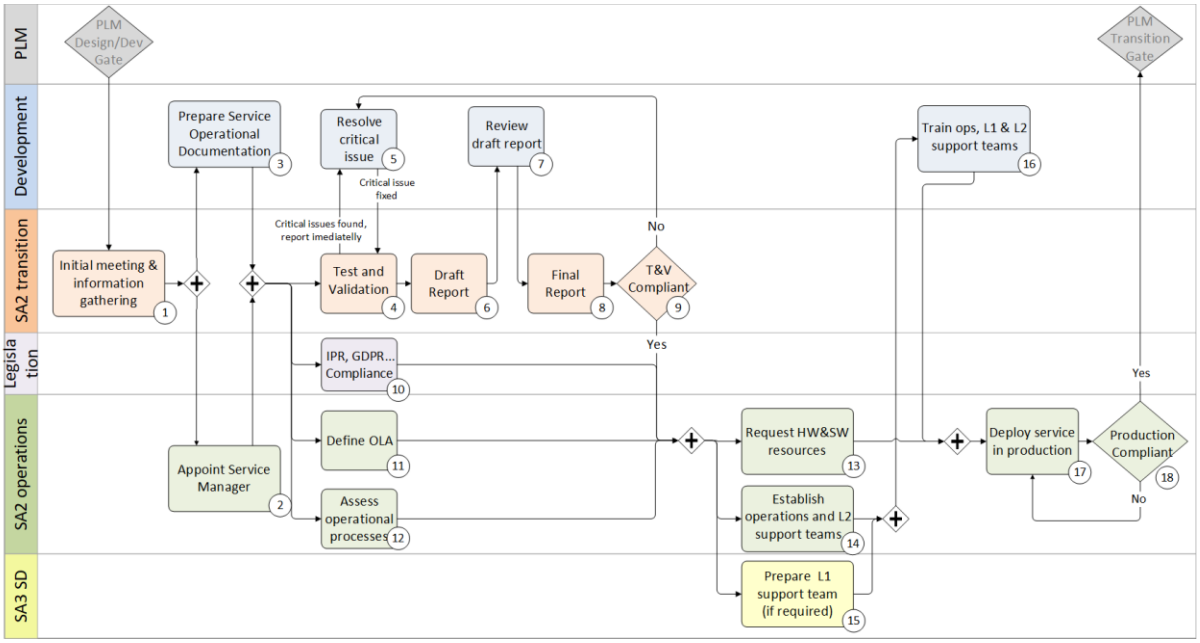

Figure 2.1: Service transition to SA2 production

The service transition process is described in more detail below, where the numbers in parenthesis refer to the relevant steps as shown in Figure 2.1.

At the beginning of the process, the SA2 transition team holds an initial meeting with the development team (1), during which basic information such as service definition and scope is exchanged, the timeline for transition is defined and any dependencies or risks are identified. Based on this information, a service manager with the necessary skillset and experience is appointed and delegated to the operational task in SA2 (2). He/she is also responsible for managing relevant operational and support teams for the service once in production.

The service development team is required to submit the complete service documentation using the template (3) [SA2_SDT] that is designed specifically for the purpose by SA2 and aligned with the ITIL service design package [ITIL_SDP]. This template contains a holistic set of service documentation including service definition, description and architecture, operational manuals, instructions and requirements, a data inventory for GDPR evaluation, etc. The service documentation's compliance with the SA2 template, i.e. its correctness and completeness, is assessed as part of the validation and testing process and again during the service's deployment in the production environment.

The SA2 Task 1 team performs validation and testing of services against various quality and security criteria, which are explained in detail in the next section. Based on the service definition and scope, a test plan is defined, the relevant test teams are appointed, and test artefacts prepared to initiate testing and validation (4). Any critical issues found are immediately reported to the development team (5).

Once testing and validation are completed, a draft report containing the test results and recommendations is prepared (6). The draft report is then shared with the development team so that they can review it and identify any false positives (7). Based on this review, a final report is prepared (8). If no critical issues are found, the service is considered compliant with SA2 quality requirements and to have passed testing and validation (9). Otherwise, any critical issues identified will need to be resolved by the development team before the service can be deployed in production.

The validation of service compliance with legal requirements such as IPR and GDPR is performed by the relevant Networking Activities (NAs) in the project (10).

In parallel with service validation and testing, based on the service documentation provided, the service manager prepares the Operational Level Agreements (OLAs) with the relevant teams (11) and assesses operational processes and their compliance with operational capabilities and practices within SA2 (12).

After this, the assets needed for production operations are prepared, and the operational & L2 support (14) and L1 service desk teams are established (15) and trained by the development team (16). The needed hardware and software resources are then acquired (13). Once all assets are ready, service deployment in the production environment can start (17). Depending on whether a service's pilot infrastructure and user data need to be preserved, a detailed deployment plan is also defined. Alongside this, the user documentation, websites and necessary trainings are prepared.

Finally, the service is deployed in production in compliance with internal practices and policies (such as for monitoring, backup&restore, archiving, etc.) and once deployment is confirmed (18) service reaches the PLM Transition gate. This PLM gate formally confirms that business and operational criteria are met, and the service can thus enter its operational stage.

The above transition process is employed when a new service reaches production. In case of new features or enhancements to an existing service already in production, the service/product manager and the development lead employ the transition process selectively, depending on the impact of the change.

3 Service Validation and Testing

The service validation and testing process (SVT) was introduced for the first time as a dedicated task within the SA4 activity (SA4 T1) in the GN4-1 project [GN4-1-D8.1]. Using the SVT process defined by ITIL [ITIL-FH], the aim of this task was to ensure that a new or changed service or service offering was *fit for purpose* and *fit for use*. In this context, the term 'fit for purpose' refers to how a service supports the target business performance or removes constraints. Thus, the quality of a service depends on its conformance to business and user demands or expectations, which can be measured by overall user satisfaction or represented by compliance with internal policy, regulatory, legal or standards documentation.

The testing schema defined in GN4-1 has been extended in GN4-2 to cover conformance with International Software Testing Qualifications Board standards [ISTQB]. The ISTQB defines comprehensive and industry-proven practices for testing, and the process accompanying software development lifecycles.

The approach used in SVT is outlined in the following sections including the test strategy, a description of roles and responsibilities, and a description of the process's execution.

3.1 Test Strategy

As defined by the ISTQB, a test strategy is a set of fundamental documentation defining the approach to testing activities within a project, describing a common understanding of how artefacts should be tested to achieve defined objectives. A test strategy identifies the types of tests to be performed, and is based on a combination of requirements, identified risks, industry standards and consultations with users and developers.

The test strategy adopted by SA2 T1 draws upon a set of best practices, guidelines and internal policies, from existing industrial models and standards (e.g., ITIL, or ISTQB certificates), tailored to the needs of GÉANT. Therefore, the strategy is consistent with industrial state-of-the-art recommendations, while also addressing elements specific to GÉANT SA2 transition process and GÉANT services.

The test strategy defines a set of test types to cover testing of the complete system, including the functional and non-functional requirements of a product and the ability to operate the service in production. The test types supported by the SA2 team are described in the following sections, while the supporting tools used during execution of specific tests are presented in Appendix B.

Each test strategy is executed in three test phases i.e.: preparation and planning, test execution and reporting; these are described later in this section. This generic strategy is customisable and adjustable to the needs and specific requirements of individual projects.

3.2 Test Types

GÉANT services are usually developed and operated within joint research activities that may also deal with various aspects of user interaction. One service may only require simple command line-based interaction with a service administrator, while another may require exhaustive interaction with end users via a graphical user interface. Given this generic nature of the GÉANT services, SA2 T1 is running its validation and testing activities based on specific bundles of the test types described below. This list of test types is by no means exhaustive, but rather is evolving and can be adjusted and extended for specific use cases, based on the needs of each service. Compliance rules and best practices that are related to the service being tested are associated to each test type.

Depending on their nature, some services may require validation and testing against the full set of test types, while others may only require validation and testing against a subset of tests. Regardless of which subset of test types is chosen, the association of standards and best practices to each test always guarantees the compliance and conformance of the service under consideration. In the spirit of agile software development methodologies, the range of available tests and test types are constantly being extended and improved.

Figure 3.1 shows the generic test types supported by SA2 T1 as part of its validation and testing process.

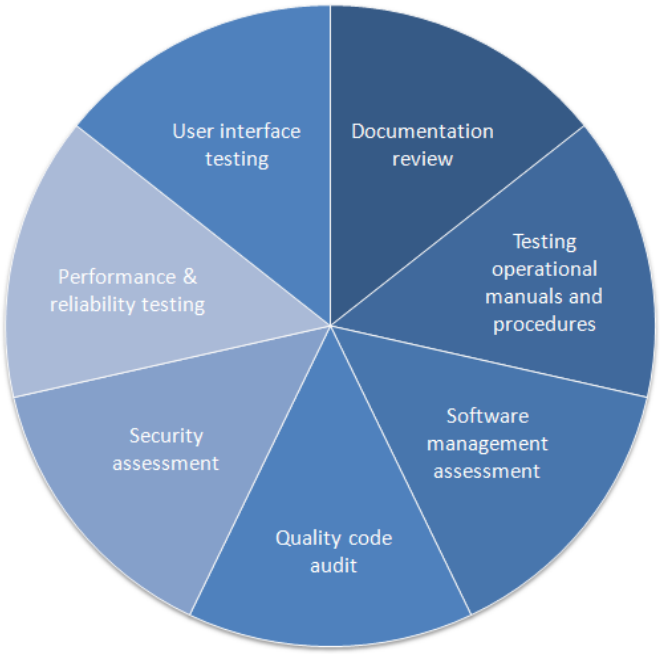

Figure 3.1: General validation and test types supported by SA2 T1

The individual tests and the steps involved are described in the sections below.

3.2.1 Documentation Review

- DESCRIPTION: A verification and review of the service documentation provided by the development team. In particular, documentation is checked for completeness, compliance with the defined template and conformance with standards and best practices.
- METHODOLOGY: The service documentation is provided by the development team and is then reviewed by qualified auditors to check its compliance against the SA2 service documentation template, and that it is complete, correct and comprehensive.
- DEFAULT SCOPE: Complete service documentation assessment. Some specific examples include:
  - Architectural overview;
  - End-user documentation review;
  - Developer guide review;
  - Testing procedures review;
  - Support team documentation review.
- STANDARDS/BEST PRACTICES:
  - **SA2 service documentation template** [SA2_SDT] – defined by SA2 and aligned with the ITIL service design package [ITIL_SDP];
  - **ISO/IEC 26514:2008** – this standard defines requirements for development of user documentation [ISO/IEC 26514];
  - **GN3 Software Documentation Best Practice Guide** [SWDocBP] – describes specific needs and constraints for software documentation.

3.2.2 Operational Manuals and Procedures Testing

- DESCRIPTION: Verifies whether the service's operational procedures (as outlined in the service operational documentation) enable the service to be operated by a future service operations team. The main focus of this test is on operation of the service under standard and emergency conditions, and the user support offered by the service provider.
- METHODOLOGY: A testing team without prior knowledge of the service attempts to follow guidelines and procedures and verifies them against the system's behaviour. User support is verified with respect to the available communication channels, and the guaranteed levels of service.
- DEFAULT SCOPE:
  - Production Deployment Guide;
  - Production Upgrade Guide;
  - Disaster Recovery Plans;
  - Backup Procedures;
  - Service Order Procedures;
  - Problem Resolution Procedures;

- Configuration Change Procedures;
- Evaluation of user support potential and capabilities;
- Evaluation of end-user support provided.

- STANDARDS/BEST PRACTICES:
  - **SA2 service documentation template** [SA2_SDT] – describes the operational processes that should be included.

3.2.3 Software Management Assessment

- DESCRIPTION: Verifies whether the main software-related processes for a service (e.g., change management or configuration management) comply with the guidelines provided in the GN3 Software Developer Best Practice Guide [SWDevBP].
- METHODOLOGY: The relevant information about software development tools and processes is collected during interviews with the development team. Based on this information and the related software documentation (e.g., Developers Guide), the usage of supporting tools and services is checked in accordance with the GN3 Software Developer Best Practice Guide. During the expert review, configuration of the software build process and version control management is performed. In accordance with the IPR policy, the presence of files with product licenses and used dependencies is checked, and the IPR Coordinator and the team receive an automatically generated list of licenses for all dependencies used in the service. Finally, smoke tests for the build process, which quickly reveal any simple failures of the system, are executed.
- DEFAULT SCOPE:
  - Software development tools usage;
  - Software build process workflow;
  - Build configuration – expert review;
  - Version control management;
  - Build process – testing.
- STANDARDS/BEST PRACTICES:
  - **GN3 Software Developer Best Practice Guide** – provides recommendations on software build, integration and release processes and source code management practices.
  - **GÉANT's global IPR policy** [IPRPolicy].

3.2.4 Quality Code Audit

- DESCRIPTION: Verifies the quality of the software used for service delivery, which has a decisive impact on the maintainability of both the software and subsequently on the depending services.
- METHODOLOGY: The code of the software product is reviewed by field experts against dedicated checklists and using dedicated tools that identify suboptimal design solutions. Violations are prioritised and then reported.
- DEFAULT SCOPE:
  - Automated code analysis supported with issue review and categorisation;

- Manual code inspection;
- Software maturity assessment;
- Database Design Review.

- STANDARDS/BEST PRACTICES:
  - **ISO/IEC 250xx** Systems and software Quality Requirements and Evaluation (SQuaRE) suite – defines the framework for the evaluation of software product quality; in particular, ISO/IEC 25023:2016 includes the code quality issues [ISO/IEC 25023].

3.2.5 Security Assessment

- DESCRIPTION: This category includes both identification of vulnerabilities by security software code audits and discovery of existing defects by penetration testing. As such, it involves static techniques (reviews, inspections) as well as dynamic testing activities. As an important aspect of ensuring compliance with the upcoming GDPR, the products will be additionally assessed for proper solutions to protect Personal Identifiable Information (whenever applicable).
- METHODOLOGY: The software source code and configuration items undergo reviews aimed at discovering any vulnerabilities that could be exploited in production. In parallel, the deployed system is subject to penetration testing by an independent team (which allows to obtain more flexibility in building the test schedule). Both approaches start by running relevant automated tools, and the results obtained are then manually verified and, optionally (if within the defined scope), manual testing (e.g., reading of the source code by a human) is performed.
- DEFAULT SCOPE:
  - Automated software code analysis supported by issue review and categorisation;
  - Manual software code inspection;
  - Penetration testing;
  - Frontend security review;
  - Backend security review;
  - Security policy compliance review;
  - Best practices review;
  - Verifying adequate protection of Personal Identifiable Information (if applicable).
- STANDARDS/BEST PRACTICES:
  - **OWASP Testing Guide v. 4.0** is a testing methodology that covers three primary areas: an OWASP testing framework for Web application development, a web application testing methodology, and reporting. The guide strongly focuses on Web application security throughout the entire software development lifecycle, and not just at security testing of an implementation. It is targeted specifically at a single domain area (Web applications), which so far covers all applications that undergo the validation process.
  - **OWASP Code Review Guide v. 2.0** is a technical manual for software developers and management containing the most important rules, techniques and practices of secure code inspection. Following these guidelines can help catch more code bugs more quickly than during the testing or production phase, which is essential in developing systems responsible for critical infrastructure.

- **GDPR** – The EU General Data Protection Regulation [GDPR] replaces the Data Protection Directive 95/46/EC and is referred to as general guidance.

3.2.6 Performance & Reliability Testing

- DESCRIPTION: Verifies if the actual performance and reliability of the service meets the non-functional requirements defined for it.
- METHODOLOGY: Performance testing starts with the identification of specific types of tests (e.g., load testing) to be carried out and the corresponding acceptance criteria, based on the service specification and non-functional requirements. Next, test scenarios and test conditions are designed, implemented and executed, together with the test environment and the corresponding planned system load (number of users, data, parallel transactions).
- DEFAULT SCOPE:
  - Volume testing;
  - Load testing;
  - Stress testing;
  - Resilience testing;
  - Scalability testing.

3.2.7 User Interface Testing

- DESCRIPTION: Validates the user interface (UI), with respect to existing UI standards and user expectations. It involves a combination of review and dynamic testing techniques, focusing on ergonomics, usability and accessibility.
- METHODOLOGY: Key user scenarios are defined based on interviews with the development team and on the user documentation review. Based on these scenarios, the test cases are designed and the user interfaces are reviewed by experts. This work can be supported by dedicated tools, against checklists for accessibility (WCAG), conformance to HTML and CSS standards, and the respective acceptance criteria.
- DEFAULT SCOPE:
  - Accessibility expert review;
  - Usability expert review;
  - System Usability Scale (SUS) survey;
  - Browser compatibility testing;
  - User testing;
  - Automatic HTML/CSS validation.
- STANDARDS/BEST PRACTICES:
  - **ISO/IEC 40500:2012** – Web Content Accessibility Guidelines [WCAG] 2.0 recommendations support web developers in the creation of accessible services for users with disabilities that are easy to navigate, understand and interact with.

3.3 Teams and Actors

SA2 T1 has established various roles and responsibilities to support the service validation and testing process. Some of these roles correspond to those defined in the ISTQB process, while others (e.g., Service Manager or Ops Team Representative) are not covered by the ISTQB and have been added to comply with the GÉANT project structure.

Role name	Description	Role in the SA2 T1 testing process	Role in ISTQB process
Software Development Team Representative	A person capable of making technical decisions and who is aware of technological constraints	Estimates the effort, reports and addresses issues, verifies fixes	Developer
Test Manager	A person responsible for managing the test process: planning, managing resources, controlling tests and reporting results	Plans, controls and monitors the testing process	Test Manager
Test Engineer	• Security tester • UI tester • Software engineer	Defines, implements and reports test cases and related testware	Test Analyst, Technical Test Analyst
Service Manager	A person in charge of the service as its business owner	Clarifies functional and operational requirements	N/A in ISTQB; derived from ITIL
Ops Team Representative	Technical staff operating the service	Reports issues and defects	N/A in ISTQB; derived from ITIL

Table 3.1: Test team roles and responsibilities

3.4 Test Phases

The validation and testing process is shown schematically in Figure 3.2. Several activities may be conducted in parallel, e.g. some tests may be executed before other tests are designed. The validation and testing process is described in detail in deliverable *D8.1 Service Validation and Testing Process* [GN4-1-D8.1].

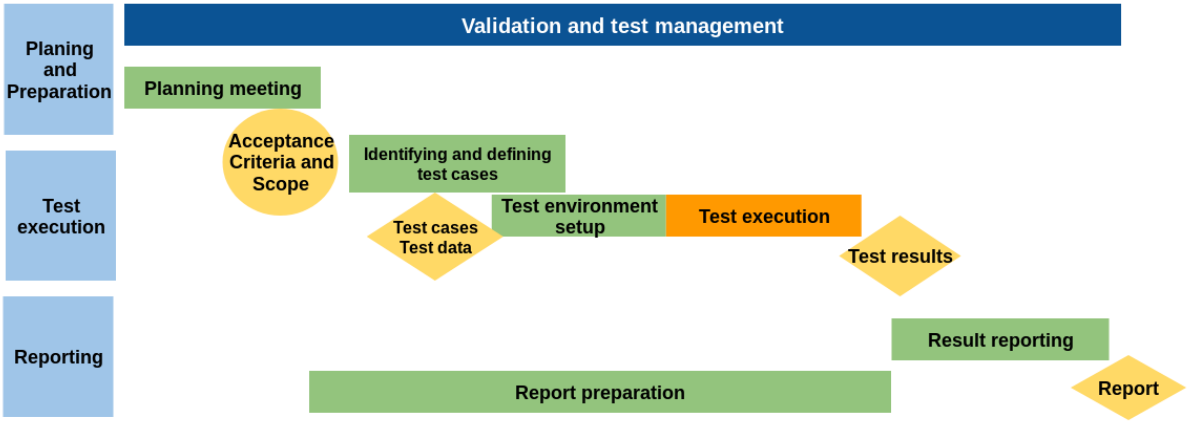

Figure 3.2: Service validation and testing process – test phases

Phase	Description	Phase of ISTQB process
Preparation and planning	The preparation and planning phase involves gathering stakeholder requirements and necessary release package elements. The development team provides a service package including service documentation, working instances, source code, functional test cases and other resources and parameters to define and complete the various tests. Necessary resources (staff, environments) are allocated and test activities are scheduled. Acceptance criteria are defined and test exit criteria determined.	Planning and Control
Test execution / Service assessment	Test execution starts with identifying and defining test cases for each required test type. Test conditions are identified. Test cases and test data is designed. The test environment setup is checked for correctness and completeness. All required tests are executed and the results are logged. All tests are planned to harmonise various aspects of a service and to be repeatable.	Analysis and Design / Implementation and Execution
Reporting	Test results are documented, evaluated against the given criteria, gathered in the report and confirmed with the development team to prevent misunderstanding of service design and implementation. The report is provided to the target stakeholders and used as an input to other relevant processes including change management and continual service improvement.	Reporting / Test Closure Activities

Table 3.2: Service testing and validation phases

The SVT process output (e.g. test report with incidents, problems and error records) can be incorporated in the continual service improvement (CSI) register to address potential improvements and bug fixes [ITIL-FH]. This guarantees that all critical issues are addressed and that services are of acceptably high quality. The completed testware – test plan, test scripts, test cases, test data, log files etc. – is stored to be re-used in future, while lessons learned are formulated to improve the SVT process.

4 Service Validation and Testing Reports

This section summarises the validation and testing projects that have been conducted by the SA2 T1 team since the start of GN4-2 in May 2016 up to March 2018. SA2 T1 performed comprehensive quality evaluations in response to:

1. Requests for service transitions to production (or major releases of a service) – with the ultimate goal of confirming that the service candidates were ready for production.
2. Requests for Quality Assurance for a service – with the aim of performing custom quality evaluations based on given acceptance criteria (e.g. vulnerability assessment).

Six validation and test projects were completed within the reporting period. A summary list of these projects is given in Table 4.1 below, while more detailed reports on each project are provided in the next sections. For reasons of security and confidentiality, the service validation and test results are disclosed only to the development teams and are not included in the test reports here. Any authorised parties interested in these results can contact the SA2 T1 Task Leader, Marcin Wolski[1], PSNC.

Service	Process type	Test types	Year
Firewall on Demand	Quality Assurance	Security assessment	2016
perfSONAR	Quality Assurance	Custom security best practices review	2016
CAT GEANTLink Installer	Service enhancement transition to production	Security assessment	2016
GÉANT Testbed as a service	Quality Assurance	Security assessment, Quality code audit, Documentation review, UI testing, Performance & reliability testing	2016/2017
Multidomain VPN Service Inventory (MDVPN SI)	Service transition to production	Security assessment	2017
CAT managed IDP	Service transition to production – preempted testing	Security assessment	2018

Table 4.1: SA2 T1 validation and testing projects

[1] marcin.wolski@man.poznan.pl

Both elements of security assessments i.e. penetration testing and security code review, were firstly performed using automated tools. The results of the automatic tests were then reviewed manually, taking into account the overall context and application environment specifics. The manual reviews excluded a number of issues that were considered either to present false positives, or to be part of a larger issue or be a feature of the given environment. Additionally, source code manual reviews (source reading) were performed. A similar approach involving automatic tests and expert reviews was applied for quality code audits.

4.1 Firewall on Demand

4.1.1 Overview

Firewall on Demand is a BGP-FlowSpec-based, multi-tenant DDoS mitigation solution allowing users (connected NRENs or recursively connected institutions with own AS and especially the NoC admins of these organisations) to control DDoS mitigations for filtering normally routed IP traffic destined for their networks by using a web UI (manual) or a REST API (automated). The validation and testing of Firewall on Demand was requested by the development team, and this case was treated as a quality assurance case.

4.1.2 Test Strategy

4.1.2.1 *Test Team*

Number of test engineers	5 experts representing 2 NRENs: PSNC and CARNET
Number of test teams	1 team formed of 3 people (PSNC) 1 team formed of 2 people (CARNET)

Table 4.2: Firewall on Demand security assessment test team composition

4.1.2.2 *Security Assessment*

Detailed scope of the security code review	• The source code analysed referred to version 1.2 of FoD. • Vulnerability detection was performed on a test instance of FoD. Vulnerability detection included testing of the following elements: ○ User Web interface components ○ Server components
Programming languages used	Python
Lines of code (without comments, empty lines etc.)	4361

Table 4.3: Firewall on Demand security assessment

4.1.3 Results

From the security point of view, the quality of source code has been assessed as high, with several minor improvements suggested in the final report. Recommendations to update versions of supporting software and libraries used to latest stable versions were also included.

4.2 perfSONAR Security Best Practices Review

4.2.1 Overview

perfSONAR is a network measurement toolkit designed to provide federated coverage of paths and help establish end-to-end usage expectations. perfSONAR provides a uniform interface that allows for the scheduling of measurements, storage of data in uniform formats, and scalable methods to retrieve data and generate visualisations. This extensible system can be modified to support new metrics and provides endless possibilities for data presentation.

The perfSONAR review was requested by its development team and treated as a quality assurance case. This process was specific in that its main purpose was to identify security best practices that will allow the perfSONAR team to increase the security of the application's development and maintenance processes.

4.2.2 Scope

The scope of the assessment was to identify security best practices that will allow the perfSONAR team to improve the security of the perfSONAR software's development and maintenance processes. The analysis was carried out in two main working areas:

1. Analysis of perfSONAR documentation and review of the website [perfsonar.net], with special attention given to its Security Considerations page and Vulnerabilities Archive. The documentation analysis was also supported with interviews with the perfSONAR project team members.
2. Installation and analysis of the dedicated perfSONAR installation in the test environment. Settings of selected environment components were reviewed. Additionally, the auditors worked with the running application installed in the test environment in order to gain the best possible understanding of it.

4.2.3 Test Strategy

4.2.3.1 *Test Team*

Number of test engineers	5 experts representing 1 NREN: PSNC
Number of test teams	1 team formed of 5 people

Table 4.4: perfSONAR security best practices review test team composition

4.2.4 Results

The evaluation clearly showed that security best practices are properly applied in the development of the perfSONAR software. The practices that were analysed covered both technical and organisational facets of the software development process. It was concluded that the deployed software instance was generally reasonably secured. No particular security vulnerabilities were identified, but some general, further improvements were suggested to the perfSONAR development team to help achieve full security hardening of a perfSONAR installed instance. An example roadmap to achieve the proposed improvements was provided to the development team, highlighting their potential positive impact on the security of the perfSONAR installation package and taking into account the effort required to implement them.

4.3 eduroam CAT GEANTLink Installer

4.3.1 Overview

GEANTLink is an eduroam installer for certain versions of the Windows operating system that is made available to end users via the eduroam Configuration Assistant Tool (CAT) site.

The main functionalities of the eduroam GEANTLink installer are:

- Imports an XML configuration file with eduroam network parameters.
- Asks for and store a username/password combination.
- Uses the configuration parameters and username/password to authenticate with the EAP TTLS PAP protocol to eduroam authentication servers.

There is also a UI with which users can review the imported settings, and a basic diagnosis app for cases where the authentication is unsuccessful.

The detailed scope of the evaluation was defined with the GEANTLink development team who delivered all necessary resources (documentation, source codes, and others). Since this was an update to the eduroam CAT, validation and testing was limited to the secure code review.

4.3.2 Test Strategy

4.3.2.1 *Test Team*

Number of test engineers	4 experts representing 1 NREN: PSNC
Number of test teams	1 team formed of 4 people

Table 4.5: GEANTLink Installer security assessment test team composition

4.3.2.2 *Security Assessment*

Detailed scope of the testing	The source code review was performed on the source code that refers to version 1.0-beta. Part of the analysis was performed on version 1.0.7. The source code was obtained on 16 September 2016, and no modifications entered after that date by the SDT could be included in the assessment.
Programming languages used	C++
Lines of code (pure - without comments, empty lines etc.)	24373

Table 4.6: GEANTLink installer security assessment

4.3.3 Results

The analysis showed that the code is solid and written using clear and consistent programming conventions, which makes it easier to analyse and further develop. The authors of the source code are aware of and consistently apply certain secure programming practices such as for example using dedicated functions for clearing memory after handling sensitive data in specific areas. A detailed analysis of sensitive operations responsible for building and maintaining a secure connection between client and server also showed a solid approach to application security.

4.4 GÉANT Testbeds Service

4.4.1 Overview

The GÉANT Testbeds Service (GTS) provides the ability to set up isolated customised virtual networks to test new concepts in networking and telecommunications.

The GTS review was requested by the development team and treated as a quality assurance case. Two main objectives were defined for the GTS quality evaluation:

- To check the system's resistance to malicious attacks.
- To verify that the management of underlying resources (virtual networks) could be performed in a secure and reliable way.

The detailed scope of the evaluation was defined with the GTS development team who delivered all necessary resources (documentation, source codes, testbed setup and others) for the execution of the required tests.

4.4.2 Test Strategy

4.4.2.1 *Test Team*

Number of test engineers	14 experts representing 5 different NRENs: LRZ, CARNET, MREN, PSNC, AMRES
Number of test teams	7 teams, 2-4 people in every team (one person could be a part of more than one team)

Table 4.7: GTS quality assurance test team composition

4.4.2.2 *Security Assessment*

Detailed scope of the testing	• The source code analysed refers to version 4.0.0 of the GTS. It was obtained on 16 December 2016 and no modifications entered after that date by the Software development team (SDT) could be included in the assessment. The application uses certain external software libraries, such as activemq, groovy, jstl, hibernate, spring, etc. • The vulnerability detection was performed in the GÉANT lab environment on a GTS test instance provided by the development team. Vulnerability detection included testing of following elements: ○ User Web interface components ○ Server components ○ Terms of Service document
Programming languages used	Java+JavaScript
Lines of code (pure - without comments, empty lines etc.)	31651

Table 4.8: GTS security assessment

4.4.2.3 *Quality Code Audit*

Detailed scope of the testing	The source code analysed refers to version 4.0.0 of the GTS. In addition to the default scope quality code review, a review of the database design was performed. Assessments were executed on a local machine with the latest DDL (sql dump) so that the test conditions were as close as possible to the real situation.

Table 4.9: GTS quality code audit

4.4.2.4 *Software Management Assessment*

Detailed scope of the testing	The software management assessment conducted by the SA2 T1 team was performed to determine the maturity of the software management processes in use by the GTS team against standards (such as git-flow [VD-ASGitBM]) and GÉANT best practices guides [QABP] [SWDevBP]. It also looked at the usage of supporting tools (SCM, CI, Issue Tracking, wiki, Static Code Analysis, binaries repository, IDE) and whether the tools provided by GÉANT were used by the SDT. Finally, the assessment took into account the ability to operate the tool in a production environment.
Test environment	• Virtual developer workstation – newly installed PC (or VM) used to verify whether it is possible to set up a development environment using the provided documentation. • Virtual build server – newly installed PC (or VM) used to verify building the distribution package. • GÉANT Software Development Infrastructure – used to perform various assessments described later in the section.
Tools	• License Gradle Plugin [License Gradle] – used to perform license check of depending libraries. • Eclipse 4.6 – used for performing Gradle build of the project and verifying developer documentation.

Table 4.10: GTS software management assessment

4.4.2.5 *User Interface Testing*

Detailed scope of the testing	User interface testing included usability and accessibility testing, together with selected elements of functional testing executed upon the user interface to validate browsers compatibility.
Test environment	The set of most popular browsers (e.g. Mozilla Firefox, Google Chrome, Internet Explorer, etc.) with their latest versions were selected to conduct usability tests including web browsers compatibility validation. Accessibility tests were performed on Mozilla Firefox and Chrome browsers that installed in the Linux OS.

Table 4.11: GTS user interface testing

4.4.2.6 *Performance and Reliability Testing*

Detailed scope of the testing	The tests included reliability testing, testbed performance and capability review as well as network performance testing to assess quantity and quality limitations.
Test procedure	The process of testing was initiated by submitting the DSL code for a Testbed comprising of two virtual machines and a virtual circuit between them. The resources were successfully submitted and

	reserved and then activated. The booking process and activation of resources were part of the reliability testing. Following successful activation of resources, the interfaces on which the virtual circuit ends (eth1 interfaces) were assigned IP parameters, and reliability testing was then continued by checking connectivity between virtual machines, as well as the connectivity of each of the virtual machines to the Internet Access Gateway (IAGW). The performance and capability review was conducted on the available VMs resources, such as the number of CPU and RAM memory etc. The peak flow by TCP and UDP protocols and other network parameters such as packet loss and jitter between two virtual machines was measured using the iPerf tool.
Test environment	At the time of testing, GTS allowed selection of resources from the following geographical locations: Amsterdam, Bratislava, Hamburg, Ljubljana, London, Madrid, Milan, Paris and Prague. In order to analyse the performance and reliability of the GTS, the testbeds were created using two different scenarios: 1. With the virtual machines located in different cities 2. With the virtual machines located in the same city. Based on this methodology each testbed consists of two virtual machines. Given the fact that the GTS facilities are in nine locations across the Europe, a total of 45 testbeds were created.

Table 4.12: GTS performance and reliability testing

4.4.3 Results

The security assessment confirmed that the GTS system is properly configured; correct user management, user permissions, file permissions and system maintenance policy are in place. The tests on the GTS system confirmed that it is capable of performing its basic functions in a reliable and secure manner. The software development workflow is mature and takes advantage of a variety of supporting tools for continuous integration and build management. Some minor quality and security issues found in the software code that might require attention, alongside proposed solutions, were described in the final report presented to the development team.

4.5 Multidomain VPN Service Inventory (MD-VPN SI)

4.5.1 Overview

The Multi-Domain Virtual Private Network Service Inventory (MD-VPN SI) was developed to collect accurate and reliable information about infrastructural components that are required for MD-VPN service delivery and to ensure that all Network Service Providers (NSPs) – Transport and VPN providers – understand MD-VPN infrastructure topology. The MD-VPN SI is a tool for storing, updating and presenting data about all infrastructure components of the MD-VPN service and their relationship to each other.

SA2 T1 supported the MD-VPN SI service transition to production. The initial transition process was accomplished in GN4-1. Within the scope of GN4-2, validation and testing of the MD-VPN SI service update focused on security aspects, i.e.:

- New and updated functionality which may present new vulnerabilities.
- Critical and major issues discovered in the previous validation and test phase.

4.5.2 Test Strategy

4.5.2.1 *Test Team*

Number of test engineers	6 experts representing 3 different NRENs: PSNC, CARNET, MREN
Number of test teams	2 teams, 2 people in every team (one person could be a part of more than one team)

Table 4.13: MD-VPN SI security assessment test team composition

4.5.2.2 *Security Assessment*

Detailed scope of the testing	• The source code for the analysed version of the MD-VPN SI was obtained on 6 June 2017 and no modifications entered after that date by the SDT could be included in the assessment. • The vulnerability detection was performed on the MDVPN-SI production service. Vulnerability detection included testing of the following elements: ○ User Web interface components ○ Server components
Used programming languages	Java + external software libraries (activation, bcmail, bcprov, bctsp, commons-codec, commons-logging, freemarker, gson, itext, etc.)
Lines of code (pure - without comments, empty lines etc.)	9262

Table 4.14: MD-VPN SI security assessment

4.5.3 Results

The results showed that the security level of MD-VPN SI is solid. No major or critical issues were found. The security code review showed that authentication and authorisation mechanisms had been implemented properly. Some minor improvements were suggested for further releases.

4.6 eduroam Managed IdP

4.6.1 Overview

eduroam Managed IdP is a new eduroam-related service that is under development in GN4-2 JRA3. The service enables institution administrators to create credentials for their users to access eduroam, and to host the function of the eduroam Identity provider.

eduroam Managed IdP is in the pilot stage and will be starting its transition to the production environment in near future. In order to optimise the validation and testing process, the development and operations teams have agreed to initiate the validation and testing of the stable code as well as vulnerability detection. Since eduroam Managed IdP is implemented as a new feature of the eduroam Configuration Assistant tool (CAT), it was agreed that the scope of the testing should also cover the CAT software.

4.6.2 Test Strategy

4.6.2.1 *Test Team*

Number of test engineers	2 experts representing 2 different NRENs: CARNET, PSNC
Number of test teams	2 teams, 2 people in PSNC team, 2 people in CARNET team

Table 4.15: eduroam Managed IdP security assessment test team composition

4.6.2.2 *Security Assessment*

Detailed scope of the testing	• eduroam CAT source code with special emphasis on code implementing eduroam Managed IdP. • The vulnerability detection was performed on the test instance of CAT including the eduroam Managed IdP service. Vulnerability detection includes testing of user web interface components.
Used programming languages	Mostly PHP, JS and some shell scripting languages to a smaller extent. PHP code has been analysed.
Lines of code (pure - without comments, empty lines etc.)	PHP code – 27 201 lines.

Table 4.16: eduroam Managed IdP security assessment

4.6.3 Results

The code appears to be solid, and no security flaws were revealed during the testing. In fact, from the security point of view the quality of the source code was assessed as being high and exemplary. No critical or major issues were found.

4.7 Summary

Each quality evaluation presented in the above sections engaged an independent team of SA2 T1 experts from the software, security and system administration domain. All tests cases were carefully planned, designed and executed in short cycles according to the SA2 T1 test strategy. Only minimal and necessary involvement of the development team was required during testing. Moreover, access to test results was restricted by default to the service manager and development team lead, according to the general agreements made during the planning phase. Any issues, incidents, problems, errors and risks identified were recorded and assessed by degree of severity and priority according to the defined schema [GN4-1-D8.1]. Development teams were immediately informed in the case where any critical issues were found.

The final reports from each testing project, including causes of detected issues and recommendations towards solving problems or improving products, were submitted to the relevant development teams, service managers and optionally other appointed stakeholders. In every case the documentation was drawn up in accordance with the SA2 template (Appendix A), to ensure consistency in reporting any defects.

5 Conclusions

During the reporting period, SA2 T1 successfully supported three transitions to SA2 production and three quality assurance cases. Its corresponding KPI – *Number of services/products that went through audit, validation or transition* – with a target number of four, has therefore been exceeded.

The deployment of a consistent process for the transition to SA2 production has brought several benefits. The validation and testing process in place ensures that the warranty and to some extent the utility of production services is addressed. The SA2 test team as a neutral body assesses the overall service quality and contributes to the validation and testing process independently from the development activity. This assessment provides additional input, including the design of a service roadmap based on recommendations on essential enhancements and bug fixes, to assist key service stakeholders in making business decisions. Moreover, the existing SA2 T1 Test Strategy covers functional as well as non-functional requirements to fully meet the expected quality criteria.

SA2 T1 has implemented important improvements to the transition process in terms of conformity to best practices and standards. Users and stakeholders expect a service not only to be robust and fully tested, but also to meet any existing regulatory constraints. Procedural standardisation based on established standards and best practices further enhances the comparability of test results for evaluated projects. The roles and responsibilities in the service transition are clarified by the definition of teams and actors in accordance with standards and well-known best practices.

Even though its KPI target for GN4-2 has already been achieved, SA2 T1 will continue to support service transitions to SA2 production for the remainder of the project and guarantee that all future services that successfully pass the transition phase are of acceptably high quality.

Work is underway to gather the requirements of development teams for other types of tests that could be applied in earlier phases of the projects. This is usually referred to as Shift Left Testing and is seen as a necessity for teams that use agile development methodologies. In such cases, tests might be executed against incompletely designed artefacts thus leading to the need to adapt test procedures, eventually resulting in new test types being defined.

Appendix A Test Report Templates

The following template tables are used in transition processes for all types of testing to summarise detailed results. The rows included consider the general requirements of all test sub-teams, and any that are not needed may be left empty or removed so that the resulting tables may differ depending on the type of tests they are used for.

A.1 Table Template to Report Single Issue

ID	➢ as regex: [A-Z]+[0-9]{3}, ➢ which is a prefix, then 3-digit number. ➢ prefix determines a type of issue, e.g. PERF for performance or SEC for security, ➢ other proposed prefixes: USE (usability), CODES (code security), CODEQ (code quality), ➢ 3 digits is enough to address all found issues of a particular type, 2 digits could be too few in some cases, ➢ examples: PERF001, SEC999.
URI or Name / Description	(optional) Include the web link if available otherwise row should not be included
Issues / Threats	Can contain a detailed description in any format, pictures, automated analysis raw data and/or threats.
Severity	• CRITICAL • MAJOR • MEDIUM • MINOR
Recommendations	recommendations to solve / mitigate the issue
Steps	Can be steps to reproduce or stack trace

Number	(Optional Row) Number of issues
Type	(Optional) normally for code review example: Info / Warning / Error
Appeared on	(Optional) normally for code review example: appeared on Line xyz or date for documents

A.2 Table Template to Report Multiple Issues

Issue	Severity	Recommendation
Issue 1 (brief description)	• CRITICAL • MAJOR • MEDIUM • MINOR	recommendations to solve / mitigate issue
Issue 2 (brief description)	• CRITICAL • MAJOR • MEDIUM • MINOR	recommendations to solve / mitigate issue
Issue 3 (brief description)	• CRITICAL • MAJOR • MEDIUM • MINOR	recommendations to solve / mitigate issue

Appendix B Test Tools

The tables below list the tools used during the tests' execution.

Security tools are divided into two groups: 'Generic' and 'Specific'. The first comprises tools that have been used in all or almost all assessments. These are for the most part tools that are used during penetration testing as well as an auxiliary tool to measure the source code parameters. The second group consists of tools specific to certain areas, e.g. software technology.

Multiple versions may be included as during the timeframe of all security assessments new editions of the relevant tools were issued.

B.1 Security Assessment (Generic Tools)

Tool	Version(s)	Description
Acunetix Web Vulnerability Scanner	10.5, 11.0.173271618	Used for automated scanning and detection of security vulnerabilities on the web interface. Link: https://www.acunetix.com/
Arachni	1.5.1	Used to assess the security of web applications. Link: http://www.arachni-scanner.com/
Burp Suite Professional v1.7.30	1.7.24, 1.7.30	Used for tracking requests or responses, for identification of vulnerabilities and to verify attack vectors for Web applications. Link: https://portswigger.net/burp
cloc	1.66, 1.74	An auxiliary tool to count the number of lines in the source code per programming language (distinguishing also comments, empty lines etc.). Links: http://cloc.sourceforge.net/, https://github.com/AlDanial/cloc
Nexpose	6.4.20, 6.4.51	Used for automated scanning and detection of security vulnerabilities on the server. Link: https://www.rapid7.com/products/nexpose/
Nessus	6.11.1	Used for automated scanning and detection of security vulnerabilities. Link: https://www.tenable.com/products/nessus/nessus-professional
Nikto	2.1.6	A Web server scanner which performs comprehensive tests against Web applications.

		Link: https://cirt.net/Nikto2
Nmap	7.40, 7.50, 7.60	Used for quick detection of remotely available communication ports and services running on each port and for reporting basic known misconfigurations. Link: https://nmap.org/
Qualys SSL Labs	1.29.2	Used for analysis and identification of SSL configuration issues Link: https://www.ssllabs.com/ssltest/
Skipfish	2.10b	Active Web application security reconnaissance tool. Link: https://github.com/spinkham/skipfish
Tamper Data	11.0.1.1	Used for tracking request or responses and security testing of a Web application. Link: https://addons.mozilla.org/pl/firefox/addon/tamper-data/
ZAP	2.6.0	Used for automated scanning and detection of security vulnerabilities on the Web interface. Link: https://www.zaproxy.org/

B.2 Security Assessment (Specific Tools)

Tool	Version(s)	Used in	Description
Bandit	0.14.0	Firewall on Demand	A tool designed to find common security issues in Python code. Link: https://github.com/openstack/bandit
cppcheck	1.75	CAT GEANTLink installer	Static code analysis tool for C/C++ programming languages that detects bugs, undefined behaviour and dangerous coding constructs. Link: http://cppcheck.sourceforge.net/
Flake	2.4.1	Firewall on Demand	Python source code analyser and style guide checker. Link: https://pypi.python.org/pypi/flake8
PHP_Code Sniffer	3.0.0	eduroam Managed IdP	Static source code analyser for PHP-based code. Link: https://github.com/squizlabs/PHP_CodeSniffer
PHPMD	2.5.0	eduroam Managed IdP	Static source code analyser for PHP-based code. Link: https://phpmd.org/
Pylint	1.4.4	Firewall on Demand	Source code, bug and quality checker for the Python programming language. Link: https://www.pylint.org

RATS	2.3	eduroam Managed IdP, Firewall on Demand	A multilanguage static source code analyser for C, C++, Perl, PHP, Python and Ruby source code. Link: https://github.com/andrew-d/rough-auditing-tool-for-security
Visual Code Grepper	2.1.0.0	GTS	Java source code scanner, as an auxiliary result the statistics of the analysed source code were also obtained. Link: https://sourceforge.net/projects/visualcodegrepp/
Xanitizer	3.0.0	GTS, MDVPN SI	Java source code scanner. Link: https://www.rigs-it.com/index.php/product.html

B.3 Quality Code Audit

Tool	Version(s)	Description
Statistics	2.7.1	shows files sorted by their extension along with size, line count LOC etc.
FindBugs	1.3.6	an Intellij IDEA plugin to manage code quality by automatic code analysis.
PMD	1.3.7	an Intellij IDEA plugin to manage code quality by automatic code analysis.
Checkstyle	1.3.4	an Intellij IDEA plugin to manage code quality by automatic code analysis.

B.4 Other Tools

Tool	Version(s)	Used in	Description
iPerf	2.0.5	GTS	Network performance testing.
DbSchema	7.3.0	GTS	Tool to reverse engineer the database schema and view it as ER diagrams.
WCAG-EM Report Tool	1.1.0, 2016-3-16	GTS	Website Accessibility Evaluation Report Generator – helps follow the steps of WCAG-EM and generate a structured report from the input provided.

			Link: https://www.w3.org/WAI/eval/report-tool
WAVE Evaluation Tool	1.0	GTS	Evaluate web accessibility within the Chrome browser. Link: https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpkjmmeebjpijfedlgcdilocofh
Nu Html Checker	16.6.29	GTS	HTML validation: The Nu Html Checker Link: https://github.com/validator/validator
W3C CSS Validator	2017-01-07	GTS	CSS validation: W3C CSS Validator Link: https://jigsaw.w3.org/css-validator/
License Gradle Plugin	0.13.1	GTS	License Gradle Plugin used to perform license check of depending libraries Link: https://github.com/hierynomus/license-gradle-plugin

References

[GDPR]	http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679
[GN4-1-D8.1]	Deliverable *D8.1 Service Validation and Testing Process*
[ISTQB]	International Software Testing Qualifications Board
[IPRPolicy]	GÉANT IPR policy
[ISO/IEC 25023]	https://www.iso.org/standard/35747.html
[ISO/IEC 26514]	https://www.iso.org/standard/43073.html
[ITIL-FH]	*ITIL Foundation Handbook*
[ITIL-SDP]	*ITIL Service Design Package*
[License_Gradle]	https://github.com/hierynomus/license-gradle-plugin
[perfsonar.net]	https://www.perfsonar.net/
[PLM]	https://services.geant.net/sites/plm/Pages/home.aspx
[QABP]	GN3 Quality Assurance Best Practice Guide 4.0
[SA2-SDT]	SA2 Service Documentation Template
[SWArchBP]	GN3 Software Architecture Strategy Best Practice Guide 4.0
[SWDevBP]	GN3 Software Developer Best Practice Guide 4.0
[SWDocBP]	GN3 Software Documentation Best Practice Guide
[VD-ASGitBM]	Vincent Driessen “A successful Git branching model”
[WCAG]	https://www.iso.org/standard/58625.html

Glossary

CAT	Configuration Assistant Tool
EAP	Extensible Authentication Protocol
EAP-TTLS	Extensible Authentication Protocol Tunneled Transport Layer Security
GDPR	The EU General Data Protection Regulation
GN4-2	GÉANT project iteration GN4-2
IDP	Identity Provider
IP	Intellectual Property
IPR	Intellectual Property Rights
LOC	Line(s) of Code
NREN	National Research and Education Network
OWASP	Open Web Application Security Consortium
PAP	Password Authentication Protocol
PHP	Personal Home Page (programming language)
RADIUS	Remote Authentication Dial In User Service
SA	Service Activity in GÉANT project
SA2	GÉANT project activity that provides operations and delivery of Trust and Identity and Multi-Domain services
SA2 T1	SA2 Service Transition and Software Management Tasks
SDLC	Software/Security Development Life Cycle
SDT	Software Development Team
SQL	Structured Query Language
SSL	Secure Socket Layer
SVT	Service Validation and Testing
TLS	Transport Layer Security
WCAG	W3C Web Content Accessibility Guidelines (WCAG) 2.0
XML	eXtensible Markup Language