Horizon 2020 Programme

H2020-SU-TDS-02-2018
Trusted digital solutions and Cybersecurity in Health and Care



DATA-PROTECTION TOOLKIT REDUCING RISKS IN HOSPITALS AND CARE CENTERS

Project Nº 826284

ProTego

D7.3 *Report on the intermediate results of experiments and demonstrations*

Responsible:	Inetum (GFI INFORMÁTICA before January 2021)
Contributors:	All partners with the leadership of Inetum
Document Reference:	D7.3. Report on the intermediate results of experiments and demonstrations
Dissemination Level:	Public
Version:	Version 1.0
Date:	31/03/2021

Executive summary

This document describes the trial tests performed for all the technical components for Risk Assessment (WP4) and Risk Mitigation (WP5) assembled and integrated into the ProTego Toolkit (WP6).

These tests have been carried out in the testbeds prepared by Ospedale San Raffaele and Marina Salud hospitals, integrating the toolkit into their respective case studies.

The trials described in this document are based on the tests described in deliverable D2.2 and the results provide feedback for deliverable D2.3 to make the necessary adjustments.

Contributors Table

DOCUMENT SECTION	AUTHOR(S)	REVIEWER(S)
I.	Luis Carrascal (Inetum)	Dave Singelee (KUL), Colin Upstil (ICE), Antonio J. Gamito (Inetum), Alexis Oviedo (Inetum), Fernando Rendon (Inetum)
II.	Pietro Vismara (OSR), Michele Cantarutti (OSR), Salvador Garcia (MS), Luis Carrascal (Inetum)	Dave Singelee (KUL), Colin Upstil (ICE), Antonio J. Gamito (Inetum), Alexis Oviedo (Inetum), Fernando Rendon (Inetum)
III.	Pietro Vismara (OSR), Michele Cantarutti (OSR), Salvador Garcia (MS), Luis Carrascal (Inetum)	Dave Singelee (KUL), Colin Upstil (ICE), Antonio J. Gamito (Inetum), Alexis Oviedo (Inetum), Fernando Rendon (Inetum)
IV.	Pietro Vismara (OSR), Michele Cantarutti (OSR), Salvador Garcia (MS), Luis Carrascal (Inetum)	Dave Singelee (KUL), Colin Upstil (ICE), Antonio J. Gamito (Inetum), Alexis Oviedo (Inetum), Fernando Rendon (Inetum)
V.	Pietro Vismara (OSR), Michele Cantarutti (OSR), Salvador Garcia (MS), Luis Carrascal (Inetum)	Dave Singelee (KUL), Colin Upstil (ICE), Antonio J. Gamito (Inetum), Alexis Oviedo (Inetum), Fernando Rendon (Inetum)
VI.	Pietro Vismara (OSR), Michele Cantarutti (OSR), Salvador Garcia (MS), Luis Carrascal (Inetum)	Dave Singelee (KUL), Colin Upstil (ICE), Antonio J. Gamito (Inetum), Alexis Oviedo (Inetum), Fernando Rendon (Inetum)

Table of Contents

I. INTRODUCTION 8

I.1. PURPOSE AND ORGANIZATION OF THE DOCUMENT 8
I.2. SCOPE AND AUDIENCE 8

II. INFRASTRUCTURES FOR TESTING ENVIRONMENTS 9

II.1. OSPEDALE SAN RAFFAELE 9
II.2. MARINA SALUD 11

III. DESCRIPTION OF THE TESTS PERFORMED 14

III.1. ACCEPTANCE TEST 14
III.1.1. Acceptance Test 14
III.1.2. Experimental Procedure 18
III.2. USABILITY TEST 20
III.2.1. Usability Inspection 20
III.2.2. Experimental Procedure 20

IV. RESULTS 22

IV.1. ACCEPTANCE TEST 22
IV.2. USABILITY INSPECTION 28

V. CONCLUSIONS 30

VI. APPENDICES 31

VI.1. ACCEPTANCE TEST EVALUATION 31
VI.2. USABILITY INSPECTION EVALUATION 77
VI.3. TOOLKIT ASSEMBLING 82
VI.3.1. Generic Deployment Steps 83
VI.3.2. OSR 84
VI.3.3. Marina Salud 85

VII. REFERENCES AND INTERNET LINKS 86

VII.1. INTERNET LINKS 86

Table of Figures

Figure II.1-1 OSR Testing Environment for FoodCoach 10
Figure II.2-1 MS Testing Environment for Pocket EHR 12
Figure III.1-1 Components, objectives, mission, impacts.....18
Figure III.1-2 Experimental Procedure Prioritization 19
Figure III.1-3 Acceptance Test Example 20
Figure III.2-1 ASQ Questionnaire 21

List of Tables

Table III.1-1 Acceptance Tests....14
Table IV.1-1 Acceptance Test....22
Table IV.1-2 Success Rate....26
Table IV.2-1 Usability Inspection....28

Table of Acronyms and Definitions

Acronym	Definition
APP	Application
AWS	Amazon Web Services
CA	Continuous Authentication
EC	European Commission
EC2	Elastic Compute Cloud
EHR	Electronic Health Records
EKS	Elastic Kubernetes Service
EMR	Electronic Medical Records
FHIR	Fast Healthcare Interoperability Resources
HL7	Health Level 7
IoT	Internet of Things
JWT	JSON Web Token
KMS	Key Management Service
MS	Marina Salud
Mxx	Month xx (of the project)
NA	Not Applicable
OSR	Ospedale San Raffaele
RKE	Rancher Kubernetes Engine
SaaS	Software as a Service
SIEM	Security Information and Event Management
SSM	System Security Modeller
TLS	Transport Layer Security
VPC	Virtual Private Cloud
VPN	Virtual Private Network

I. INTRODUCTION

I.1. Purpose and organization of the document

This is the third deliverable (D7.3) released by ProTego WP7 and provides an overview of the ongoing trials and planning for upcoming trials. It consists of the result of activities performed in this work package (Task 7.3).

The input for this deliverable was initially scheduled to be executed from M20 to M25, and the objective of this task is to setup and run trial scenarios, and go further in testing and validating new results from the other WPs. Nevertheless, during the course of the tests many issues were found that prevented the components to work together as expected. That caused a delay for this deliverable until all the issues were solved.

The major reason for these issues came as a consequence of using container technologies in the deployment. The initial tests of the different components in isolation, as reported on deliverable D7.2, were not performed using the aforementioned technologies.

This has implied reinstalling Kubernetes clusters in the testing environments, redeploying the toolkit, and testing everything several times. Nevertheless, this has not affected the development of the project since all this has been performed in parallel to the other tasks that were scheduled.

The preliminary experiments have evolved from working in isolation to validate the integrated components working together. In addition, lab-level experiments have evolved to initial real-field trials. The results will serve as feedback to the other WPs for final refinements.

The deliverable consists of three major parts: the first describes the environments where the tests have been carried out, the second describes the tests performed, and the third describes the results found.

The appendices describe additional information related to the tests performed that provide a deeper insight on the trials, and is considered to have a special relevance.

Based on the tests described in deliverable D2.2, there are two different objectives for the tests performed:

- Evaluate Acceptance test success rate, that is, the fraction of implemented scenarios over the total amount of specified scenarios.
- Evaluate Usability, that is, the extent to which it can be used by specified users with effectiveness, efficiency, and satisfaction in a specific context of use.

I.2. Scope and audience

This deliverable is considered as a public document to the ProTego project. It provides a description of the status quo of ongoing trials.

This deliverable is primarily intended for EC Officers and project-appointed reviewers. Further, the document is relevant for project members specially those involved in WP2.

II. INFRASTRUCTURES FOR TESTING ENVIRONMENTS

This section describes the testing environments that have been used in each case study.

II.1. Ospedale San Raffaele

This case study uses an existing application that has been modified to use the ProTego Toolkit. This application is FoodCoach, a food recommendation system that suggests personalized nutrition plans for end-users. It represents a traditional archetypal distributed application that allows out of boundary data exchange, opening up to the usage of personal mobile devices.

The testers installed the ProTego toolkit as a preliminary activity in order to be able to test it. In preparation for the installation process, the necessary infrastructure was setup. The first step involved the creation of four virtual machines according to the provided specifications. In particular, they were created in the same subnetwork, and each of them was given a static IP address.

The services of the ProTego Toolkit run in Docker containers. Kubernetes was installed in order to orchestrate these containers, and then Rancher was installed to add a complete UI and workload management layer to Kubernetes. By using the Rancher interface, it was possible to deploy the services of the ProTego Toolkit. Once the ProTego toolkit was setup and running, the medical applications could use the exposed services. Figure II.1-1 depicts the configured environment used during the testing activities.

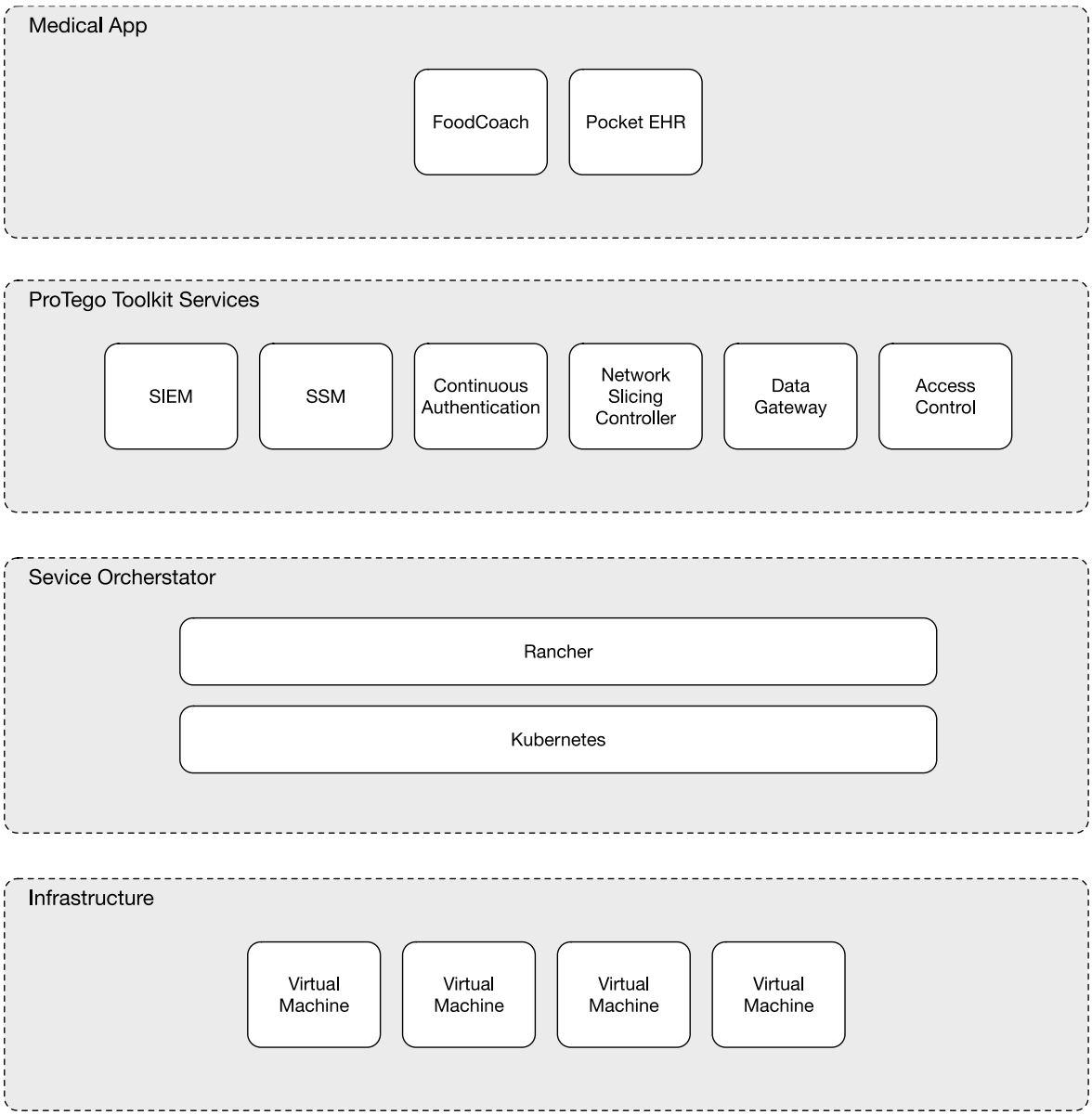


Figure II.1-1 OSR Testing Environment for FoodCoach

II.2. Marina Salud

This case study is more complex than the previous since it involves a hybrid cloud deployment. Therefore, not all the ProTego Toolkit components have been tested in this case and they will continue to be tested after the release of this deliverable. The results of those tests will be reported in deliverable D7.4. By the work performed so far, the following basic components, configuration, and functionalities, have been setup:

1. Medical APP frontend: The Pocket EHR mobile application has been developed, which allows patients to enter health parameters of interest to follow up after a heart failure hospitalization process. The operation of an IoT device has been simulated through the use of a Raspberry Pi device, that sends vital parameters automatically. Pocket EHR, in addition, allows doctors to make complex queries to the Data Gateway, obtaining indicators over all patients on the platform (patients with alert synthoms, etc.)
2. Cloud premises: The interfaces to the Internet (API Gateway) and the backend of the mobile application (Lambda functions) have been developed. A Kubernetes cluster has also been created based on AWS EKS service, and managed by an instance of Rancher. Within the cluster, the pods for the Data Gateway and KMS have been deployed in an isolated way using AWS VPC, so that they are only accessible from Lambda functions, which minimizes the exposure of the ProTego Toolkit components, thus minimizing the attack surface as well.
3. Hospital premises: Integration routes have been developed in Rhapsody, a commercial Integration Engine. They allow to make requests to the ProTego Data Gateway to retrieve data entered by patients using Pocket EHR, and carry out the transformation from FHIR to HL7 2.3, to integrate that data into the patient's EMR in the hospital, implemented through CERNER Millenium Clinical History system, one of the most advanced Medical Record systems in the market.

In this way, it is possible for patients, either through the Pocket EHR mobile APP or through an IoT device, to enter data into the ProTego Data Gateway installed in the cloud environment, and this information can be integrated into the patient's medical record. The hospital premises are not exposed to the Internet, and the ProTego services are isolated and can only be requested from the Lambda functions in the cloud premises.

The image below depicts the current progress of the use case, illustrating the pending work as well:

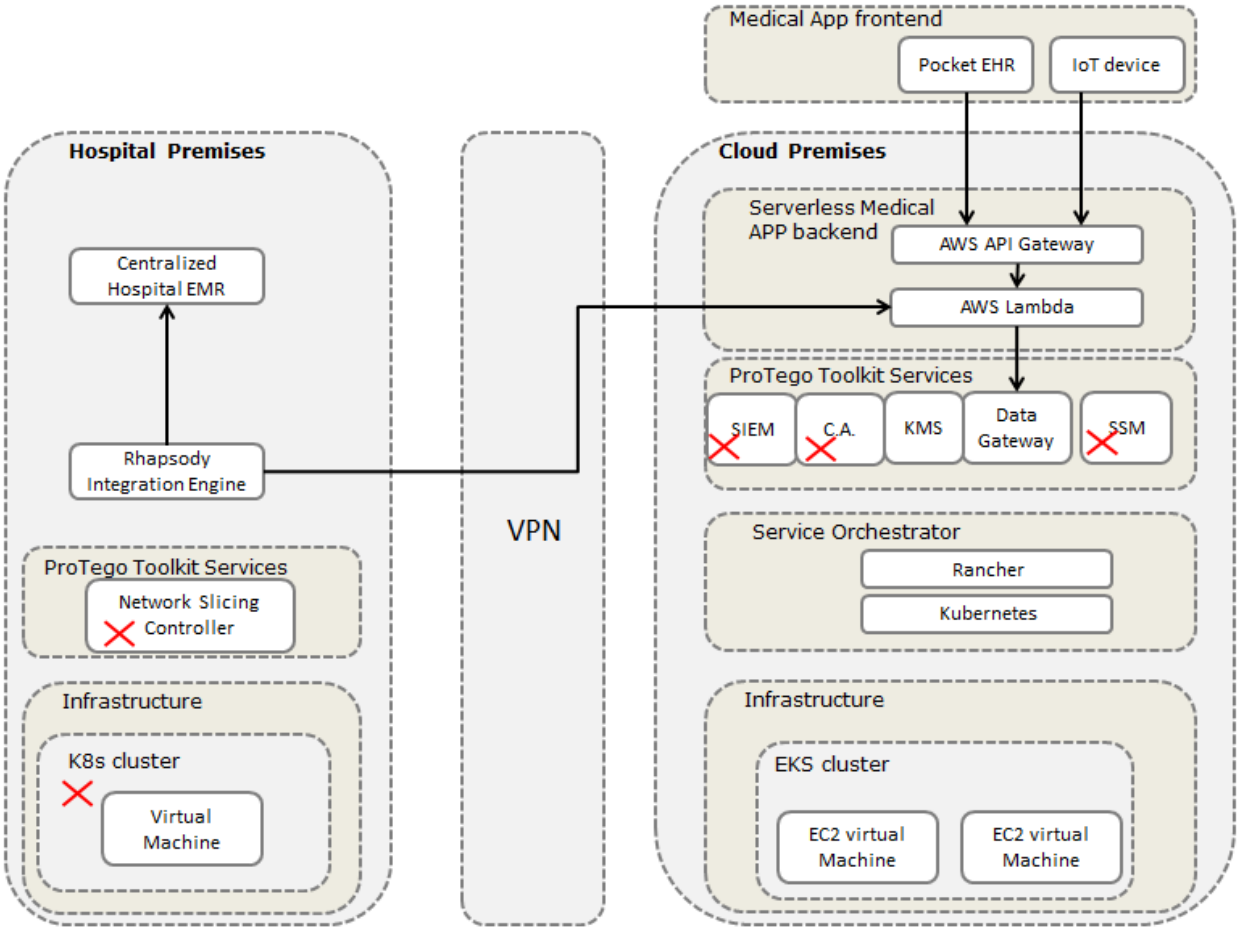


Figure II.2-1 MS Testing Environment for Pocket EHR

Acting as a ProTego adopter, Marina Salud provides the underlying infrastructure and logic elements outside the ProTego Toolkit:

- Cloud provider account, as the repository to all the elements outside the hospital premises.
- Pocket EHR frontend plus emulated IoT device.
- Pocket EHR serverless backend.
- EKS cluster, managed by Rancher.
- Isolation of the EKS cluster (and services) through VPC.
- VPN between the hospital and cloud premises.
- Standard base logic to integrate the ProTego Toolkit managed data into a commercial EMR system.

The ProTego Data Gateway and KMS were deployed in the EKS cluster offering the exposed services to the backend of the Pocket EHR application.

The items marked with an X in the image above will be tested in later stages of the project:

- Deploy and configure the SIEM into cloud premises.
- Deploy the SSM into cloud premises. Nevertheless, although not yet deployed in the testing environment, but available as SaaS, the SSM has been used to perform an in-depth Risk Analysis of this case study.
- Integration between the SSM and the SIEM to recalculate risk levels automatically.
- Continuous Authentication integration:

  - Frontend: Pocket EHR should register the user's token into the CA frontend. This represents a challenge since they are not developed using the same or compatible programming language.
  - Backend: Before any request is made from the Pocket EHR frontend to the Data Gateway, CA will validate that the user represented by the token has not been compromised. This logic has been validated in a development environment and will further be incorporated into the cloud premises.

- The Network Slicing Controller will be deployed in the hospital premises to use this technology in wireless LANs. So far this has only been tested in a lab environment and worked as expected.

III. DESCRIPTION OF THE TESTS PERFORMED

This section reports the methods used by testers for the validation of the ProTego toolkit. The first subsection describes the Acceptance test and the relative experimental procedure carried out in order to execute this test. The second subsection focuses instead on the description of Usability Inspection adopted to evaluate the system usability.

III.1. Acceptance Test

III.1.1. Acceptance Test

The acceptance tests constitute the portion of the acceptance criteria utilized to evaluate the implemented functionalities and their associated quality attributes. Each Acceptance test is composed by a set of scenarios. Each scenario describes through an example how a use case should function in the executable software [1].

For convenience, Table III.1-1 reports the Acceptance test and related scenarios evaluated in this deliverable.

Table III.1-1 Acceptance Tests

Acceptance test	Scenario
AT01: Register user	S01.1: Administrator registers a new user
	S01.2: Insufficient authorization
	S01.3: User identifier already taken
AT02: Conduct first-time risk assessment	S02.1: Security operator assesses risks
	S02.2: Insufficient authorization
	S02.3: First time assessment already conducted
AT03: Assess prospective risks to the infrastructure	S03.1: Security operator assesses application risks
	S03.3 Insufficient authorization
	S03.3: First time risk assessment is still to be conducted
AT04: Register application	S04.1: System operator registers a new application
	S04.2: Insufficient authorization
	S04.3: Already existing application
AT05: Configure application network slices	S05.1: Security operator configures network slices
	S05.2: Insufficient authorization

	S05.3: Network slices are already configured
	S05.4: Non-existing application
AT06: Configure application logging mechanism	S06.1: Security operator specifies logging mechanism
	S06.2: Insufficient authorization
	S06.3: Logging mechanism is already configured
	S06.4: Non existing application
AT07: Specify application access control	S07.1: Security operator specifies access control
	S07.2: Insufficient authorization
	S07.3: Application access control is already active
	S07.4: Non existing application
AT08: Specify application query templates	S08.1: Data operators specifies application query templates
	S08.2: Insufficient authorization
	S08.3: Application already has associated query templates
	S08.4: Non existing application
AT09: Store initial medical data	S09.1: Data operator stores initial medical data
	S09.2: Insufficient authorization
	S09.3: Application initial resources are already stored
	S09.4: Non existing application
AT10: Register IoT device	S10.1: Network operator registers an IoT device
	S10.2: Insufficient authorization
	S10.3: IoT device already registered
	S10.4: Invalid identity certificate
AT11: Register mobile device	S11.1: Network operator registers a new mobile device
	S11.2: Insufficient authorization
	S11.3: Mobile device is already registered
	S11.4: Network slices are not configured
	S11.5: Non existing application

AT12: Store medical data	S12.1: Patient stores her own weight
	S12.2: Insufficient authorization
	S12.3: Resource with the same id already exists
	S12.4: Authorization token is expired
	S12.5: Authorization token is ill-formed
AT13: Retrieve medical data	S13.1: Nutritionist queries for some patient's weights
	S13.2: Insufficient authorization
	S13.3: Medical data have been tampered with
	S13.4: Non existing query
	S13.5: Authorization token is expired
	S13.6: Authorization token is ill formed
AT14: Assign IoT device to application user	S14.1: Nutritionist assigns a new IoT device
	S14.2: Nutritionist tries to assign a device to another's nutritionist patient
	S14.3: Admin tries to assign a device
	S14.4: IoT device is already assigned
	S14.5: Authorization token is expired
	S14.6: Authorization token is ill formed
AT15: Log custom application event	S15.1: Correct application key
	S15.2: Wrong application key
AT16: Establish secure communication	S16.1: IoT device establishes a connection
	S16.2: IoT device beyond the range of communication
	S16.3: Connection already established
	S16.4: Network glitch
	S16.5: Failed the proof-of-possession challenge
	S16.6: Unregistered device
AT17: Send medical data	S17.1: IoT device sends some medical data
	S17.2: IoT device gets disconnected
	S17.3: Session error

AT18: Report suspicious activity	S18.1: Mobile agent reports suspicious activity
	S18.2: Mobile device is not registered
AT19: Respond to alert	S19.1: Security operator responds to an alert
	S19.2: Insufficient authorization
AT20: Review alerts	S20.1: Security operator review alerts
	S20.2: Insufficient authorization
AT21: Review new risk evaluation	S21.1: Security operator reviews changes in the evaluation
	S21.2: Insufficient authorization
	S21.3: First time risk assessment is still to be conducted
AT22: Reflect infrastructure changes	S22.1: Security operator reflects recent changes
	S22.2: Insufficient authorization
	S22.3: First time risk assessment is still to be conducted

III.1.2. Experimental Procedure

After the initial setup, the testers followed the experimental procedure reported below:

1. **Prioritization**: In the course of the development of the ProTego Toolkit, some functionalities underwent a shift of priorities, as they became less relevant than initially envisioned. For this reason, the testers assigned to each scenario a priority, in order to highlight those functionalities most relevant to demonstrate the project objectives (shown in Figure III.1-1).

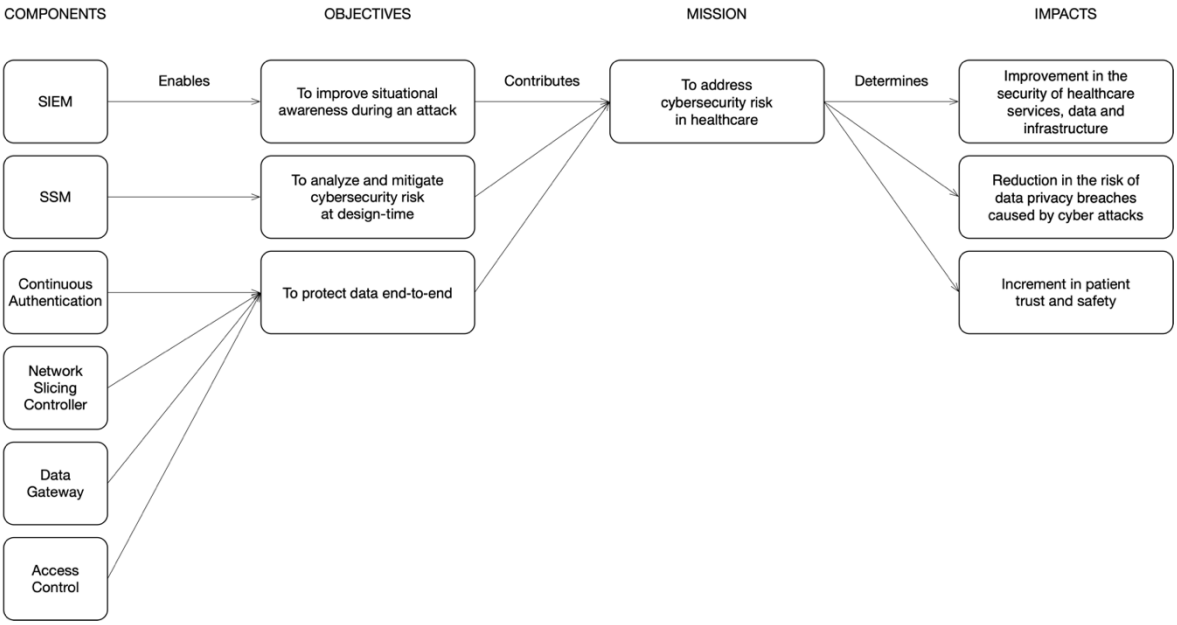


Figure III.1-1 Components, objectives, mission, impacts

Specifically, the testers prioritized the scenarios according to two dimensions: importance and urgency [2]. Every requirement can be considered as being either important to achieving project objectives or not so important, and as being either urgent or not so urgent.

As shown in the Figure III.1-2, these alternatives yield three possible combinations, which have been used to define a priority scale [1]:

- **High priority**: The specified functionalities are both important and urgent to achieve project objectives.
- **Medium priority**: The specified functionalities are important but less urgent to achieve project objectives.
- **Low priority:** The specified functionalities are neither important nor urgent to achieve project objectives. However, these scenarios reflect the functionalities that would be desirable in a market-ready solution.

Moreover, during the course of the project some functionalities no longer fit in the ecosystem of the project technologies. In light of this, these scenarios have been considered ***outdated*** and thus excluded by the evaluation.

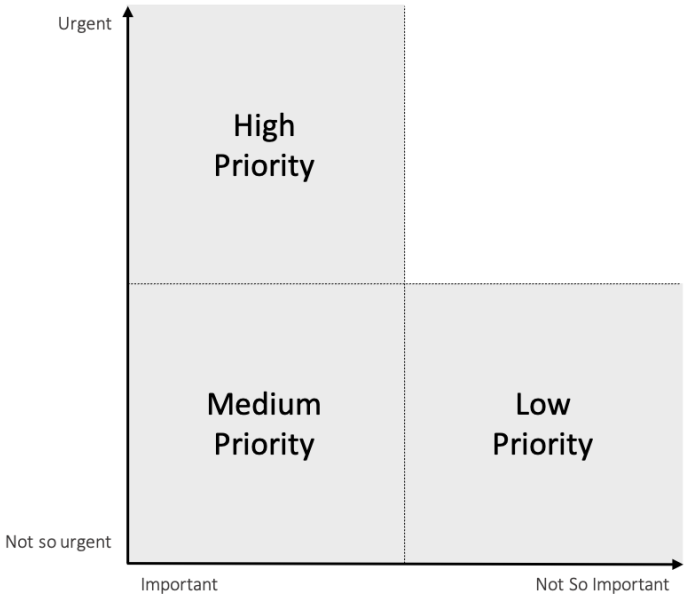


Figure III.1-2 Experimental Procedure Prioritization

2. **Execution**: The testers exercised ProTego functionalities, in order to verify for each scenario, the satisfaction required preconditions, the event or action causing the interaction, and the expected outcome resulting from such an interaction [3].
3. **Evaluation**: The testers assigned *Passed* evaluation to those scenarios in which the system behaviours corresponded to the behaviours described in the scenario. Otherwise, the testers assigned *Failed* as evaluation. Figure III.1-3 shows a *Passed* evaluation for the unsuccessful scenario “Network slices are already configured” of the Acceptance Test “Configure application network slices”. Some functionalities of the Toolkit were initially supposed to be provided by a centralized system. Because such a centralized system has not been developed, the responsibility of providing these functionalities was offloaded to the individual components of the Toolkit instead. In the light of this consideration, the following rationale was applied in order to assess the status of the scenarios of these specific cases:
    - If a certain functionality is individually provided by all the components that require it, the corresponding test is marked as *Passed*.
    - If a certain functionality is not provided by all the components that require it, the corresponding test is marked as *Failed*. For the sake of completeness, if at least some of the components provide that functionality, a list of those components is provided.

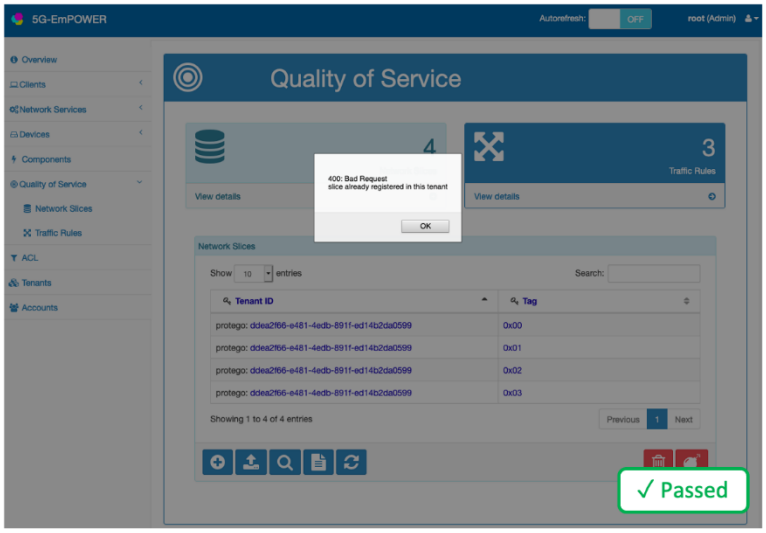


Figure III.1-3 Acceptance Test Example

4. **Analysis**: The testers calculated the success rate, that is, the percentage of *Passed* scenario over the total amount of scenarios. The results have been reported according to the priority scale above mentioned: High priority, Medium priority, Low priority.

III.2. Usability Test

III.2.1. Usability Inspection

In order to evaluate the user acceptance of ProTego, the testers adopted a Usability Inspection test [4]. The aim of this test is to identify and measure usability problems before to test with end-users. The outcome will serve as an input to improve the experience of use of the ProTego demonstrator for the final release. The usability inspection was conducted to validate, besides the Use Cases specified in D2.2, also the initial setup process.

III.2.2. Experimental Procedure

1. **Task selection:** In this phase, the testers excluded the tasks to be tested during the inspection according to three criteria:
    a. Automated: The actor of the use cases is not a human being, but rather an automated agent. For this reason, it is not possible to assess the task usability.
    b. Outdated: The task no longer fits in the ecosystem of the project technologies.
    c. Failed: The task is not sufficiently mature to be tested.

2. **Execution:** The tester responded to the After-Scenario Questionnaire [5] after performing the Use Cases' tasks. The questionnaire consists of the following three statements:
    - Q1 "I am satisfied with the ease of completing the tasks in this scenario.''
    - Q2 ''I am satisfied with the amount of time it took to complete the tasks in this scenario.''
    - Q3 ''I am satisfied with the support information (online help, messages, documentation) when completing the tasks.''

    Each statement is accompanied by a 7-point rating scale, from ''strongly disagree'' to ''strongly agree,'' as shown in Figure III.2-1 . The questionnaire touches on different

usability metrics: effectiveness (Q1), efficiency (Q2), support (Q3), and satisfaction (Q1, Q2, Q3).

		1	2	3	4	5	6	7	
1. Overall, I am satisfied with the ease of completing the tasks in this scenario	strongly disagree	○	○	○	○	○	○	○	strongly agree
2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario	strongly disagree	○	○	○	○	○	○	○	strongly agree
3. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks	strongly disagree	○	○	○	○	○	○	○	strongly agree
		1	2	3	4	5	6	7	

Figure III.2-1 ASQ Questionnaire

3. **Evaluation:** The effectiveness, efficiency and support scores were reported for each use cases examined. Satisfaction was calculated as the average score of Q1, Q2 and Q3. In addition, the testers described a set of qualitative suggestions with the aim of providing useful insights to improve usability for the final release.

IV. RESULTS

This section reports a summary of the results concerning the Acceptance Test and the Usability Inspection. For a full description refer to appendices VI.1. Acceptance Test Evaluation and VI.2. Usability Inspection Evaluation.

IV.1. Acceptance Test

Table IV.1-1 and Table IV.1-2 summarize the results concerning the prioritization and evaluation of the Acceptance test and relative scenarios. Those scenarios evaluated as outdated have been marked in grey. For a complete description of the results, refer to the section VI.1.

Table IV.1-1 Acceptance Test

Acceptance test	Scenario	Priority			Evaluation	
		High	Medium	Low	Passed	Failed
AT01: Register user	S01.1: Administrator registers a new user	●			✓	
	S01.2: Insufficient authorization	●			✓	
	S01.3: User identifier already taken		●		✓	
AT02: Conduct first-time risk assessment	S02.1: Security operator assesses risks	●			✓	
	S02.2: Insufficient authorization	●			✓	
	S02.3: First time assessment already conducted		●			✗
AT03: Assess prospective risks to the infrastructure	S03.1: Security operator assesses application risks	●			✓	
	S03.3 Insufficient authorization	●			✓	
	S03.3: First time risk assessment is still to be conducted		●		✓	
AT04: Register application	S04.1: System operator registers a new application					
	S04.2: Insufficient authorization					
	S04.3: Already existing application					
	S05.1: Security operator configures network slices	●			✓	

AT05: Configure application network slices	S05.2: Insufficient authorization	●			✓	
	S05.3: Network slices are already configured		●		✓	
	S05.4: Non-existing application		●		✓	
AT06: Configure application logging mechanism	S06.1: Security operator specifies logging mechanism	●			✓	
	S06.2: Insufficient authorization	●			✓	
	S06.3: Logging mechanism is already configured		●		✓	
	S06.4: Non existing application					
AT07: Specify application access control	S07.1: Security operator specifies access control	●			✓	
	S07.2: Insufficient authorization	●			✓	
	S07.3: Application access control is already active		●		✓	
	S07.4: Non existing application					
AT08: Specify application query templates	S08.1: Data operators specifies application query templates					
	S08.2: Insufficient authorization					
	S08.3: Application already has associated query templates					
	S08.4: Non existing application					
AT09: Store initial medical data	S09.1: Data operator stores initial medical data			●		✗
	S09.2: Insufficient authorization			●		✗
	S09.3: Application initial resources are already stored			●		✗
	S09.4: Non existing application					
AT10: Register IoT device	S10.1: Network operator registers an IoT device					
	S10.2: Insufficient authorization					
	S10.3: IoT device already registered					
	S10.4: Invalid identity certificate					
AT11: Register mobile device	S11.1: Network operator registers a new mobile device	●			✓	
	S11.2: Insufficient authorization	●			✓	

	S11.3: Mobile device is already registered		●		✓	
	S11.4: Network slices are not configured					
	S11.5: Non existing application					
AT12: Store medical data	S12.1: Patient stores her own weight	●			✓	
	S12.2: Insufficient authorization	●			✓	
	S12.3: Resource with the same id already exists					
	S12.4: Authorization token is expired		●			✗
	S12.5: Authorization token is ill-formed		●		✓	
AT13: Retrieve medical data	S13.1: Nutritionist queries for some patient's weights	●			✓	
	S13.2: Insufficient authorization	●			✓	
	S13.3: Medical data have been tampered with	●				✗
	S13.4: Non existing query					
	S13.5: Authorization token is expired		●			✗
	S13.6: Authorization token is ill formed		●		✓	
AT14: Assign IoT device to application user	S14.1: Nutritionist assigns a new IoT device	●				✗
	S14.2: Nutritionist tries to assign a device to another nutritionist's patient		●			✗
	S14.3: Admin tries to assign a device		●			✗
	S14.4: IoT device is already assigned		●			✗
	S14.5: Authorization token is expired		●			✗
	S14.6: Authorization token is ill formed		●			✗
AT15: Log custom application event	S15.1: Correct application key	●			✓	
	S15.2: Wrong application key	●			✓	
AT16: Establish secure communication	S16.1: IoT device establishes a connection	●			✓	
	S16.2: IoT device beyond the range of communication					
	S16.3: Connection already established					
	S16.4: Network glitch					

	S16.5: Failed the proof-of-possession challenge					
	S16.6: Unregistered device					
AT17: Send medical data	S17.1: IoT device sends some medical data	●			✓	
	S17.2: IoT device gets disconnected					
	S17.3: Session error					
AT18: Report suspicious activity	S18.1: Mobile agent reports suspicious activity	●			✓	
	S18.2: Mobile device is not registered		●		✓	
AT19: Respond to alert	S19.1: Security operator responds to an alert	●			✓	
	S19.2: Insufficient authorization	●			✓	
AT20: Review alerts	S20.1: Security operator review alerts	●			✓	
	S20.2: Insufficient authorization		●		✓	
AT21: Review new risk evaluation	S21.1: Security operator reviews changes in the evaluation	●				✗
	S21.2: Insufficient authorization	●				✗
	S21.3: First time risk assessment is still to be conducted		●			✗
AT22: Reflect infrastructure changes	S22.1: Security operator reflects recent changes	●				✗
	S22.2: Insufficient authorization		●			✗
	S22.3: First time risk assessment is still to be conducted		●			✗

Table IV.1-2 Success Rate

	# Scenarios	✓ Passed	✕ Failed	✓ Success Rate
Low Priority	3	0	3	0%
Medium Priority	22	11	11	50%
High Priority	31	26	5	84%
Total	56	36	17	64%

Generally speaking, the results show that 84% of the high priority acceptance test passed, meaning that the majority of the most important functionalities expected by ProTego have been successfully implemented. However, the medium priority scenarios have not reached the same level of maturity (50% of success rate), pointing out that the toolkit does not offer yet all the functionalities expected in a mature solution.

The SSM has been evaluated by exercising the following acceptance tests: AT02, AT03, AT22. The results show that the component has the capability of analysing a system (such as OSR and MS infrastructures) in order to identify potential threats and determine possible mitigation strategies. For example, in the case of OSR, the analysis showed that adding a password strength test to the Frontend service would decrease the potential threats related to this part of the infrastructure. At the moment, the integration between SSM and SIEM has not been finalised. Because of that, the component is not ready to demonstrate its capability to recalculate the risks in response to run-time changes.

The SIEM component has been evaluated with the following acceptance tests: AT06, AT15, AT18, AT19, AT20, AT21. The tested scenarios demonstrated that the component is able to monitor application logs, and that it is also able to combine the information gathered from those logs in order to infer a more comprehensive understanding of the nature of the threats to the system. This greatly improves run-time awareness during an attack situation. In some tests (like AT18), messages were sent to the SIEM manually, in order to simulate the receival of messages from other components of the Toolkit that are currently not ready for integration. This indicates that the SIEM can already receive and process these messages, and it is ready for integration.

The Continuous authentication component has been evaluated through AT18. Although mobile device security capability works in a non-integrated environment, some issues emerged during the integration activity. Because of that, the functionality has been tested by manually pushing a warning on Kafka, and by verifying that the SIEM received the corresponding notification. These results indicates that ProTego is not ready to demonstrate the capability of detecting suspicious activity from mobile device, while it is mature enough to demonstrate the capability of reacting accordingly in case of threats.

The Network Slicing component has been evaluated with the following acceptance tests: AT05 and AT11. The tested scenarios demonstrated that, once mobile devices were registered, they could connect to the hospital's network that had been set up for the tests. The tests showed also that it was possible to split the network into multiple slices which allowed to isolate network traffic, and which were individually configured to adjust their capacity as needed.

The Access Control & Key management has been evaluated through AT07. The test demonstrated that the component has the capability to specify the application access control by manually intervening on the policy. The access control rules are correctly enforced so that the component is able to control the access to the medical data stored within the healthcare infrastructure. This component manages the access to the data stored in the Data gateway and therefore the two components are tightly coupled.

The Data gateway has been evaluated with the following acceptance tests: AT12, AT13. The results show that the Access control and the Data gateway are properly integrated demonstrating

their capability of storing and retrieving the patient's records in ProTego. Indeed, the requests were handled correctly, thus accepting the authorized ones and rejecting those unauthorized. This functionality proved useful for the hospitals in preserving health data protection at rest.

The IoT solution has been evaluated with the following acceptance tests: AT10, AT14, AT16, AT17. In the course of the development, the functionality related to the registration of IoT devices has become outdated since a shift occurred with the adopted technologies. Indeed, this mechanism will be replaced with the introduction of a refresh token associated with the patient, which makes the registration of the IoT itself unnecessary. Consequently, this new mechanism will also be reflected in the assignment of IoT devices to application users. That being said, the IoT device was able to connect with the Data Gateway by using a JWT refresh token and thus, demonstrating its capability to establish a connection with ProTego.

IV.2. Usability Inspection

Table IV.2-1 summarizes the evaluation concerning the usability inspection. For a complete description of the results, refer to the section VI.2.

Table IV.2-1 Usability Inspection

	ID	TASK NAME	Excluded			Evaluated			
			Automated	Outdated	Failed	Effectiveness	Efficiency	Support	Satisfaction
Installation	IN01	Cluster deployment				5	5	6	5,3
	IN02	Install Data Gateway				4	4	5	4,3
	IN03	Install Network Slicing				5	4	7	5,3
	IN04	Install SIEM				6	5	6	5,6
	IN05	Install SSM				NA	NA	NA	NA
	IN06	Install Continuous authentication			•				
Configuration	AT01	Register user				NA	NA	NA	NA
	AT02	Conduct first-time risk assessment				6	6	6,5	6,2
	AT03	Assess prospective risks to the infrastructure				6	6	6	6
	AT04	Register application		•					
	AT05	Configure application network slices				6	6,5	5	5,8
	AT06	Configure application logging mechanism				5	5	6	5,3
	AT07	Specify application access control				5	5	5	5
	AT08	Specify application query templates							
	AT09	Store initial medical data							
	AT10	Register IoT device							
	AT11	Register mobile device				6	6	6	6

	AT12	Store medical data	●						
	AT13	Retrieve medical data	●						
	AT14	Assign IoT device to application user		●					
	AT15	Log custom application event	●						
	AT16	Establish secure communication	●						
	AT17	Send medical data	●						
	AT18	Report suspicious activity	●						
Monitoring	AT19	Respond to alert				6	6	6	6
	AT20	Review alert				6	6	6	6
	AT21	Review new risk evaluation			●				
	AT22	Reflect infrastructure changes			●				

Overall, the installation process was simple to follow. It required two phases: first, the installation of the cluster, and then the installation of the components in the cluster. The installation of the cluster was simple to follow, required an acceptable amount of time and was supported with the necessary resources. The installation of the components varied depending on the specific component: some of them required more time and effort than expected. Effectiveness and efficiency could be improved by providing more comprehensive tutorials and instructions.

In order to configure the ProTego toolkit, each of the components needed to be configured individually. The comprehensive tutorial of the SSM made the task effective and efficient to perform. The SIEM configuration was simple to perform, however, it can be improved by making it less coupled with the application. The configuration of the Network Slicing proved simple and intuitive, mostly thanks to its UI. The configuration of the access control policy was sufficiently clear, however it would be desirable to configure it in a more user-friendly way.

Concerning the infrastructure monitoring, the interaction with the SIEM provided satisfactory results, as the UI presented the information in a clear and concise way, thus allowing to monitor the application events.

V. CONCLUSIONS

As already expressed in IV.1. Acceptance Test most of the high priority functionality is already implemented and half of the medium priority functionality is developed too. The remaining functionality is currently under development and will be tested during M31 to M36, and the results of the testing will be reported in deliverable D7.4. Nevertheless, the work carried out already allows to draw some conclusions regarding the ProTego Toolkit:

- The selected technology for the integration and deployment of the components, based on the Kubernetes cluster, allows the use of the Toolkit in different technological environments, without the need to generate different versions to adapt to the differences in those environments. Specifically, the correct deployment and setup has been tested in a typical on-premise environment and in an environment in the AWS cloud as well, with satisfactory results in both cases.
- The Toolkit offers unique interfaces for accessing secured data, for use by different consumers: both applications developed in different languages and platforms, from Dart mobile applications and Java web applications) and for IoT devices, provided that the different data consumers adapt their logic for the use of JWT tokens.
- The Toolkit allows to delegate authentication in external systems based on JWT tokens. The correct operation has been verified in JWT tokens managed by authentication systems *Keycloak* and *AWS Cognito* with a satisfactory result.
- The Toolkit allows the management of data compatible with the HL7 FHIR standard.
- Since the Toolkit is based on FHIR, it allows the integration of the managed data in medical record systems on the market, where the organization only has to carry out conversions from the standard FHIR to the specific data format of the medical record system. As most current medical record systems are HL7 compliant, the mappings to be performed are based on two standard formats, making such data integration feasible.
- The Toolkit allows the application of measures in the design phase of the system, which is extremely important when designing secure systems. It also allows knowing what effects certain security measures would have on the global risk level, without the system being yet implemented.
- The deployment of the Toolkit in both pilots indicates that the support of ProTego partners is recommended during the deployment and start-up phase. This characteristic is common to most systems oriented to health organizations. After the deployment, the toolkit allows to update its components in an easy way that requires minimal interaction, thanks to tools such as Rancher and Helm.
- Besides providing a way to collect and analyse application logs, the Toolkit also allows to define custom rules to raise alerts in specific situations where a certain combination of events occur. For this reason, the Toolkit does not simply offer a generic approach to monitoring applications, but rather it allows to monitor them by targeting threats that are specific to them. Moreover, custom rules allow to infer additional information from the logs that would otherwise be only obtainable by constantly observing the logs. This functionality has been perceived as very valuable to improve situational awareness during an attack in the hospitals.
- The Toolkit provides a way to protect data in transit in the hospitals. The hospital network can be split into several slices, and these slices can each be given a specific configuration. This way, it is possible to ensure that a specific set of health data is securely exchanged on a specific slice. Moreover, each slice of the network can be assigned a certain priority, so that, for example, data travels faster in a specific slice.

VI. APPENDICES

VI.1. Acceptance Test Evaluation

AT01: Register user

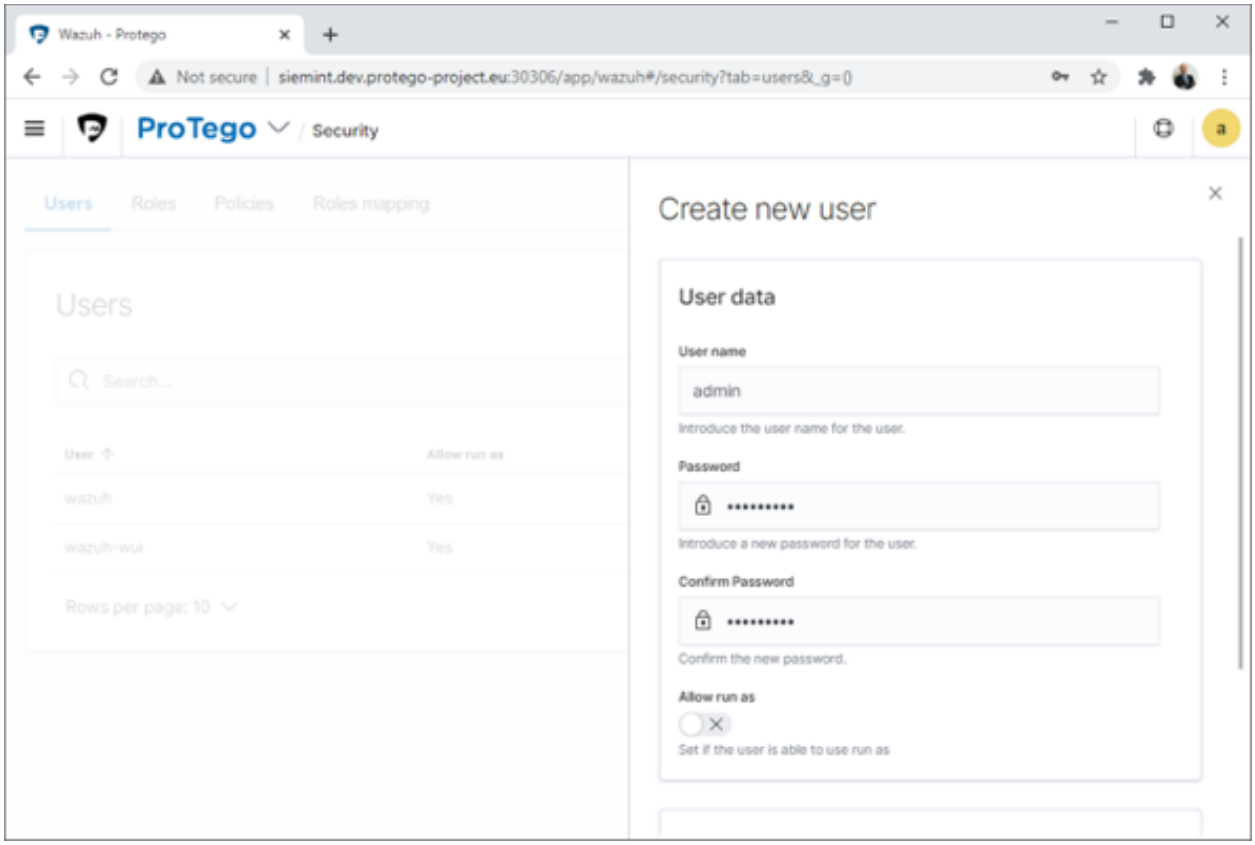
ID: S01.1

Name: Administrator registers a new user

Priority: Low

Evaluation: ✓ Passed

Description: There is not a central system in ProTego to register users. However, all the ProTego components have their own functionality to register users (the image shows the SIEM log in process).



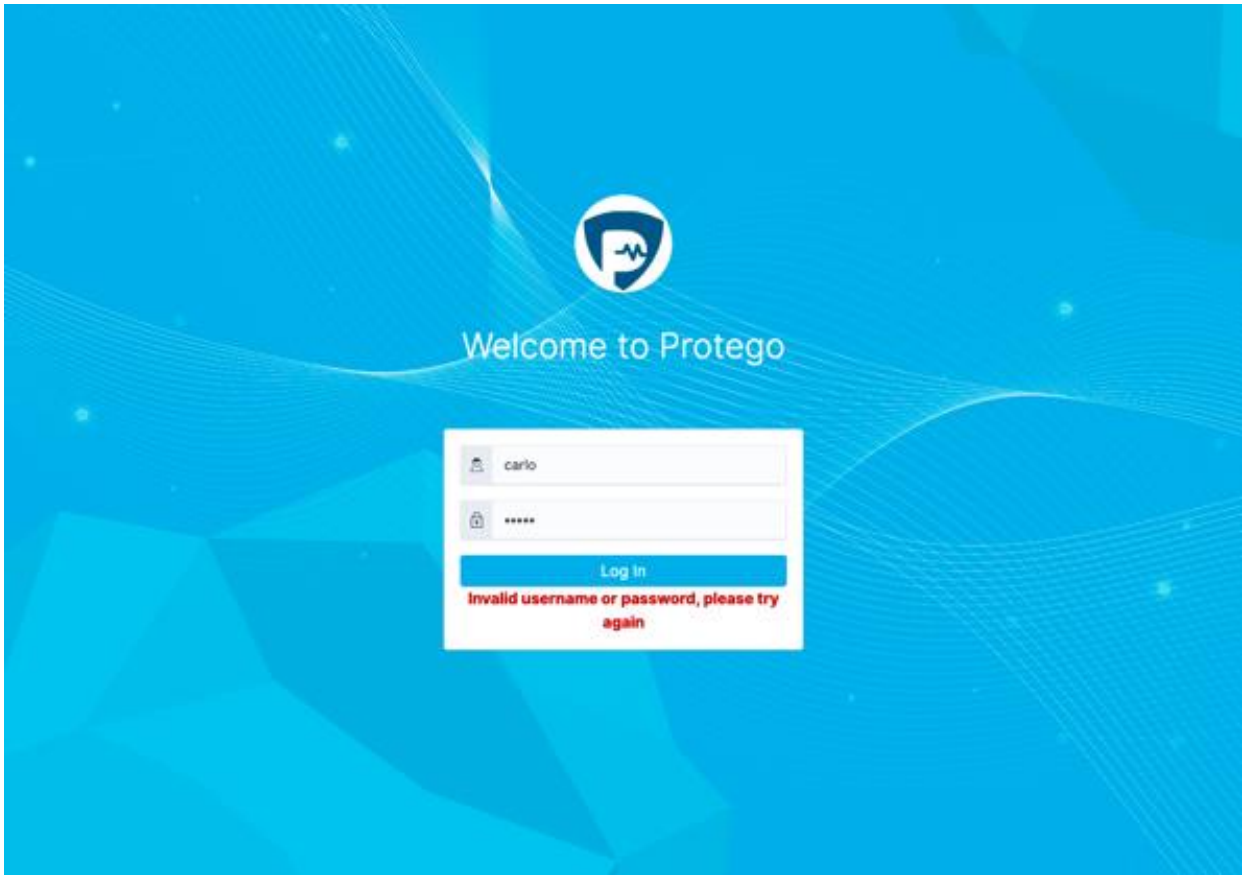
ID: S01.2

Name: Insufficient authorization

Priority: Low

Evaluation: ✓ Passed

Description: The testers verified that each component prevents unauthorized users to perform this task.



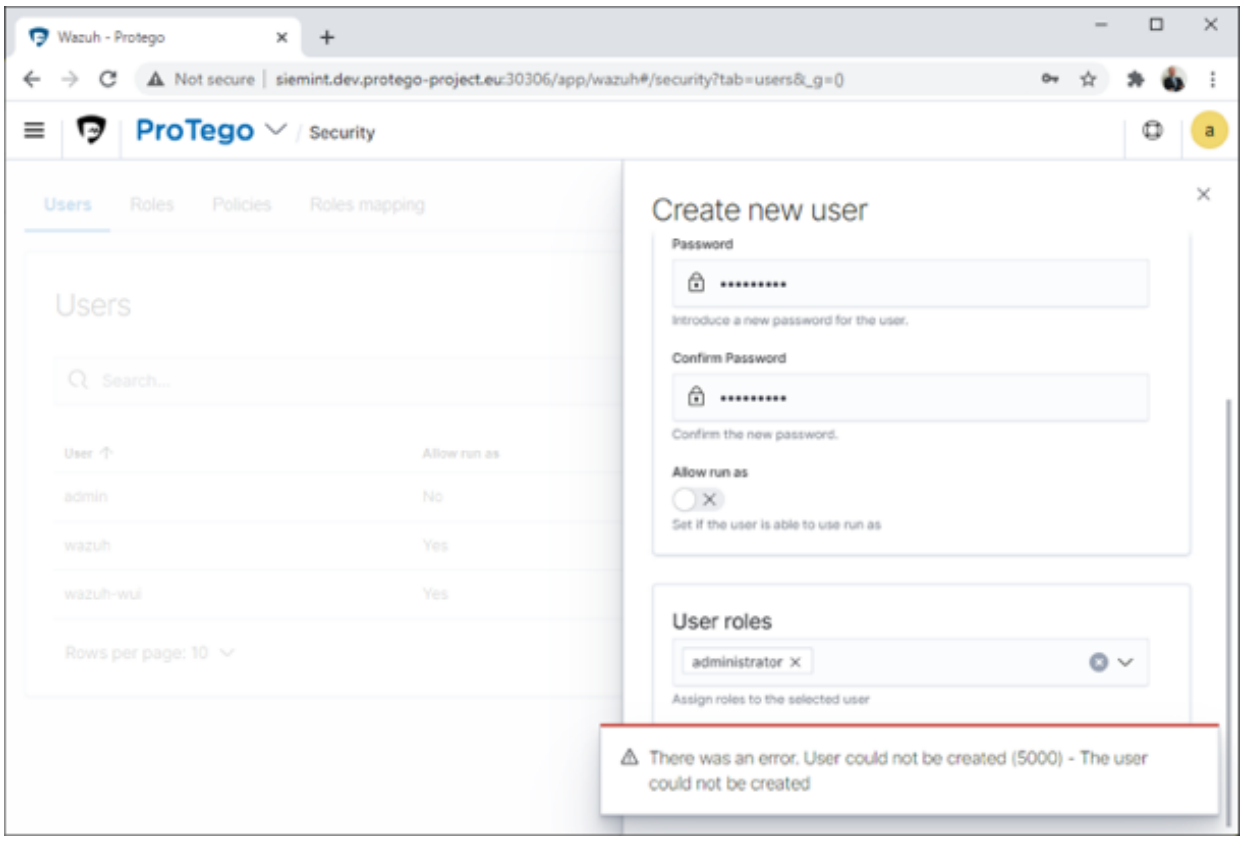
ID: S01.3

Name: User identifier already taken

Priority: Low

Evaluation: ✓ Passed

Description: The testers verified that each component prevents registration of users with same ID.



AT02: Conduct first-time risk assessment

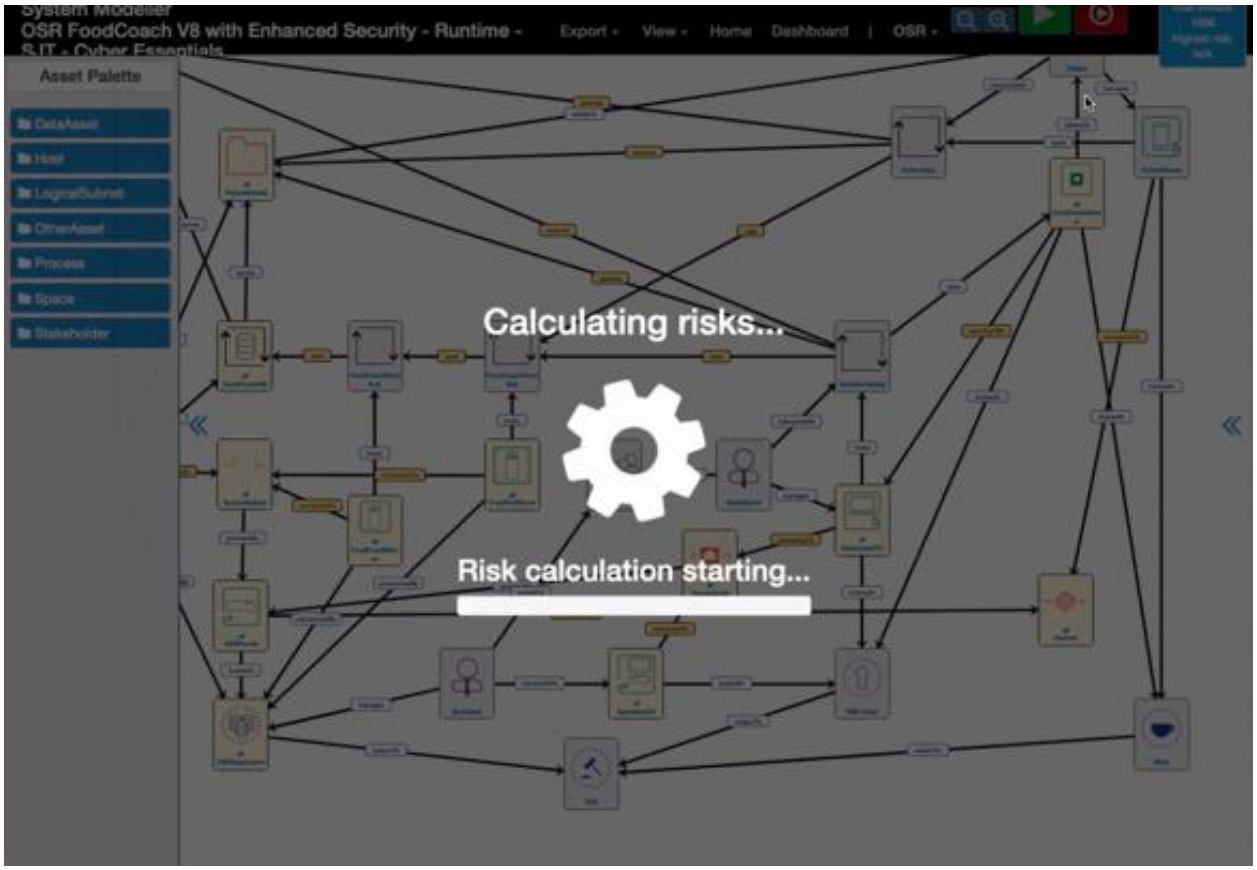
ID: S02.1

Name: Security operator assesses risks

Priority: High

Evaluation: ✓ Passed

Description: The testers logged in SSM tool, modeled OSR infrastructure including FoodCoach application and calculated the associated risks.



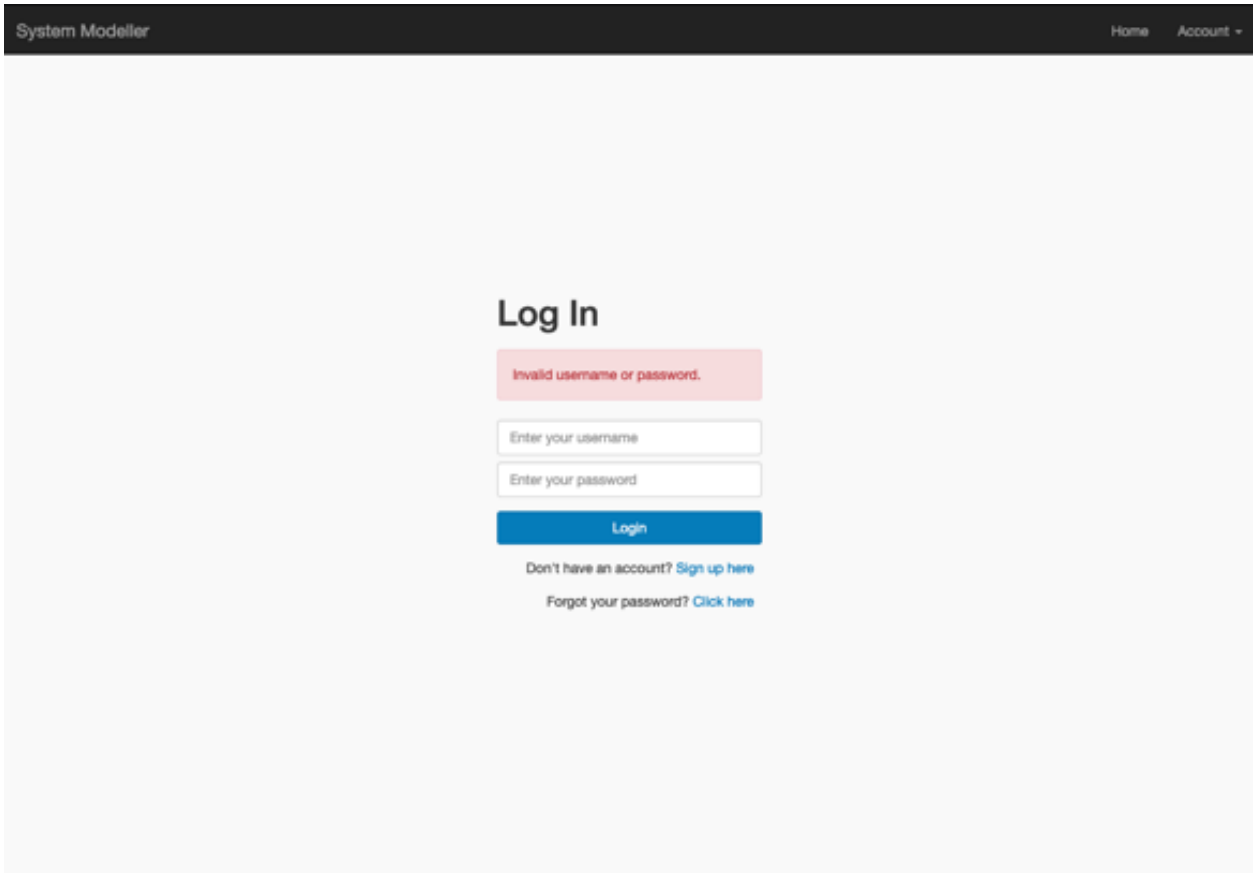
ID: S02.2

Name: Insufficient authorization

Priority: High

Evaluation: ✓ Passed

Description: The testers tried to log in SSM tool with credential not associated with Security operator and the access failed.



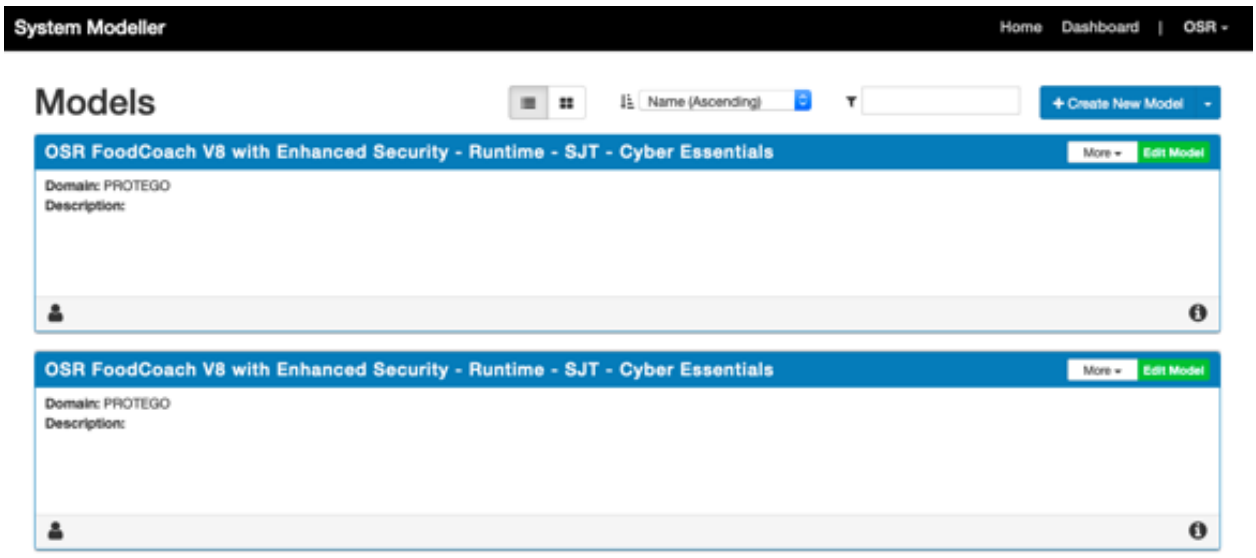
ID: S02.3

Name: First-time assessment already conducted

Priority: Medium

Evaluation: ✕ Failed

Description: The testers verified that SSM tool doesn't prevented the user from creating multiple models with the same name. For future development is suggested to improve this issue.



AT03: Assess prospective risks to the infrastructure

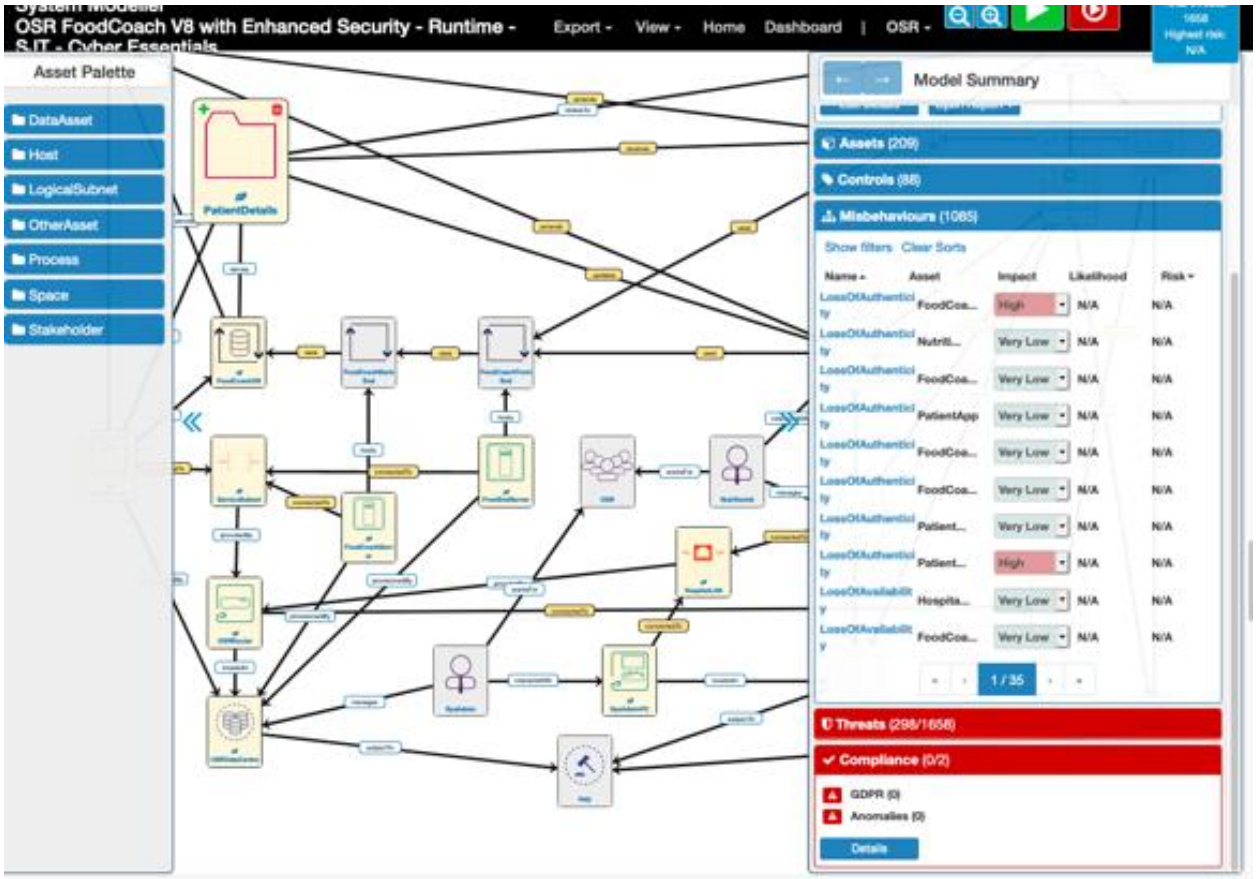
ID: S03.1

Name: Security operator assesses application risks

Priority: High

Evaluation: ✓ Passed

Description: The testers select the application asset modelled in the infrastructure, the SSM displays the associated risks.



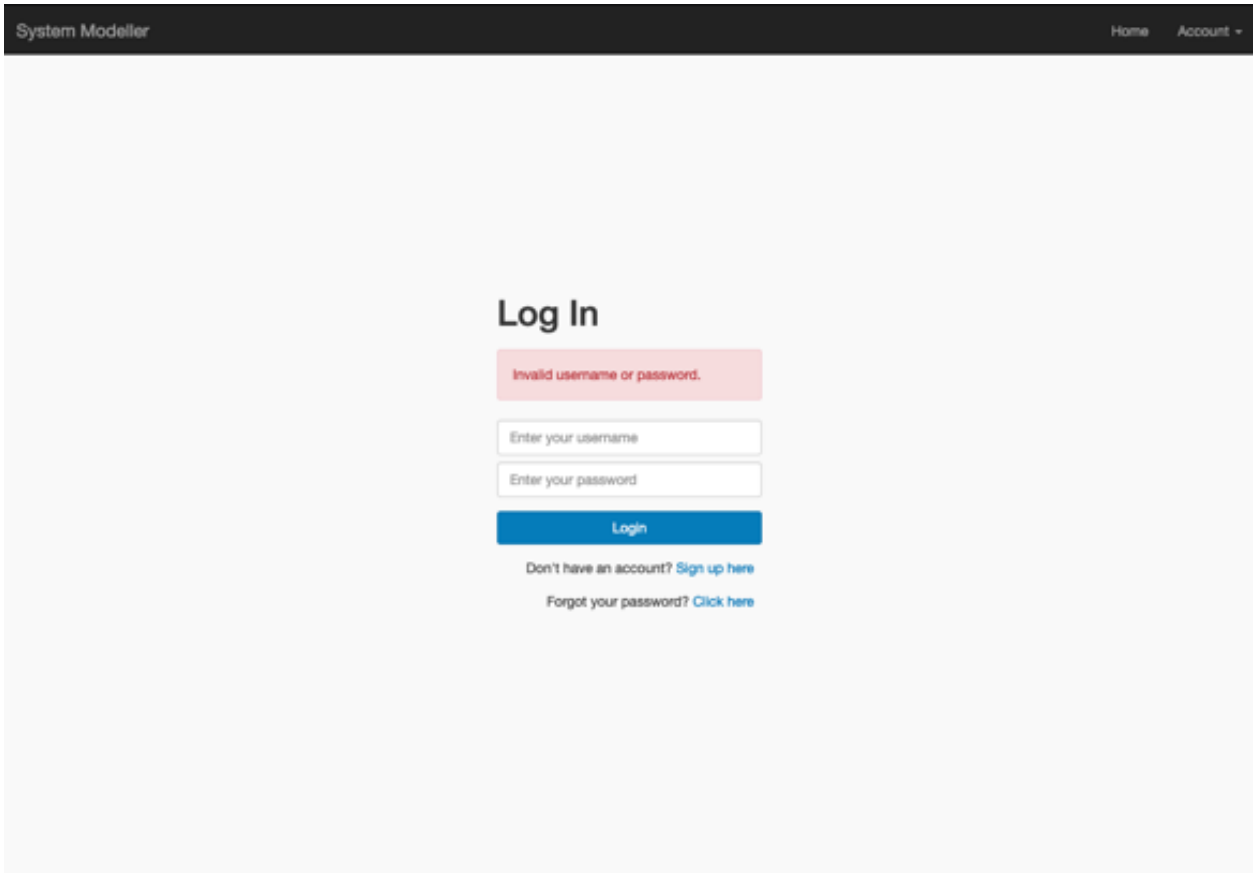
ID: S03.2

Name: Insufficient authorization

Priority: High

Evaluation: ✓ Passed

Description: The testers tried to log in SSM tool with credential not associated with Security operator and the operation failed.



ID: S03.3

Name: First time risk assessment is still to be conducted

Priority: Medium

Evaluation: ✓ Passed

Description: the testers are made aware that the risk assessment was not already performed, and no models are available.



AT04: Register application

ID: S04.1
Name: System operator registers a new application
Priority: Outdated
Evaluation: -
Description: This scenario is outdated because in order to achieve the research objectives, there is not the need of a centralized system in ProTego registering new applications. However, this functionality would be desirable in a more mature state of development.

ID: S04.2
Name: Insufficient authorization
Priority: Outdated
Evaluation: -

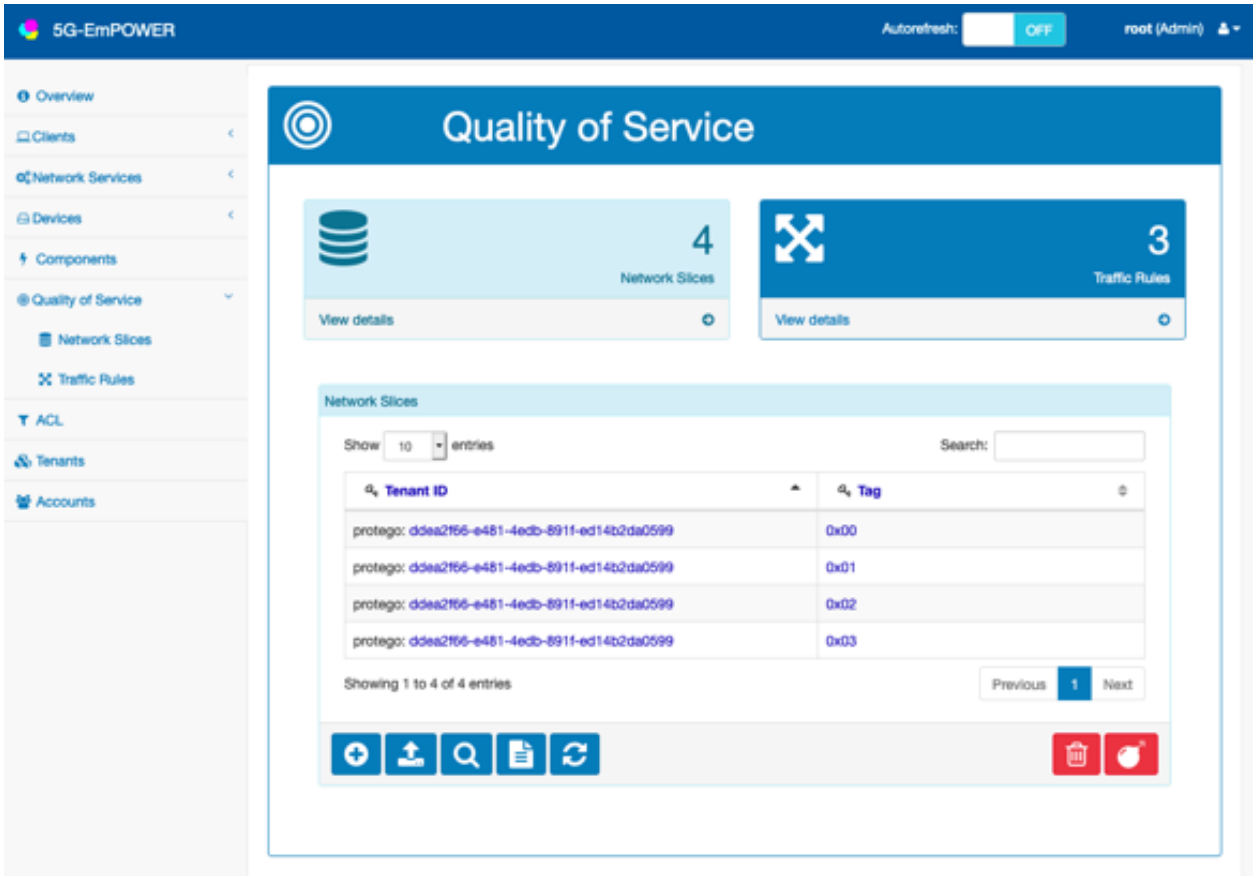
Description: Please refer to the description above.

ID: S04.3
Name: Already existing application
Priority: Outdated
Evaluation: -

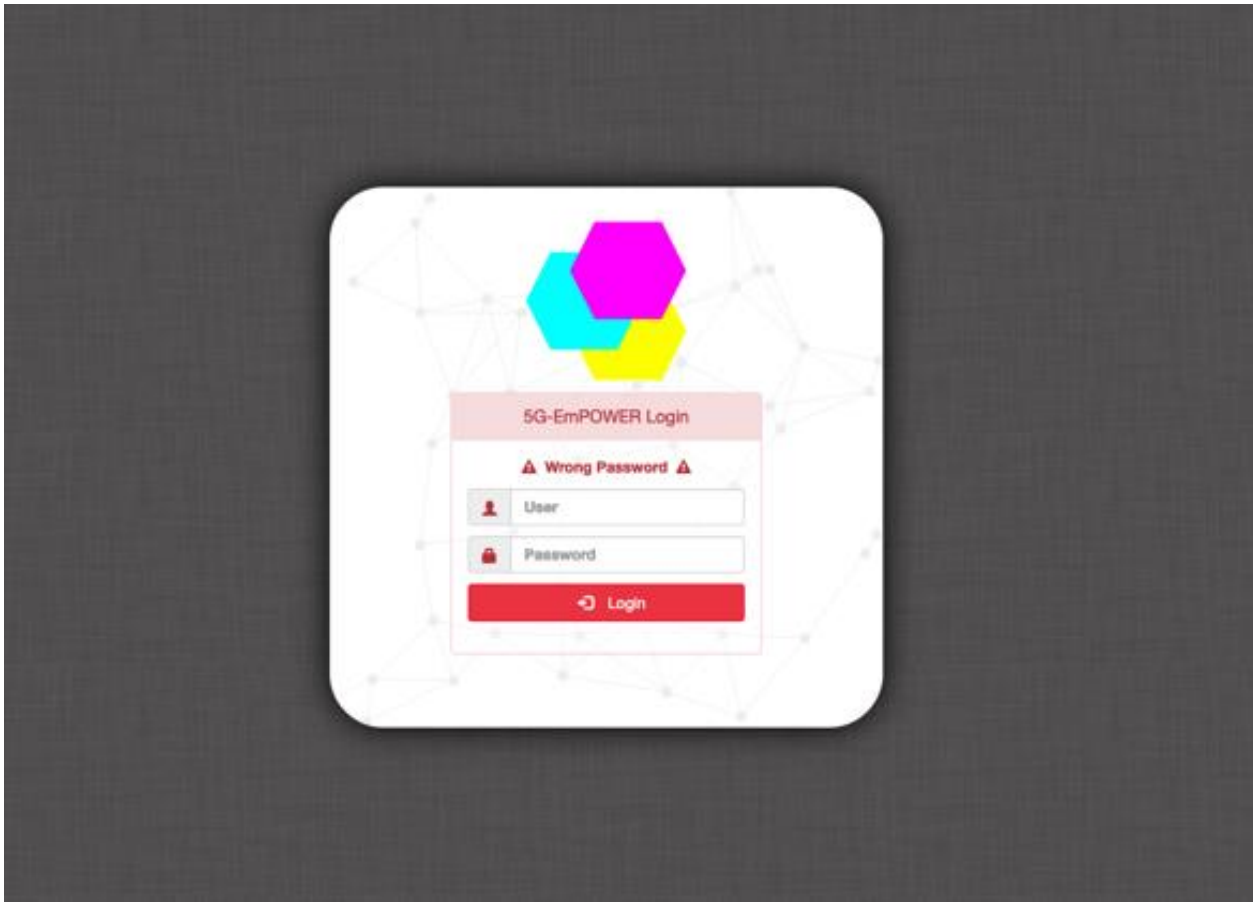
Description: Please refer to the description above.

AT05: Configure application network slices

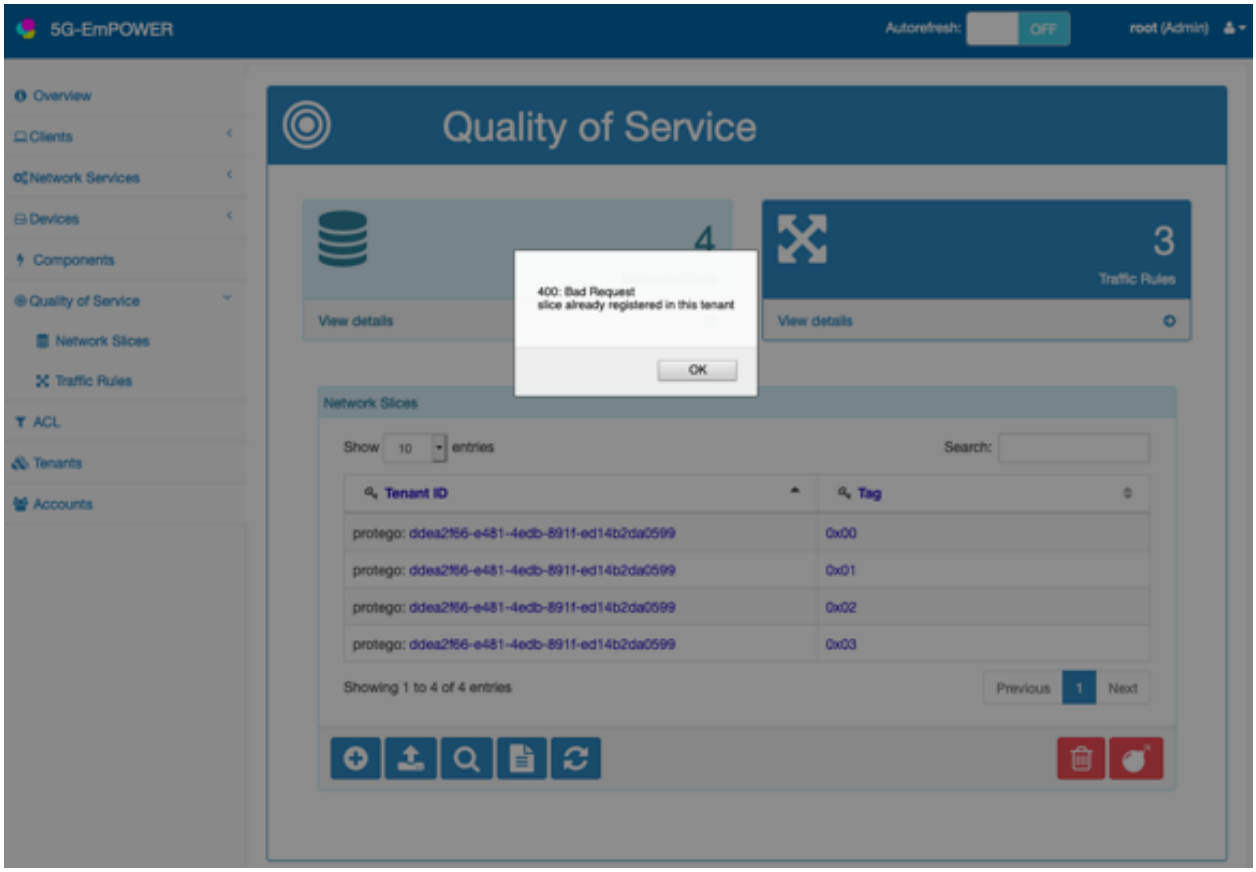
ID: S05.1
Name: Security operator configures network slices
Priority: High
Evaluation: ✓ Passed
Description: The tester logged in Network slicing controller, he created the tenant and the slices, and for each slice, he configured the application address.



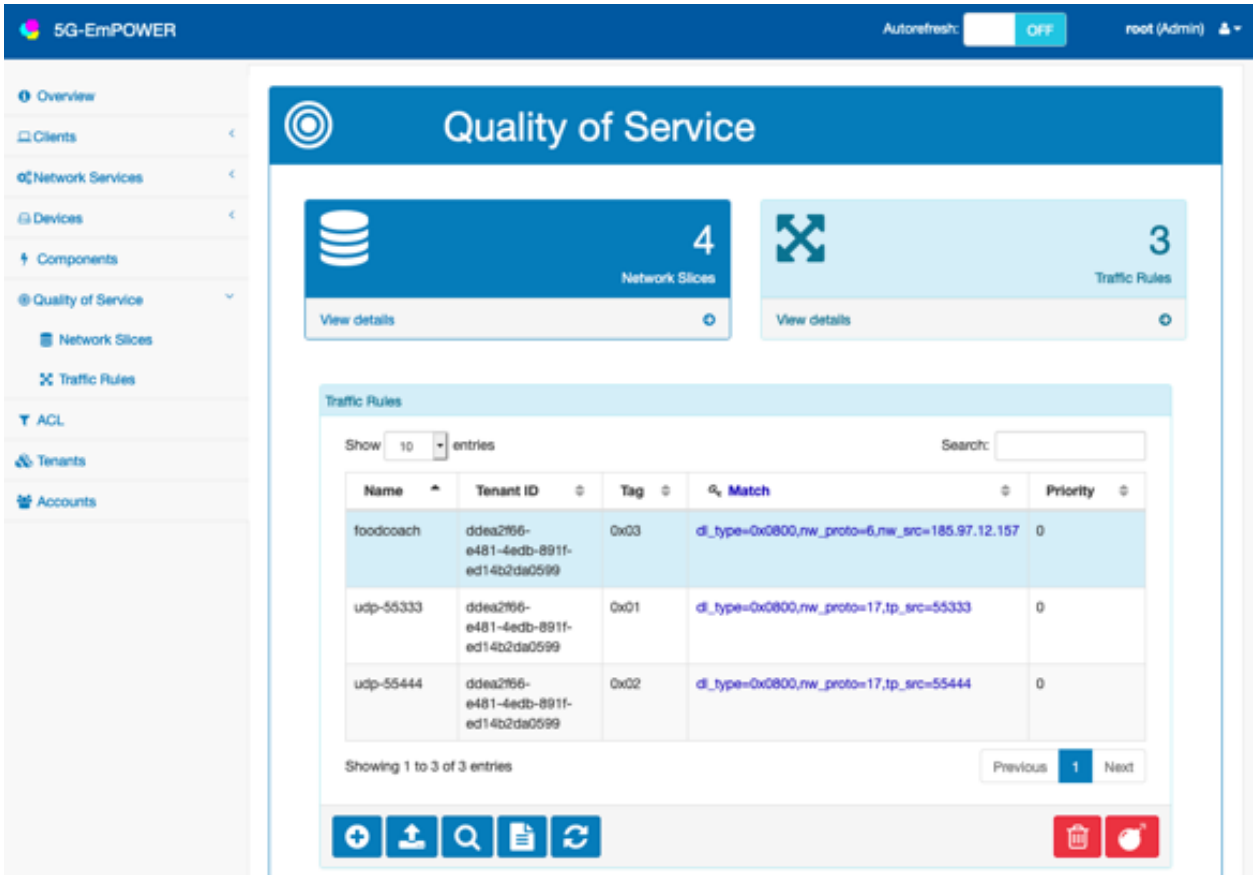
ID: S05.2
Name: Insufficient authorization
Priority: High
Evaluation: ✓ Passed
Description: The testers tried to log in SSM tool with credential not associated with Network operator and the operation failed.



ID: S05.3
Name: Network slices are already configured
Priority: Medium
Evaluation: ✓ Passed
Description: The application network slices were already configured, the testers tried to repeat the operation, the operation failed.



ID: S05.4
Name: Non-existing application
Priority: Medium
Evaluation: ✓ Passed
Description: the testers verified that the Network slicing controller is capable of configuring the slices only after that the application tenant has been registered.



AT06: Configure application logging mechanism

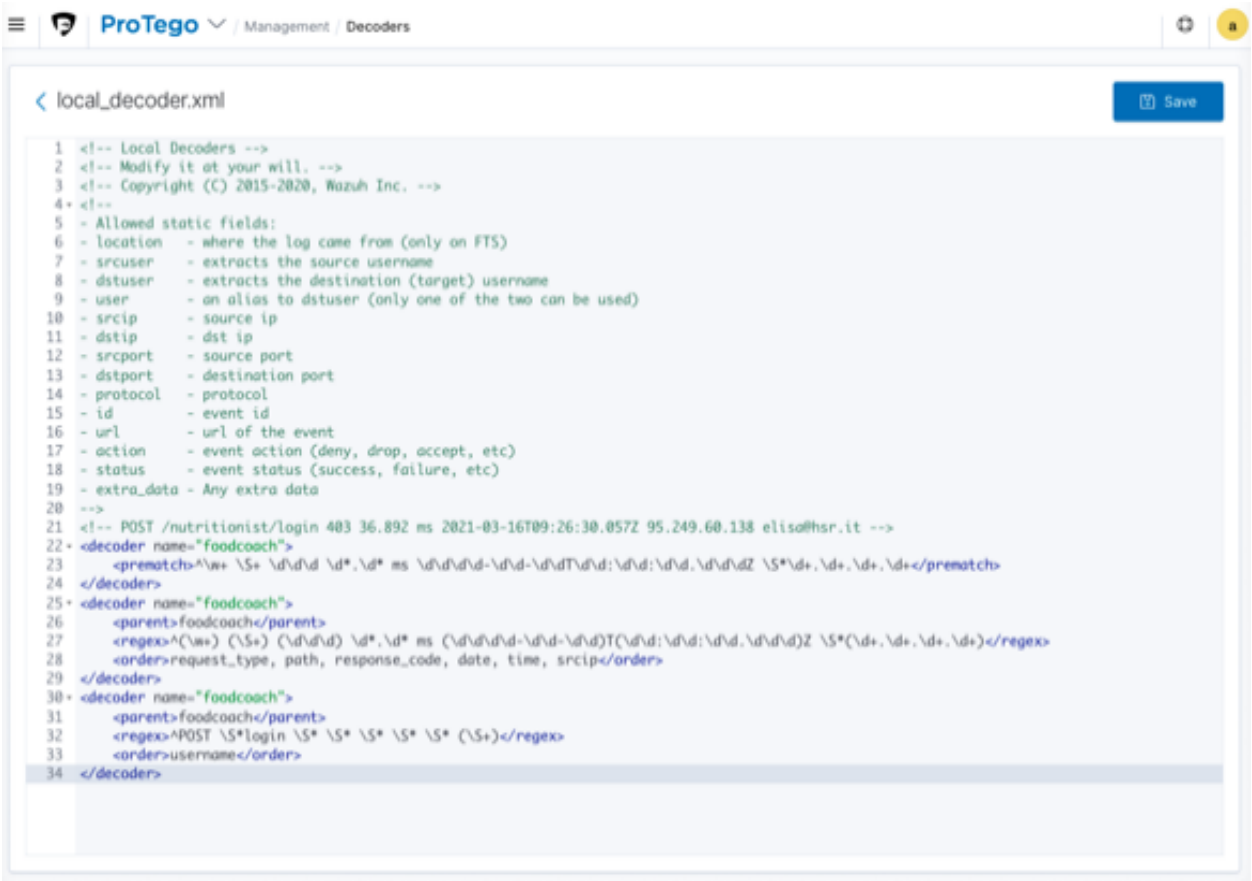
ID: S06.1

Name: Security operator specifies logging mechanism

Priority: High

Evaluation: ✓ Passed

Description: The testers configured logging mechanism to monitor the application.



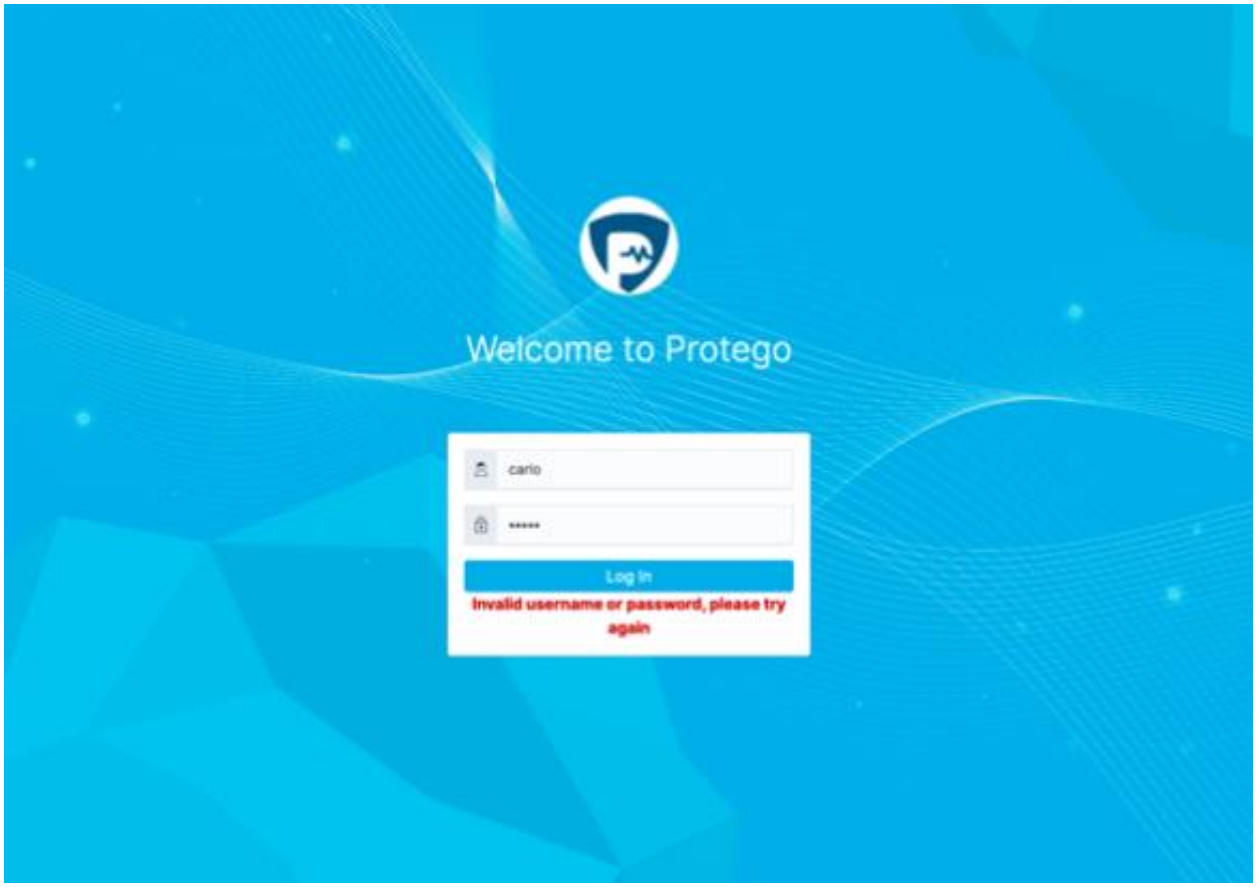
ID: S06.2

Name: Insufficient authorization

Priority: High

Evaluation: ✓ Passed

Description: The testers tried to log in SIEM tool with credential not associated with Network operator and the operation failed.



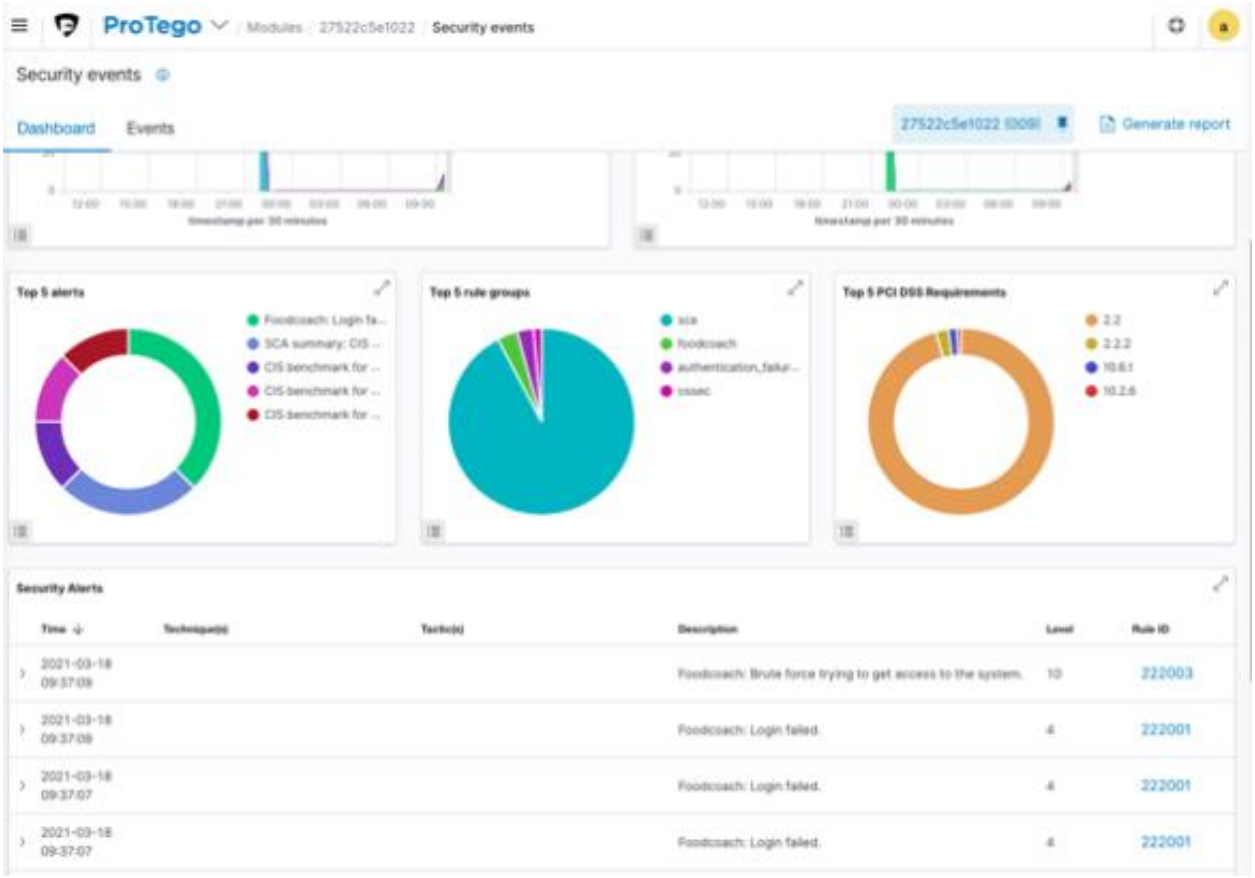
ID: S06.3

Name: Logging mechanism is already configured

Priority: Medium

Evaluation: ✓ Passed

Description: The testers accessed the ProTego dashboard and verified that the logging mechanism had already been configured.



ID: S06.4

Name: Non-existing application

Priority: Outdated

Evaluation: -

Description: This scenario is outdated because there is not the need of a centralized system in ProTego registering applications.

AT07: Specify application access control

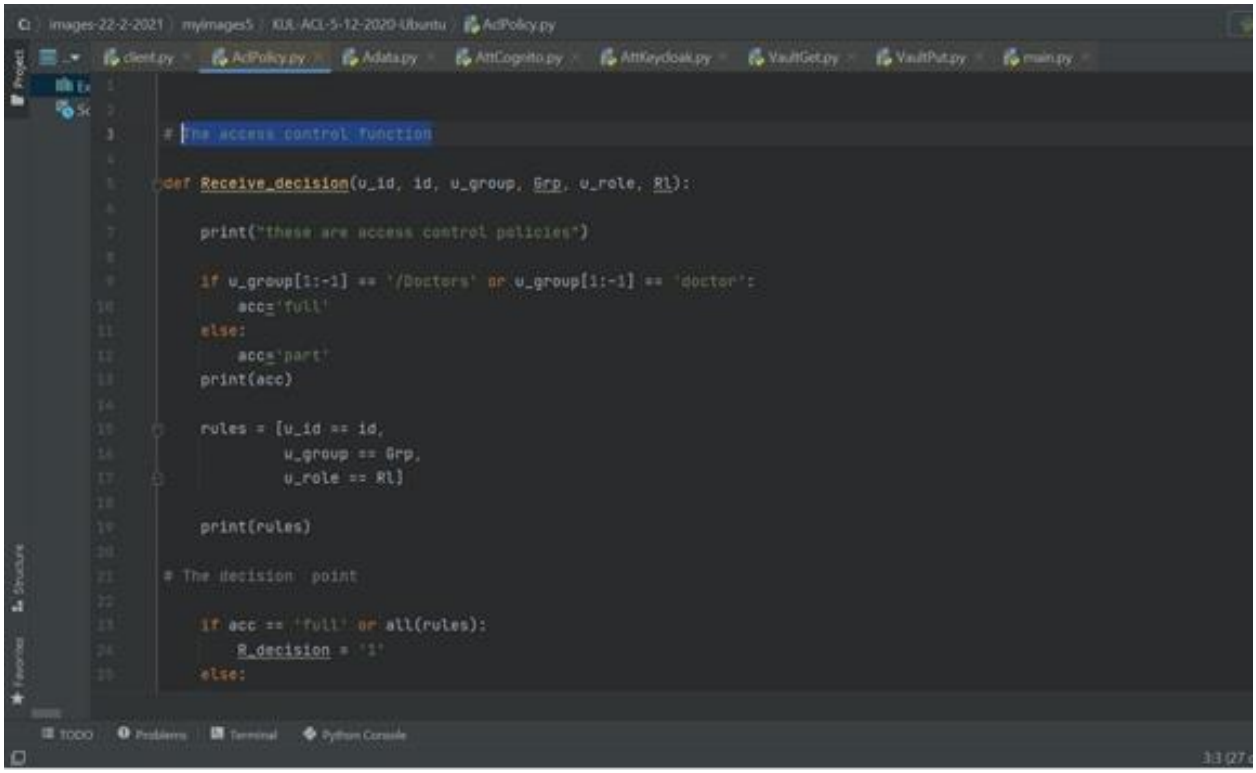
ID: S07.1

Name: Security operator specifies access control

Priority: High

Evaluation: ✓ Passed

Description: There is not a central system in ProTego to specify access control. However, the testers verify the possibility to specify the application access control by manually intervene on the policy. For future development is suggested to include more user-friendly way to configure the access control.



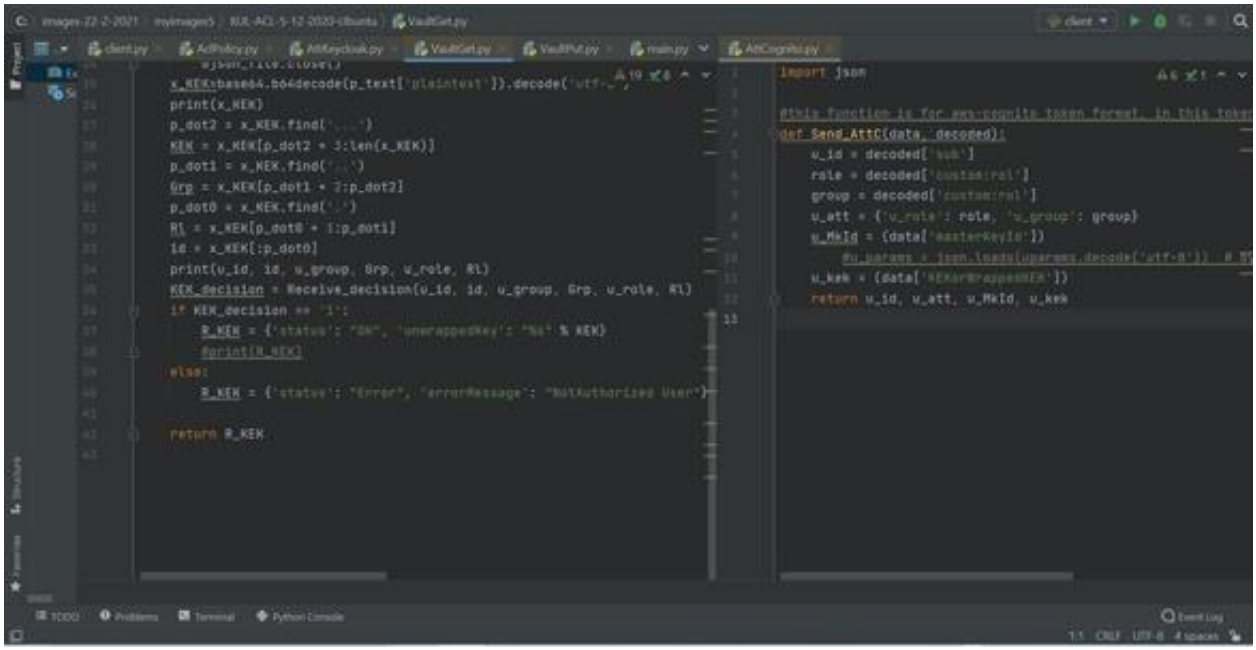
ID: S07.2

Name: Insufficient authorization

Priority: High

Evaluation: ✓ Passed

Description: The testers verify that only authorized user can specify the access control policy (i.e., change or update the security policies in place).



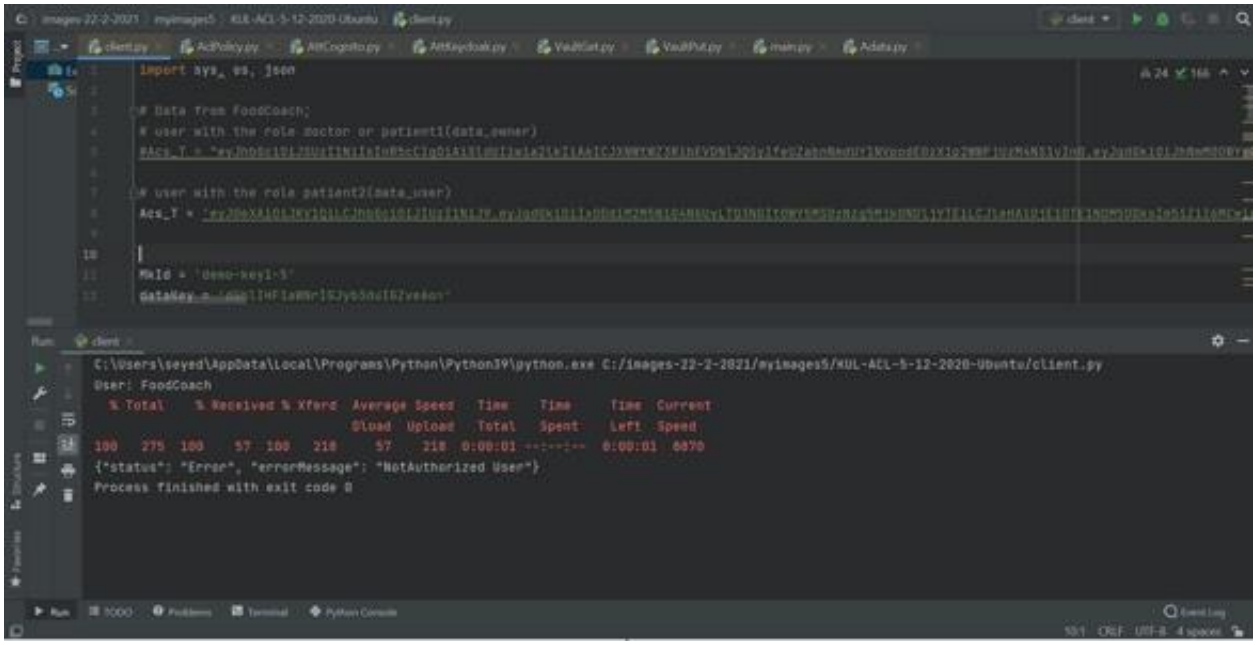
ID: S07.3

Name: Application access control is already active

Priority: Medium

Evaluation: ✓ Passed

Description: The user can manually verify whether access control is already active. For example, this can be tested by trying to access data when presenting a wrong user token.



ID: S07.4

Name: Non-existing application

Priority: Outdated

Evaluation: -

Description: This scenario is outdated because based on the presence of an application-centric access control. This is not the case, as a user-centric access control system is in place in ProTego.

AT08: Specify application query templates

ID: S08.1

Name: Data operators specifies application query templates

Priority: Outdated

Evaluation: -

Description: This scenario is outdated since there is not the need of a centralized system specifying application query templates in order to achieve ProTego research objectives. However, this functionality could be desirable in a more mature state of development.

ID: S08.2

Name: Insufficient authorization

Priority: Outdated

Evaluation: -

Description: Please refer to the description above.

ID: S08.3

Name: Application already has associated query templates

Priority: Outdated

Evaluation: -

Description: Please refer to the description above.

ID: S08.4

Name: Non-existing application

Priority: Outdated

Evaluation: -

Description: Please refer to the description above.

AT09: Store initial medical data

ID: S09.1

Name: Data operator stores initial medical data

Priority: Low

Evaluation: ✕ Failed

Description: There is not a centralized system in ProTego storing initial medical data. However, through the application it is possible to store initial medical data. A centralized system to perform this operation would be desirable in a more mature state of development.

ID: S09.2

Name: Insufficient authorization

Priority: Low

Evaluation: ✕ Failed

Description: Please refear to the description above.

ID: S09.3

Name: Application initial resources are already stored

Priority: Low

Evaluation: ✕ Failed

Description: Please refear to the description above.

ID: S09.4

Name: Non-existing application

Priority: Outdated

Evaluation: -

Description: This scenario depends on the centralized system.

AT10: Register IoT device

ID: S10.1

Name: Network operator registers an IoT device

Priority: Outdated

Evaluation: -

Description: In the course of the development of the IoT solution, the current functionality has become outdated and it will be replaced with a mechanism that will introduce into the IoT a refresh token associated with the patient, which makes unnecessary to register the IoT itself.

ID: S10.2

Name: Insufficient authorization

Priority: Outdated

Evaluation: -

Description: Refer to description above.

ID: S10.3

Name: IoT device already registered

Priority: Outdated

Evaluation: -

Description: Refer to description above.

ID: S10.4

Name: Invalid identity certificate

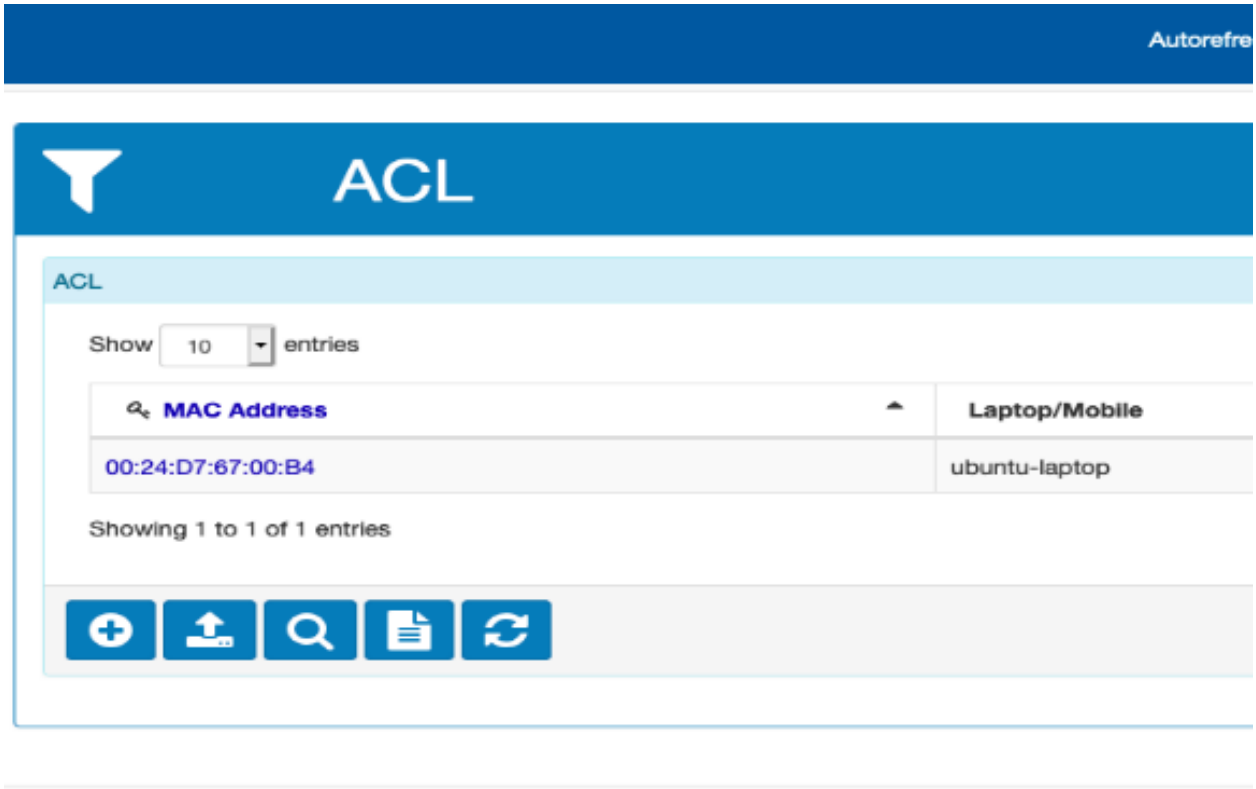
Priority: Outdated

Evaluation: -

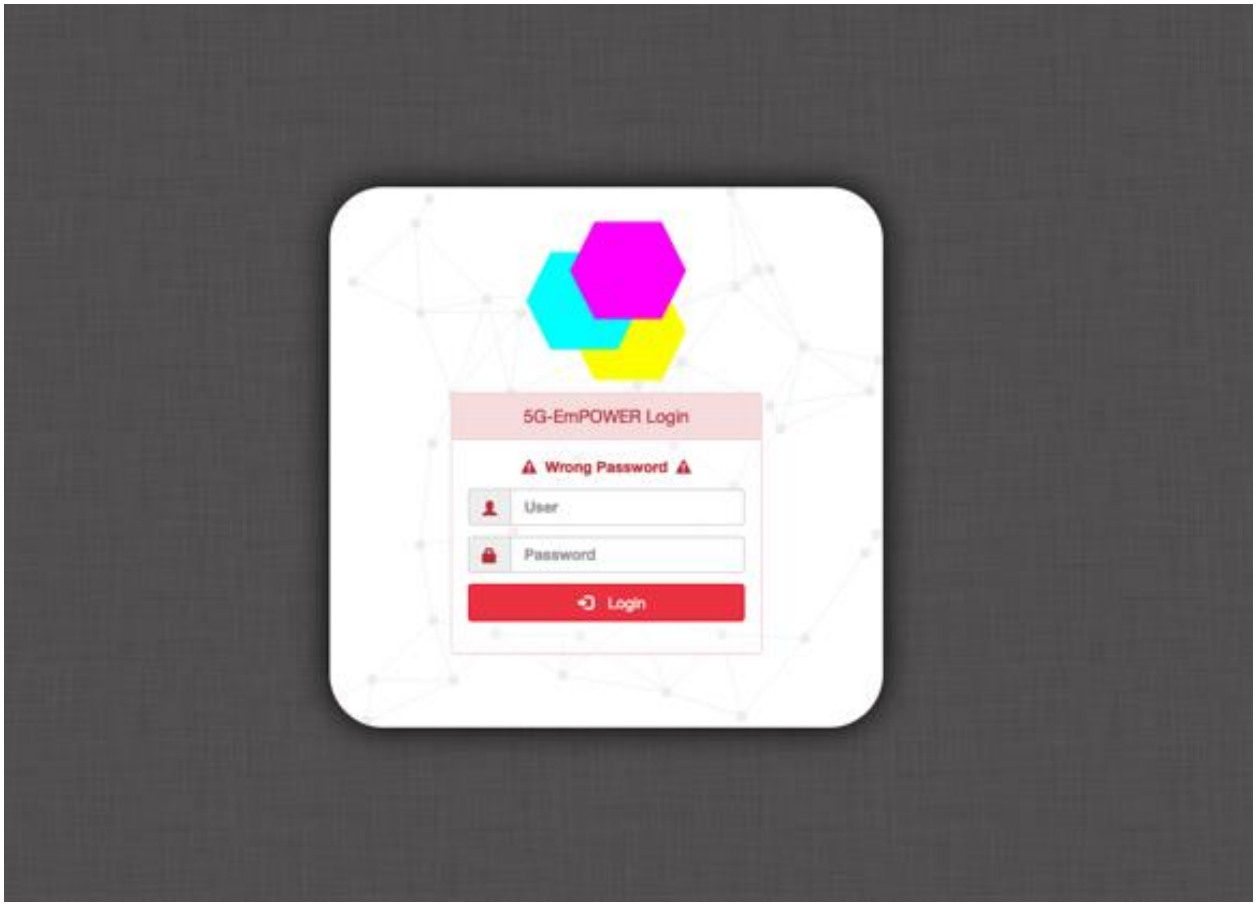
Description: Refer to description above.

AT11: Register mobile device

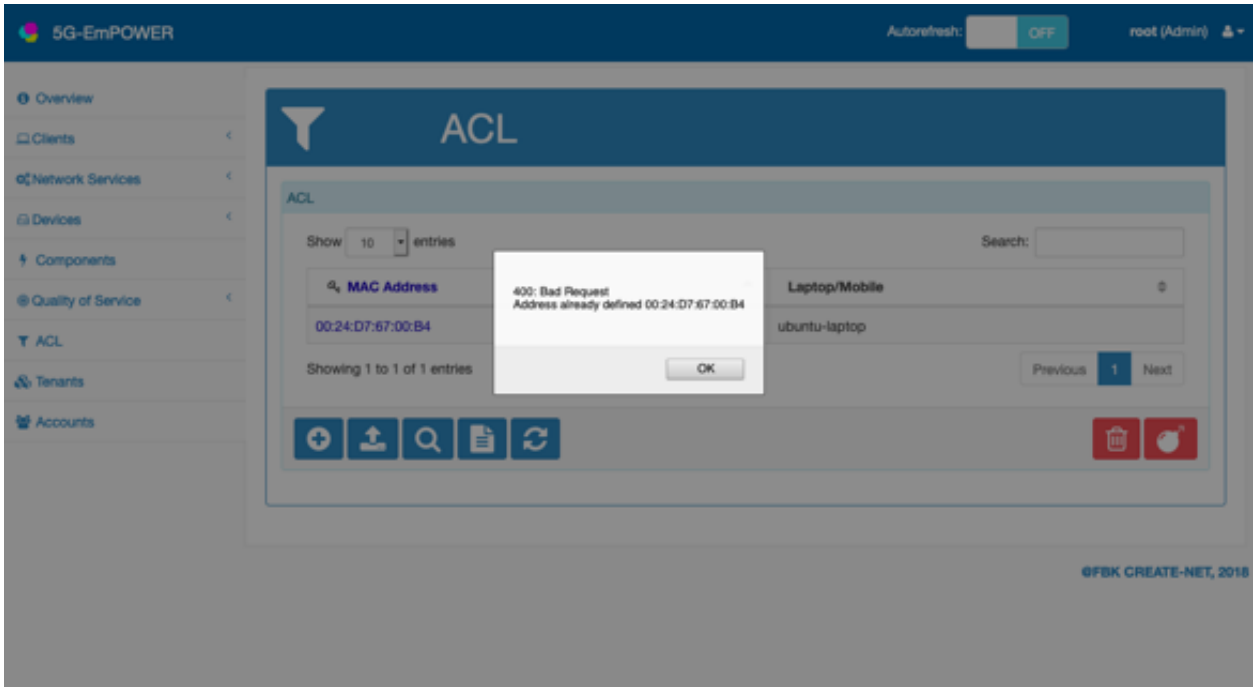
ID: S11.1
Name: Network operator registers a new mobile device
Priority: High
Evaluation: ✓ Passed
Description: The tester logged in Network Slicing controller and register the device providing Name and MAC Address.



ID: S11.2
Name: Insufficient authorization
Priority: High
Evaluation: ✓ Passed
Description: The testers tried to log in network slicing controller with credential not associated with Network operator and the operation failed.



ID: S11.3
Name: Mobile device is already registered
Priority: Medium
Evaluation: ✓ Passed
Description: The mobile device is already configured, the testers tried to repeat the operation, the operation failed.



ID: S11.4
Name: Network slices are not configured
Priority: Outdated
Evaluation: -
Description: Given that the control is not centralized, the current scenario is not relevant.

ID: S11.5
Name: Non existing application
Priority: Outdated
Evaluation: -
Description: Given that the control is not centralized, the current scenario is not relevant.

AT12: Store medical data

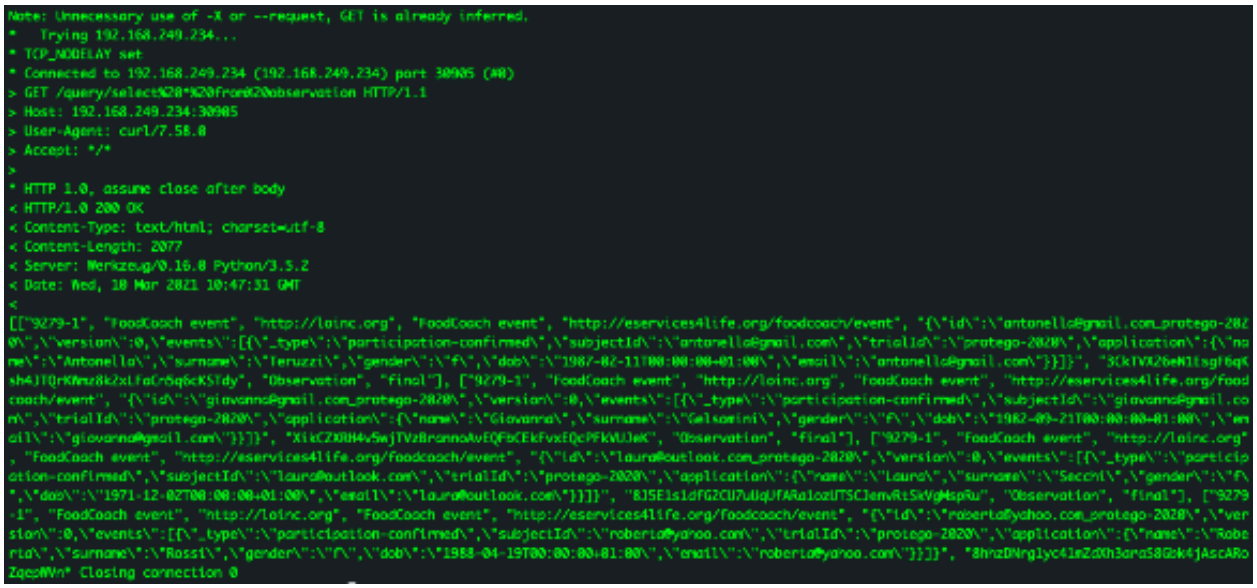
ID: S12.1

Name: Nutritionist stores patient's participations

Priority: High

Evaluation: ✓ Passed

Description: The testers, acting as the Nutritionist, got the authorization token and stored the patient's records in ProTego.



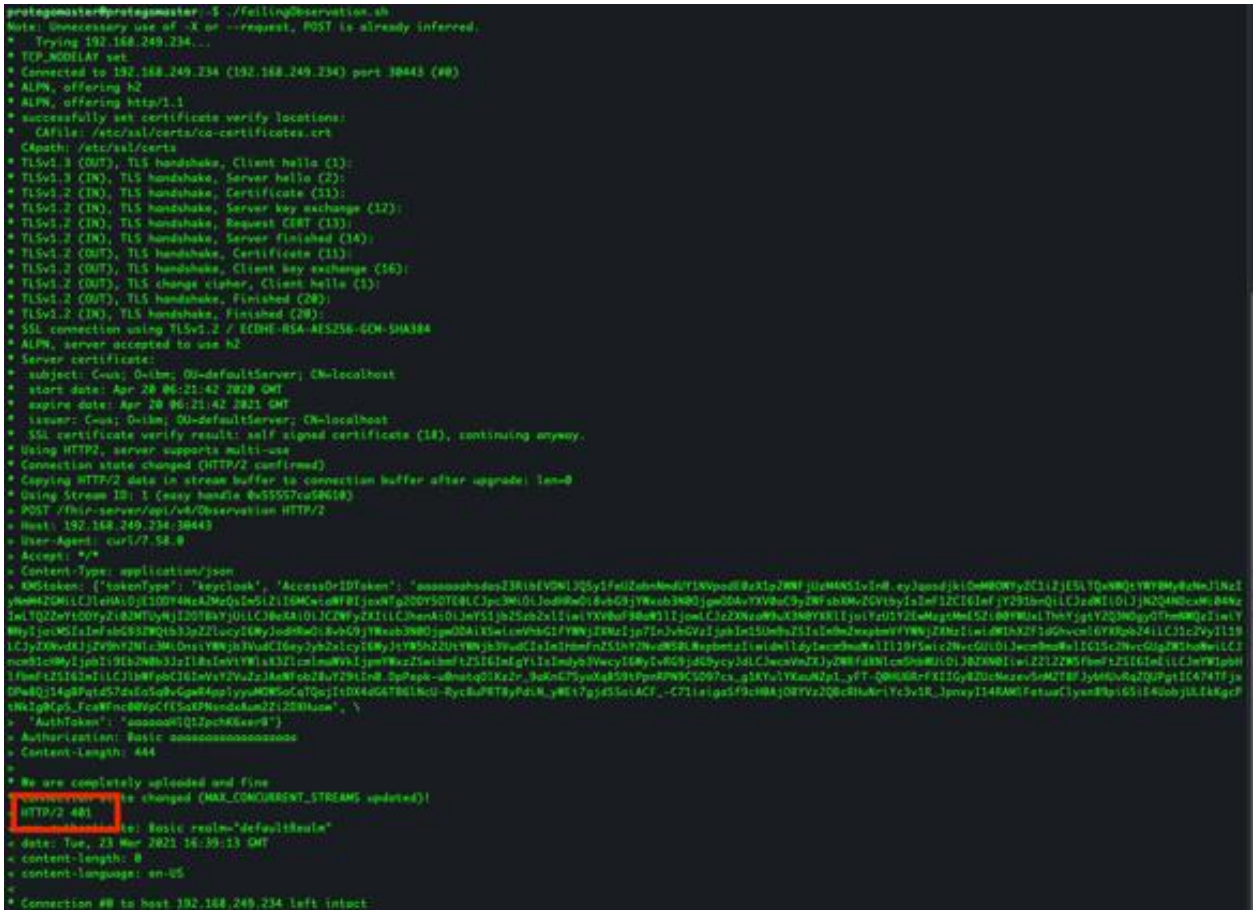
ID: S12.2

Name: Insufficient authorization

Priority: High

Evaluation: ✓ Passed

Description: The tester, acting as an external user, tried to store the patient's records in ProTego and the request was rejected.



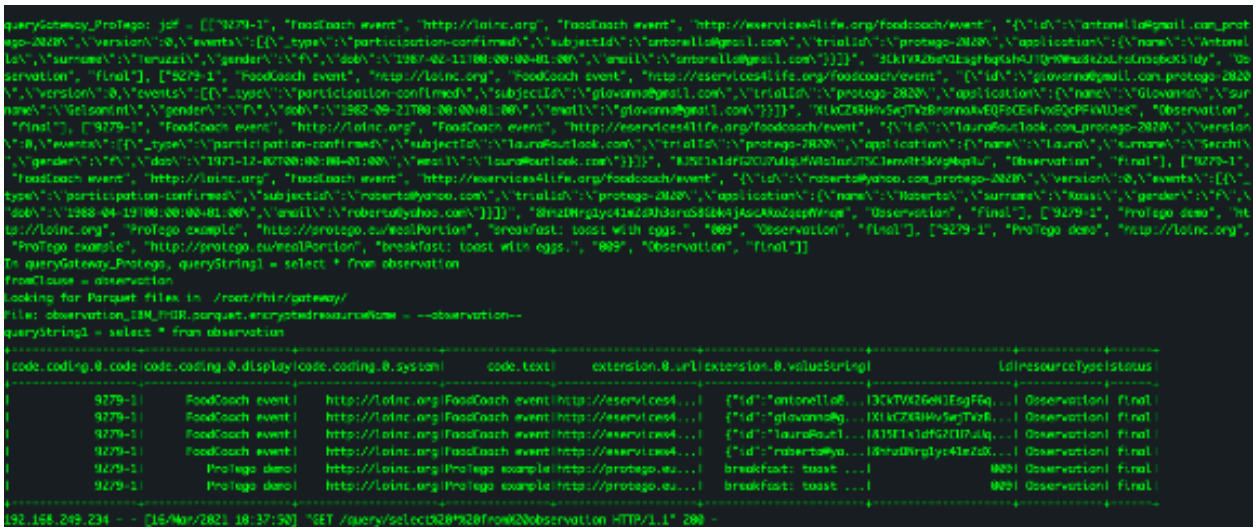
ID: S12.3

Name: Resource with the same id already exists

Priority: Outdated

Evaluation: -

Description: FHIR standard allows this operation.



ID: S12.4

Name: Authorization token is expired

Priority: Medium

Evaluation: ✕ Failed

Description: The testers, acting as the nutritionist, logged in the application. After a certain amount of time the token expired. When the testers tried to record a new participation, the request was not rejected.

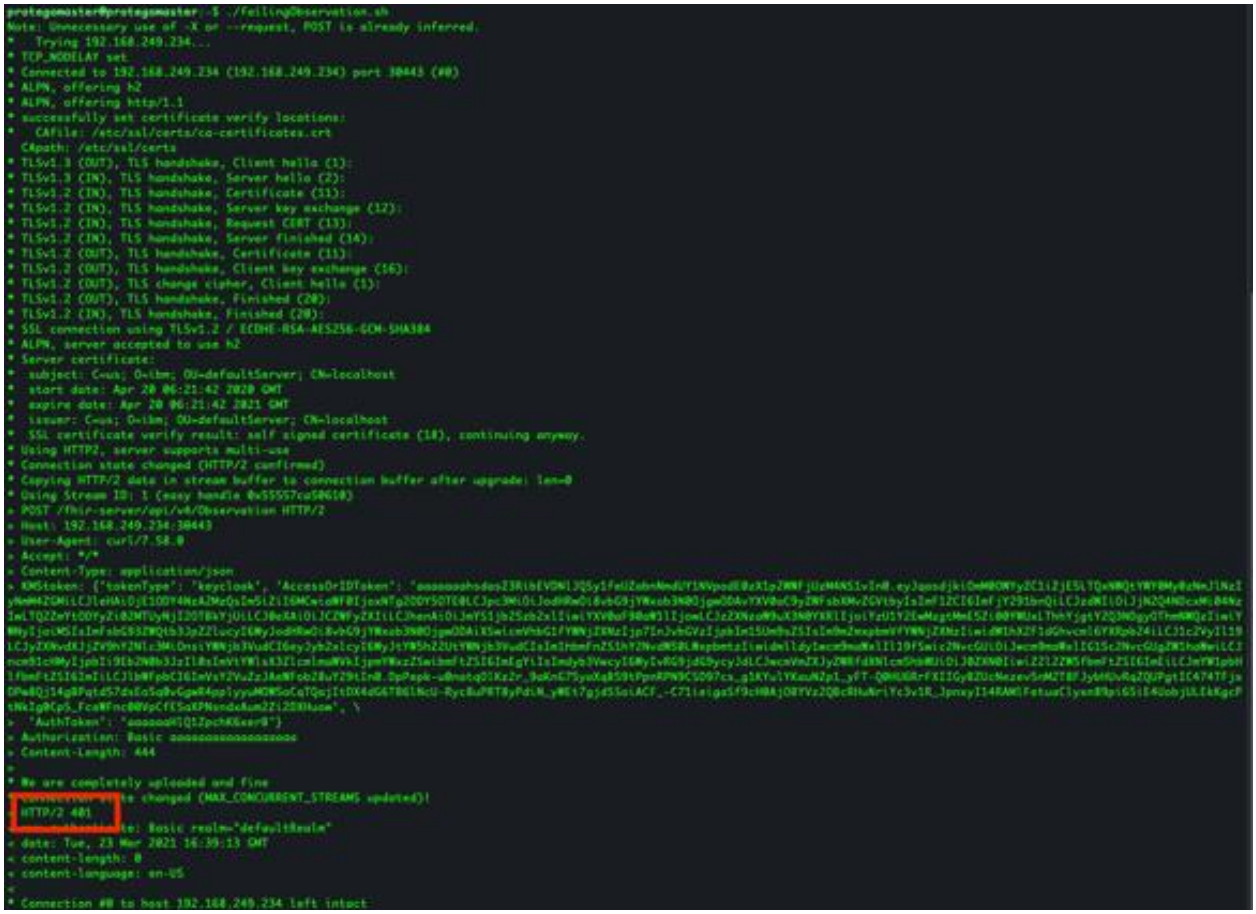
ID: S12.5

Name: Authorization token is ill-formed

Priority: Medium

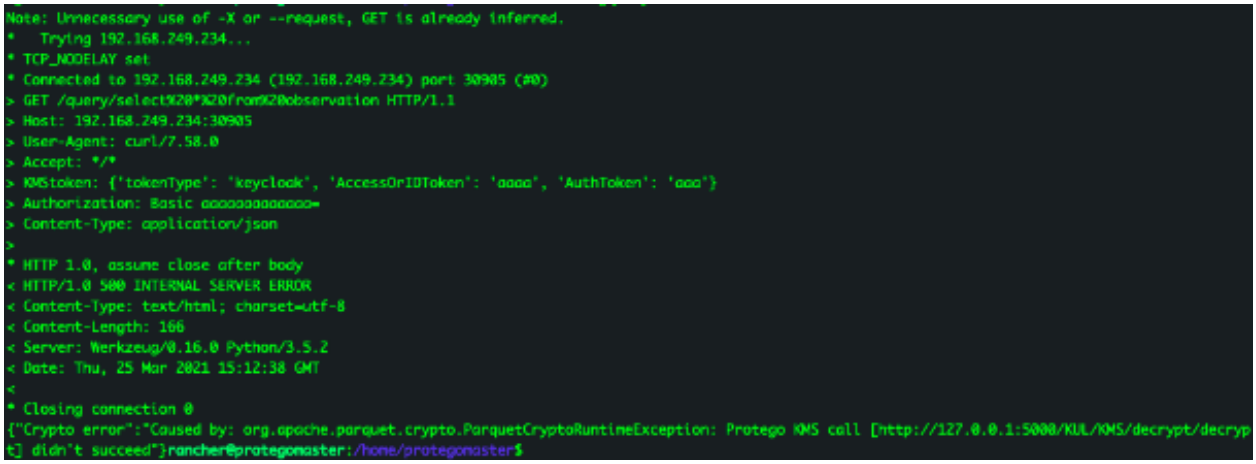
Evaluation: ✓ Passed

Description: The testers, impersonating the nutritionist, forged an ill-formed token. When they tried to store a new record, the request was rejected.

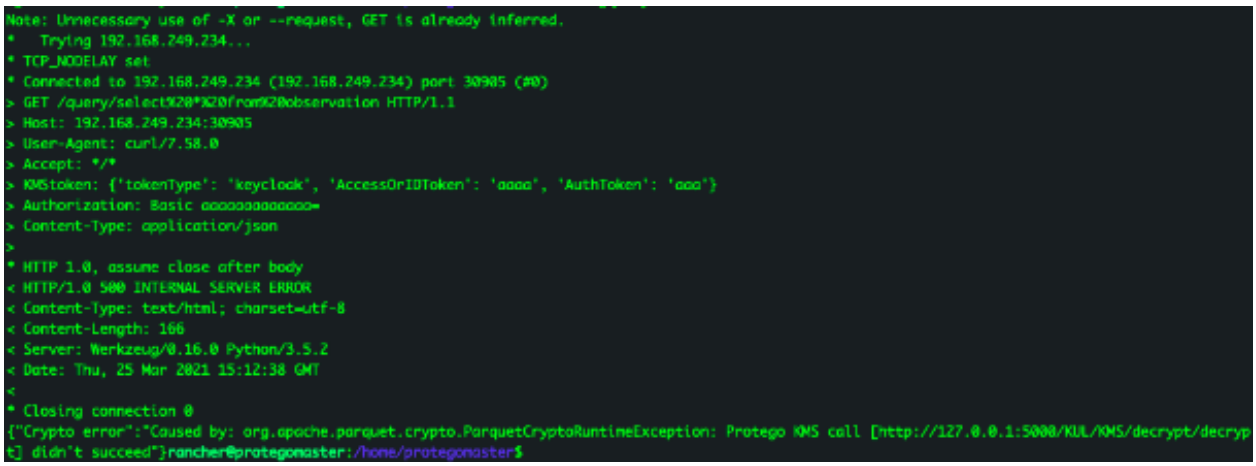


AT13: Retrieve medical data

ID: S13.1
Name: Nutritionist queries for some patient's participations
Priority: High
Evaluation: ✓ Passed
Description: The testers, acting as the Nutritionist, got the authorization token and retrieved the patient's records in ProTego.



ID: S13.2
Name: Insufficient authorization
Priority: High
Evaluation: ✓ Passed
Description: The testers, acting as an external user, created a token for an unauthorized user. Requesting the data with this token resulted in the request being rejected.

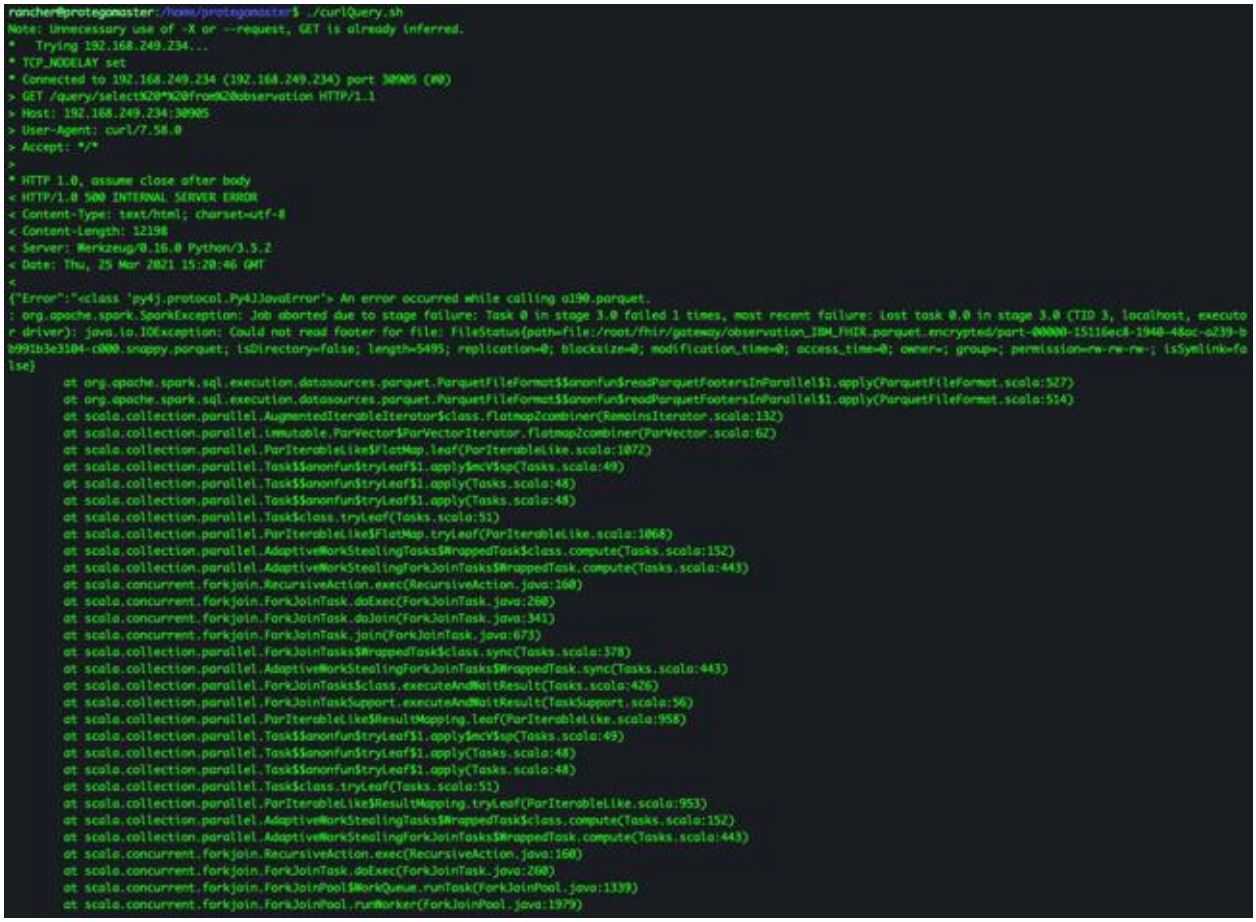


ID: S13.3
Name: Medical data have been tampered with **Priority:** High
Evaluation: ✗ Failed
Description: This scenario could not be tested due to technical issues.

ID: S13.4
Name: Non existing query
Priority: Outdated
Evaluation: -
Description: This functionality is outdated since there is not the need of a centralized system specifying application query templates in order to achieve ProTego research objectives.

ID: S13.5
Name: Authorization token is expired
Priority: Medium
Evaluation: ✗ Failed
Description: This scenario has not been implemented yet.

ID: S13.6
Name: Authorization token is ill-formed
Priority: Medium
Evaluation: ✓ Passed
Description: The testers, acting as an external user, created an ill-formed token. Requesting the data with this token resulted in the request being rejected.



AT14: Assign IoT device to application user

ID: S14.1
Name: Assign a new IoT device
Priority: High
Evaluation: ✗ Failed
Description: The current functionality has not been implemented yet. However, in the next release, this functionality is expected to be implemented by using a refresh token associated to the IoT user that must be registered in the IoT device before its first use, and that will be a required element to allow the device interact with the ProTego toolkit.

ID: S14.2
Name: Assign a device to another's nutritionist patient
Priority: High
Evaluation: ✗ Failed
Description: Refer to description above.

ID: S14.3
Name: Admin tries to assign a device
Priority: Medium
Evaluation: ✗ Failed
Description: Refer to description above.

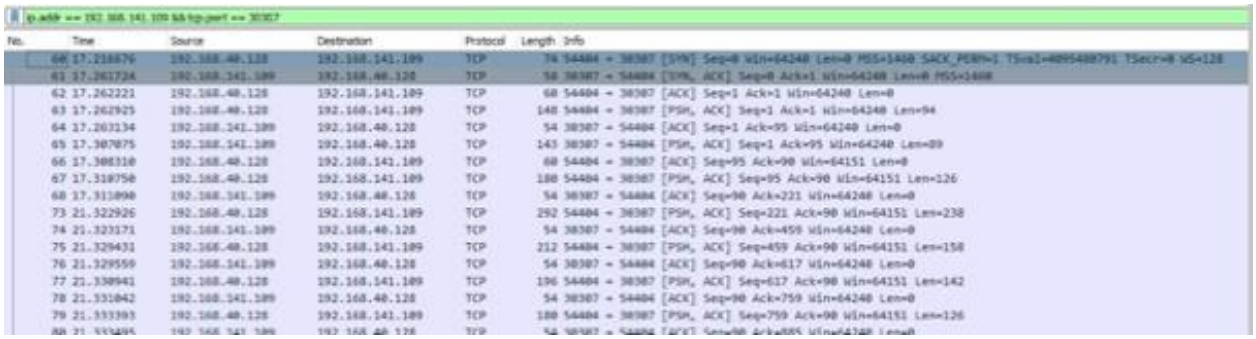
ID: S14.4
Name: IoT device is already assigned
Priority: Medium
Evaluation: ✗ Failed
Description: Refer to description above.

ID: S14.5
Name: Authorization token is expired
Priority: Medium
Evaluation: ✗ Failed
Description: Refer to description above.

ID: S14.6
Name: Authorization token is ill formed
Priority: Medium
Evaluation: ✗ Failed
Description: Refer to description above.

AT15: Log custom application event

ID: S15.1
Name: Correct application key
Priority: High
Evaluation: ✓ Passed
Description: The testers, verify that the application logs the custom application events. The agent that collects the logs is registered using the TLS protocol to obtain an encryption key that is later used to ship the logs to the component analyzes them.



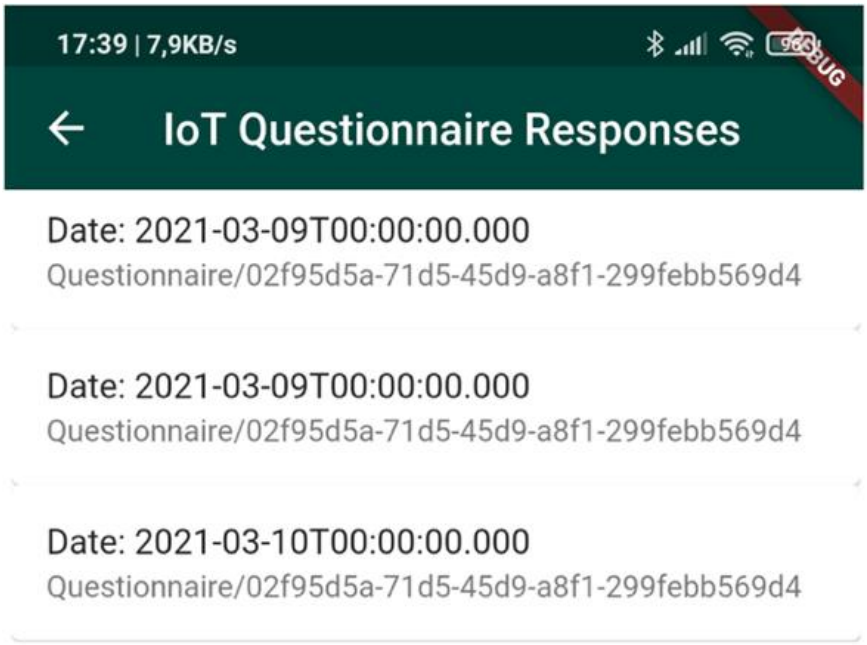
ID: S15.2
Name: Wrong application key
Priority: High
Evaluation: ✓ Passed

Description: The testers verify that any attempt trying to ship logs without the proper encryption key is rejected.

```
2021/03/25 13:25:19 ossec-agent: WARNING: (4101): Waiting for server reply (not started). Tried: '192.168.142.109'.
2021/03/25 13:25:19 ossec-agent: WARNING: Unable to connect to any server.
2021/03/25 13:25:19 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:19 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:29 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:29 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:39 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:39 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:49 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:49 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:59 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:25:59 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
2021/03/25 13:26:00 ossec-agent: INFO: Requesting a key from server: 192.168.142.109
2021/03/25 13:26:02 ossec-agent: ERROR: Unable to connect to 192.168.142.109:1515
2021/03/25 13:26:12 ossec-agent: WARNING: (4101): Waiting for server reply (not started). Tried: '192.168.142.109'.
2021/03/25 13:26:12 ossec-agent: WARNING: Unable to connect to any server.
2021/03/25 13:26:12 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:26:12 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
2021/03/25 13:26:22 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:26:22 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
2021/03/25 13:26:32 ossec-agent: INFO: Closing connection to server (192.168.142.109:30307/tcp).
2021/03/25 13:26:32 ossec-agent: INFO: Trying to connect to server (192.168.142.109:30307/tcp).
```

AT16: Establish secure communication

ID: S16.1
Name: IoT device establishes a connection
Priority: High
Evaluation: ✓ Passed
Description: The testers executed a routine in the emulated IoT device that could connect with the Data Gateway by using a JWT refresh token**.**



ID: S16.2
Name: IoT device beyond the range of communication
Priority: Outdated
Evaluation: -
Description: This scenario no longer fit in the ecosystem of the project technologies.

ID: S16.3
Name: Connection already established
Priority: Outdated
Evaluation: -
Description: This scenario no longer fit in the ecosystem of the project technologies.

ID: S16.4
Name: Network glitch
Priority: Outdated
Evaluation: -
Description: This scenario no longer fit in the ecosystem of the project technologies.

ID: S16.5
Name: Failed the proof-of-possession challenge
Priority: Outdated
Evaluation: -
Description: This scenario no longer fit in the ecosystem of the project technologies.

ID: S16.6
Name: Unregistered device
Priority: Outdated
Evaluation: -
Description: This scenario no longer fit in the ecosystem of the project technologies.

AT17: Send medical data

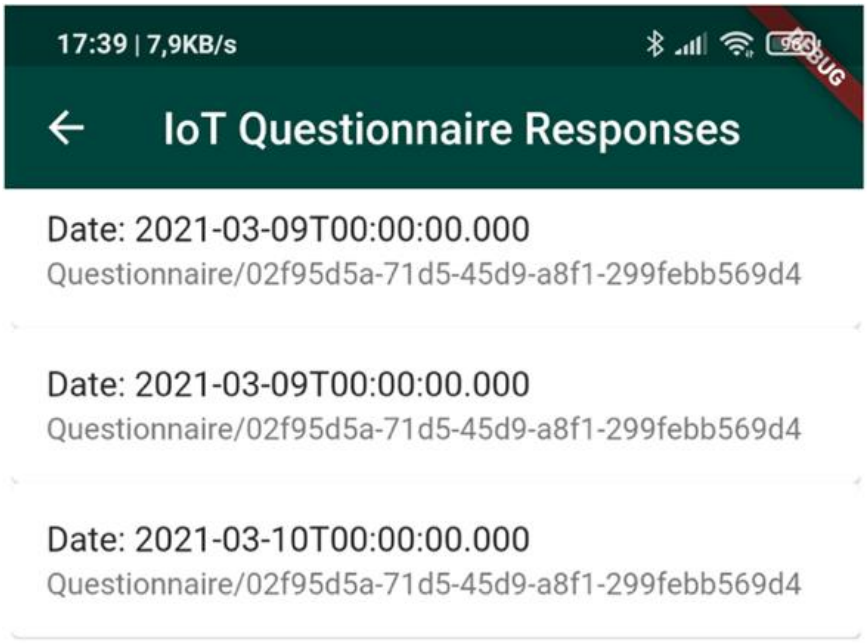
ID: S17.1

Name: IoT device sends some medical data

Priority: High

Evaluation: ✓ Passed

Description: The testers executed a routine in the emulated IoT device that using a hardcoded refresh token communicated with the Data Gateway. The medical data that was pre-created in the emulated IoT device is sent to the Data Gateway and stored securely.



ID: S17.2

Name: IoT device gets disconnected

Priority: Outdated

Evaluation: -

Description: In the course of the development of the IoT solution, the current functionality has become outdated as it was based on the use of a Bluetooth communication protocol, and has been changed to REST requests in which there is not a constant connection and, instead, connections are stablished and closed for each request. This change will be reflected in the next release.

ID: S17.3

Name: Session error

Priority: Outdated

Evaluation: -

Description: Please refer to the description above.

AT18: Report suspicious activity

ID: S18.1
Name: Mobile agent reports suspicious activity
Priority: High
Evaluation: ✓ Passed

Description: For demonstration purposes, the functionality has been tested by manually pushing a warning (I.e., suspicious activity detected for device ID#12345) on Kafka, and by verifying that the SIEM received the corresponding notification. Testing with continuous authentication is yet to be implemented.



ID: S18.2
Name: Mobile device is not registered
Priority: Medium
Evaluation: ✓ Passed

Description: For demonstration purposes, the functionality has been tested by manually pushing a warning (I.e., mobile device is not registered) on Kafka, and by verifying that the SIEM received the corresponding notification. Testing with continuous authentication is yet to be implemented.



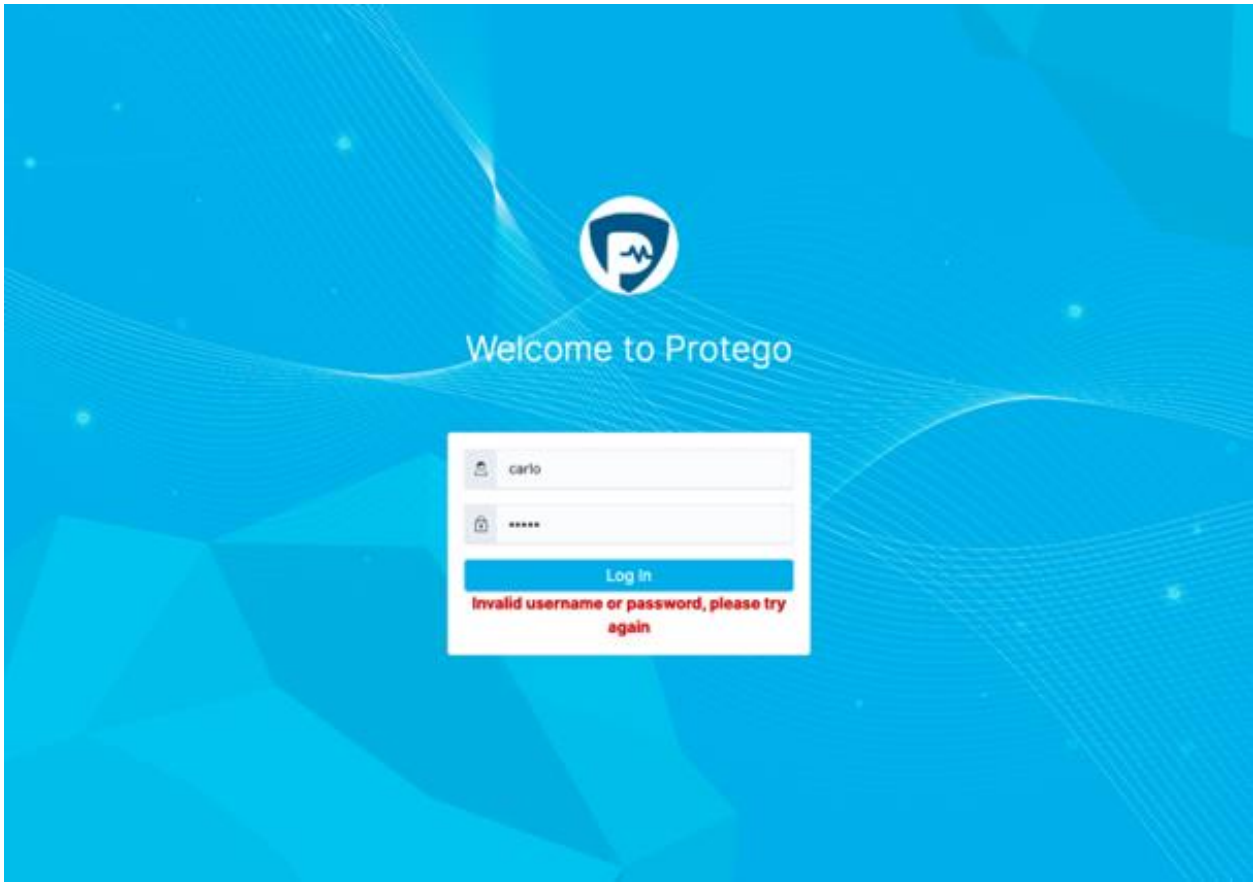
AT19: Respond to alert

ID: S19.1
Name: Security operator responds to an alert
Priority: High
Evaluation: ✓ Passed
Description: The testers attempted to log into FoodCoach as a nutritionist with the wrong credentials. After three unsuccessful attempts, an alert was raised to warn of a possible ongoing brute force attack.



ID: S19.2
Name: Insufficient authorization
Priority: High
Evaluation: ✓ Passed

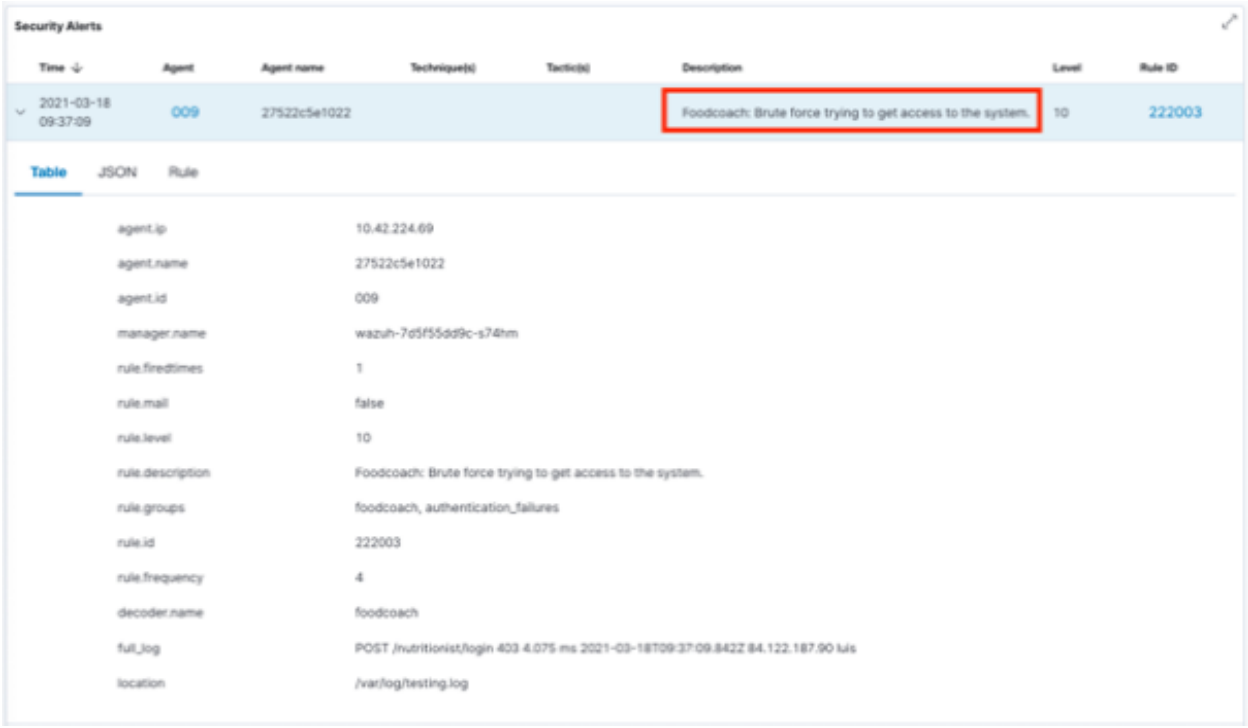
Description: The testers verified that only authorized users can access to this functionality.



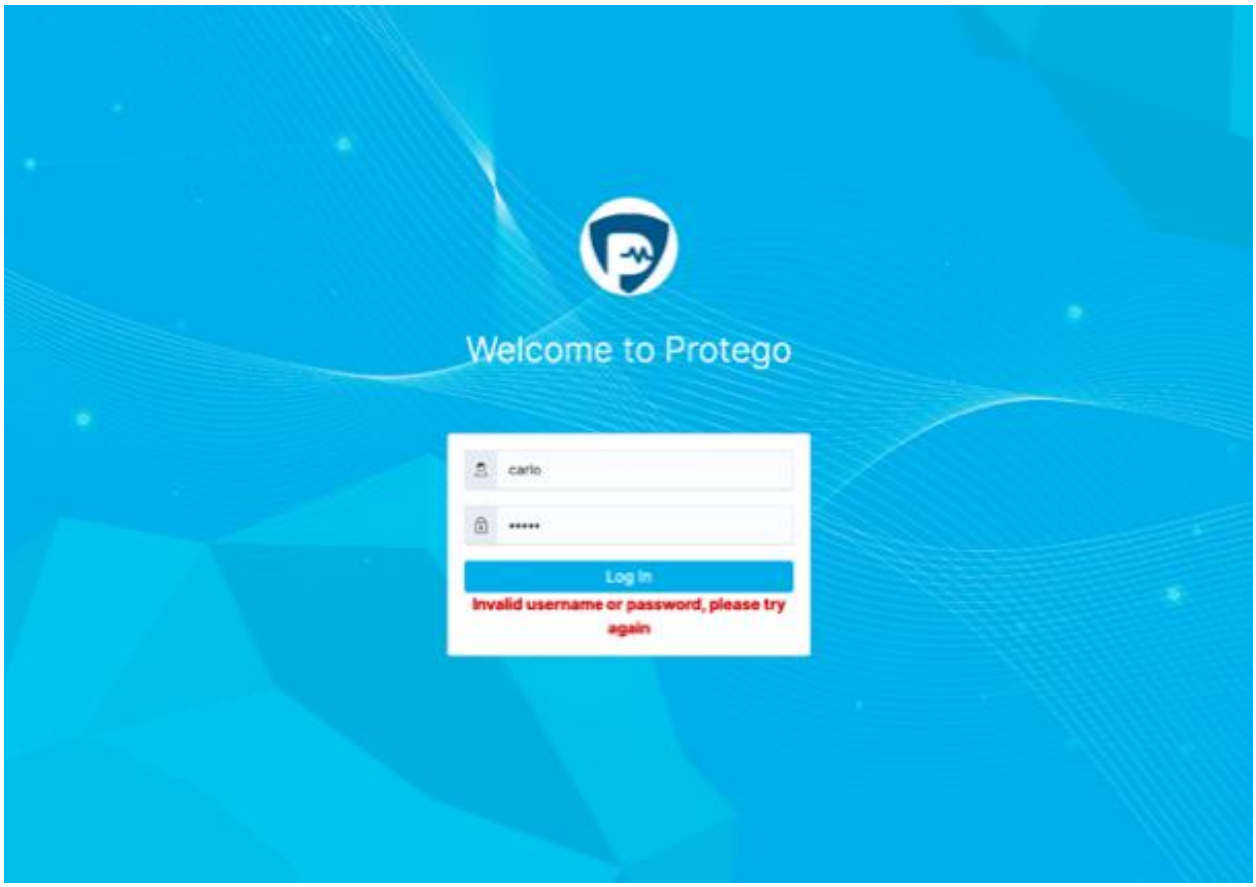
AT20: Review alerts

ID: S20.1
Name: Security operator review alerts
Priority: High
Evaluation: ✓ Passed

Description: The testers logged into ProTego and requested to review the alerts. The system displayed the alerts and their details.



ID: S20.2
Name: Insufficient authorization
Priority: High
Evaluation: ✓ Passed
Description: The testers verified that only authorized users can access this functionality.



AT21: Review new risk evaluation

ID: S21.1

Name: Security operator reviews changes in the evaluation

Priority: Medium

Evaluation: ✕ Failed

Description: For demonstration purposes, the testers manually requested a recalculation of the risks. The recalculation is not reflected on the SSM at the moment.



ID: S21.2

Name: Insufficient authorization

Priority: Medium

Evaluation: ✕ Failed

Description: The current functionality has not been integrated yet. The recalculation is not reflected on the SSM at the moment.

ID: S21.3

Name: First time risk assessment is still to be conducted

Priority: Medium

Evaluation: ✕ Failed

Description: The current functionality has not been integrated yet. The recalculation is not reflected on the SSM at the moment.

AT22: Reflect infrastructure changes

ID: S22.1
Name: Security operator reflects recent changes
Priority: Medium
Evaluation: ✕ Failed

Description: The current functionality has not been integrated yet.

ID: S22.2
Name: Insufficient authorization
Priority: Medium
Evaluation: ✕ Failed

Description: The current functionality has not been integrated yet.

ID: S22.3
Name: First time risk assessment is still to be conducted
Priority: Medium
Evaluation: ✕ Failed
Description: The current functionality has not been integrated yet.

VI.2. Usability Inspection Evaluation

ID: IN01

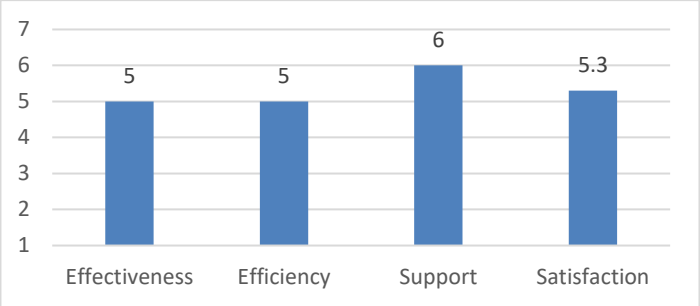
Name: Cluster deployment

Effectiveness: 5/7

Efficiency: 5/7

Support: 6/7

Satisfaction: 5,3/7



Description: Overall the installation process was simple to follow. A few minor improvements can be done. For example, to simplify things, the Ansible scripts could be launched from the tester's computer, instead of the control node of the cluster. The installation process requires some time, however, that is understandable because of the complexity of the task. The partners provided a tutorial that was comprehensive of all the necessary details.

ID: IN02

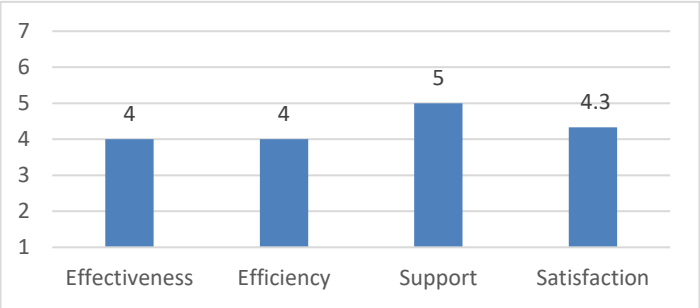
Name: Install Data Gateway

Effectiveness: 4/7

Efficiency: 4/7

Support: 5/7

Satisfaction: 5/7



Description: Overall the installation process was simple to follow. However, updating the component required some effort related to versioning of the Docker image. This could be improved for the next release to reduce the required time. The partners provided the necessary support to overcome the emerged issues.

ID: IN03

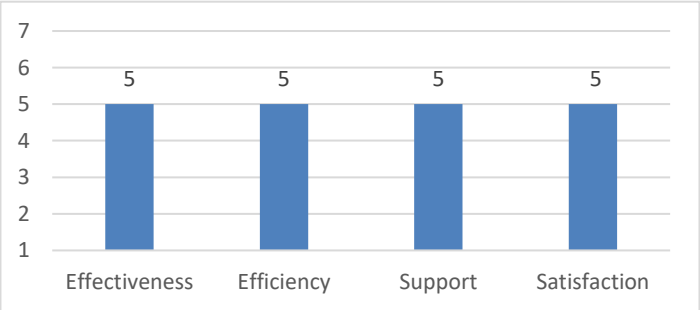
Name: Install Network Slicing

Effectiveness: 5/7

Efficiency: 5/7

Support: 5/7

Satisfaction: 5/7



Description: Overall the installation process was quite simple to follow. A few minor improvements can be done. For example, the base image of the controller is Ubuntu, but the version is not specified. The version should be added to avoid reproducibility issues. Moreover, the YML files should be delivered separately from the PDF, because writing them directly in the PDF could mess up the indentation.

The partners provided the necessary support; however, the installation process took around two weeks which can be reduced in the next release by writing a tutorial that describes the necessary steps for the component deployment in a clear and concise way.

ID: IN04

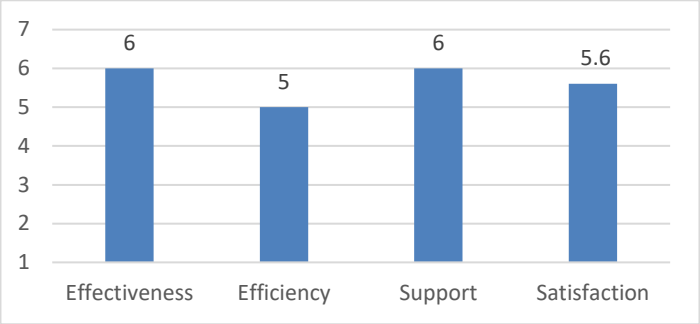
Name: Install SIEM

Effectiveness: 6/7

Efficiency: 5/7

Support: 6/7

Satisfaction: 5,6/7



Description: The installation process was simple to follow. The component is inherently complex because it is made up of many sub-components. The partners provided a thorough description of the components by including also a presentation to support the activity. The required amount of time to perform the installation process was acceptable. Some extra time was required to set up a way to exchange logs between the application and the SIEM.

ID: IN05

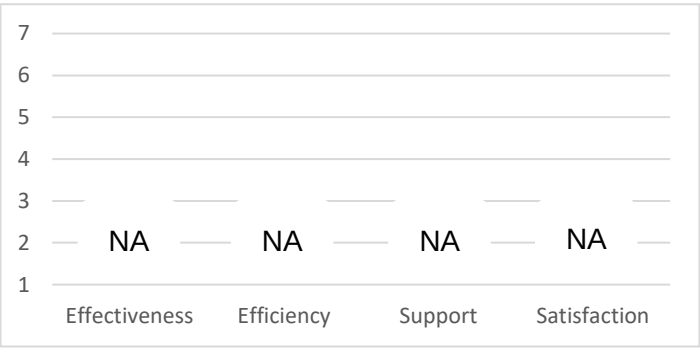
Name: Install SSM

Effectiveness: NA

Efficiency: NA

Support: NA

Satisfaction: NA



Description: The installation of the SSM cannot be evaluated in the usability inspection because**,** at the current stage of the project, the SSM has been used as a service, which was deployed outside the hospital premises.

ID: AT01

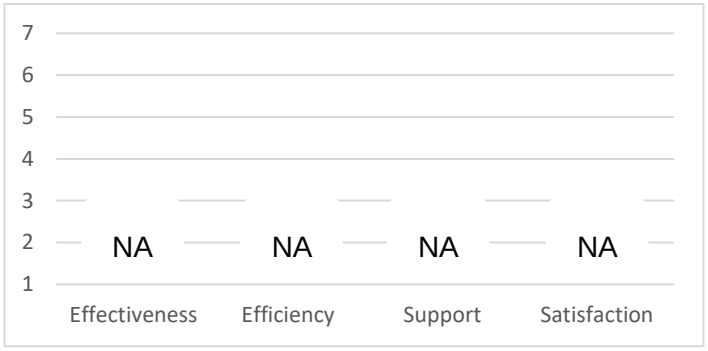
Name: Register user

Effectiveness: NA

Efficiency: NA

Support: NA

Satisfaction: NA



Description: Each component offers its own way to register users. For this reason, there is not a single centralized feature that can be evaluated and the task has been marked as Not Applicable.

ID: AT02

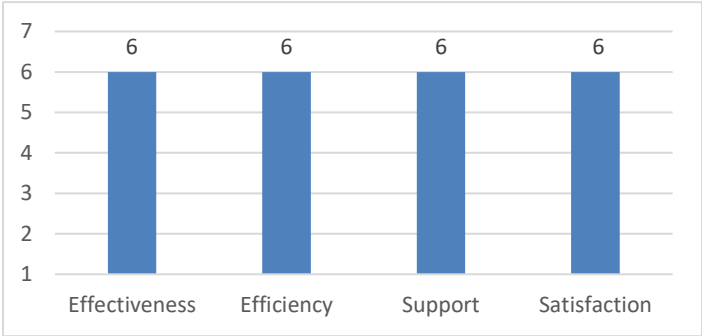
Name: Conduct first-time risk assessment

Effectiveness: 6/7

Efficiency: 6/7

Support: 6/7

Satisfaction: 6/7



Description: The task was quite simple to reproduce despite the inherent complexity of the component, thanks to the comprehensive tutorial slides. Moreover, the partner helped the testers in creating the infrastructural model.

ID: AT03

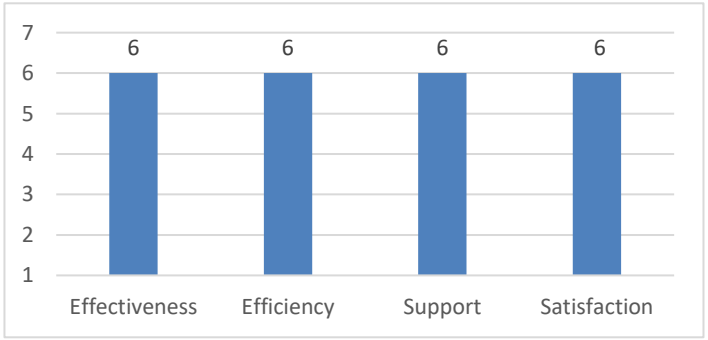
Name: Assess prospective risks to the infrastructure

Effectiveness: 6/7

Efficiency: 6/7

Support: 6/7

Satisfaction: 6/7



Description: The task was quite simple to reproduce despite the inherent complexity of the component, thanks to the comprehensive tutorial slides. Moreover, the partner helped the testers in creating the infrastructural model.

ID: AT05

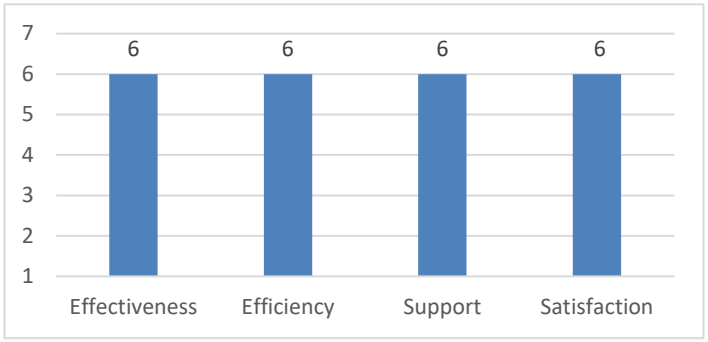
Name: Configure application network slices

Effectiveness: 6/7

Efficiency: 6/7

Support: 6/7

Satisfaction: 6/7



Description: The task was quite simple to perform. Indeed, the UI made the configuration simple and intuitive and just a few clicks were required. The partner provided a clear tutorial to support this activity.

ID: AT06

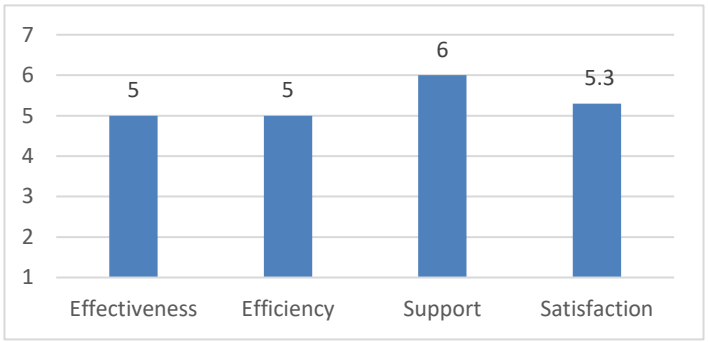
Name: Configure application logging mechanism

Effectiveness: 5/7

Efficiency: 5/7

Support: 6/7

Satisfaction: 5,3/7



Description: The SIEM is made up of 13 sub-components and one of them must be installed along with the application. This is the easiest way to read the application logs, however for future development it is suggested to adopt a less invasive configuration. The configuration of the other sub-components required almost no interaction, as everything was set up as necessary by default.

ID: AT07

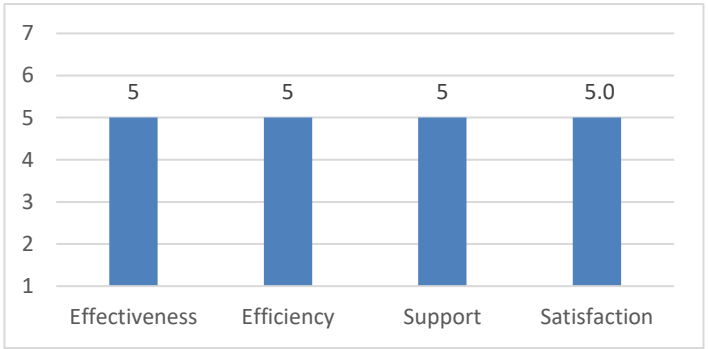
Name: Specify application access control

Effectiveness: 5/7

Efficiency: 5/7

Support: 5/7

Satisfaction: 5/7



Description: The configuration of access control policies is currently hardcoded in the component. Thus, for future development it is suggested to provide a more user-friendly way to customize the configuration. Moreover, in a more mature state of development it would be desirable to configure the access control policy with more fine-grained rules.

ID: AT11

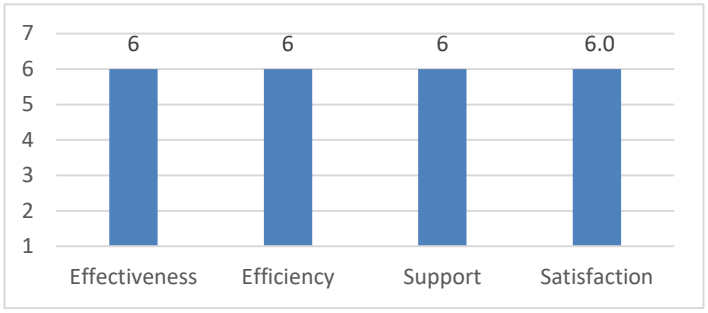
Name: Register mobile device

Effectiveness: 6/7

Efficiency: 6/7

Support: 6/7

Satisfaction: 6/7



Description: The tester configured the MAC address of the device within the UI, that made this operation simple and intuitive. The operation steps were provided by the partner in a clear manner and they required a very short time.

ID: AT19

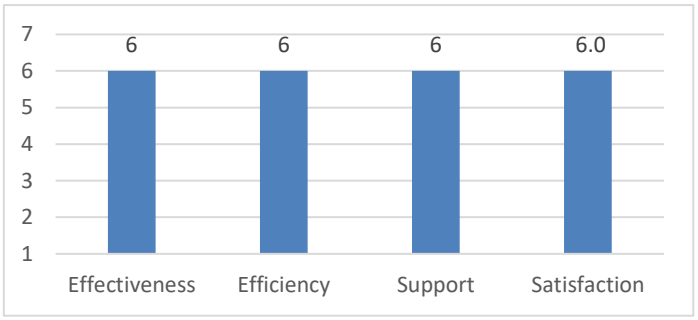
Name: Respond to alert

Effectiveness: 6/7

Efficiency: 6/7

Support: 6/7

Satisfaction: 6/7



Description: The alert functionality can be customized, so the tester can be prompted with relevant information to identify a possible threat to the system. In the case of an alert raised after multiple attempts to access an account with the wrong credentials, the system displays some details such as: the type of threat, the application that is undergoing a possible attack, the geolocation of the potential attacker.

ID: AT20

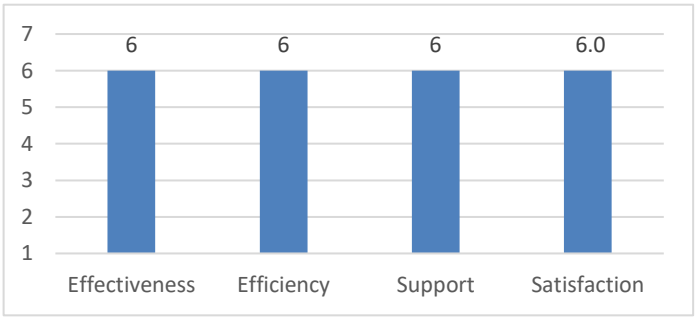
Name: Review alert

Effectiveness: 6/7

Efficiency: 6/7

Support: 6/7

Satisfaction: 6/7



Description: Despite the fact that the dashboard displays several functionalities, the navigation that leads to the requested functionality is quite intuitive. Once the user reached the page to review an alert, he could easily select the alert that he wanted to review.

VI.3. Toolkit Assembling

This section describes the work involved with deployment of the ProTego Integration Toolkit (WP6) with the infrastructure provided by Ospedale San Raffaele and Marina Salud to create testbeds for their respective case studies.

The Integration Toolkit is based around Kubernetes and other open-source technologies to allow for the integration of the technical components for Risk Assessment (WP4) and Risk Mitigation (WP5).

The following figure shows the component diagram for the Integration Toolkit. For more detailed information, see deliverable D6.2 – Intermediate prototype Architecture, requirements and integrated toolkit.

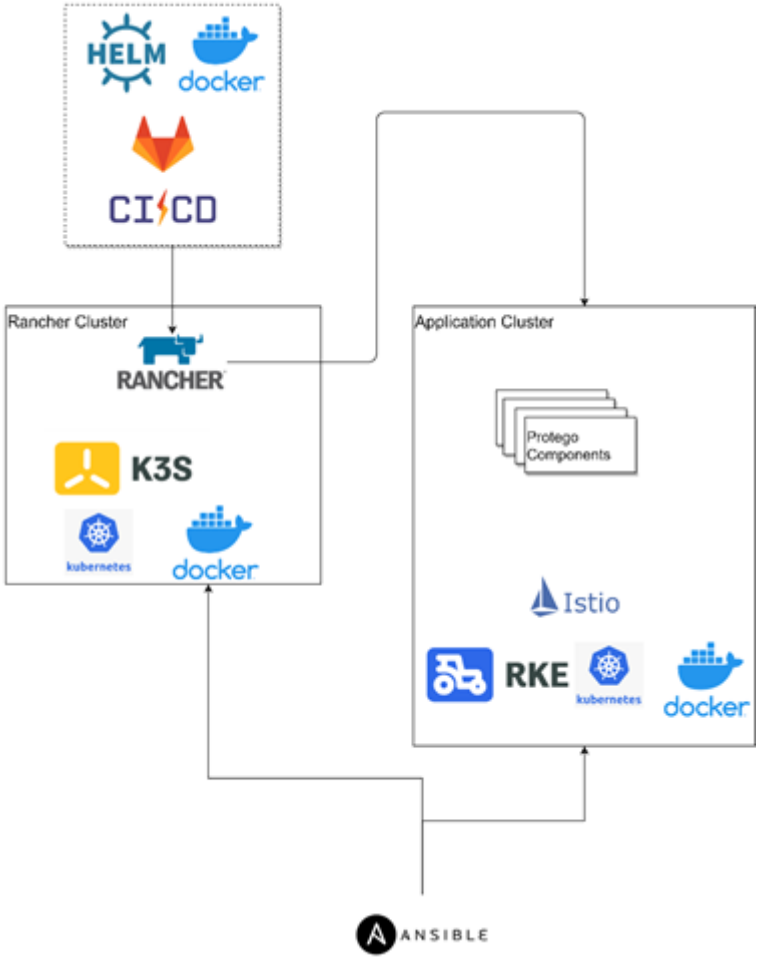


Figure VI.3-1 Component Level Diagram for the Integration Toolkit

As displayed in Figure VI.3-1, the components described below have been deployed to the testbeds.

Ansible: a configuration management tool, it allows for remote deployment and configuration of software. IT admin users from the hospitals used the Ansible playbooks provided by ICE to deploy the integration stack, which is comprised mainly of Kubernetes and Rancher.

Docker: the standard in container technology, used to package the components and is installed using the Ansible playbooks.

Kubernetes: the standard in container orchestration technologies, used to deploy the components and integrate them. Installed using the Ansible playbook. Kubernetes distributions being used are K3s, a lightweight Kubernetes distribution to deploy the Rancher cluster in the two use cases. Rancher RKE distribution is being used as the application cluster for the OSR use case and AWS EKS for the Marina Salud use case.

Helm: package manager for Kubernetes. Kubernetes description files for ProTego components are packaged as a Helm Chart for every component. It provides a way to deploy all necessary Kubernetes resources as a single unit. Helm is installed using the Ansible playbook.

Rancher: addresses the operational challenges of managing Kubernetes clusters. Installed using the Ansible playbook.

GitLab: git server used to provide the repository for Docker images and Helm Charts. Rancher is able to "pull", that is, download the container from a GitLab server and deploy the components to Kubernetes. For the use case, ProTego is using the GitLab Cloud version.

Istio: service mesh technology, has not been deployed for this stage of the project, and will be deployed and used during final integration and tests. It will be installed using the Ansible playbook.

CI/CD: continuous integration and continuous deployment pipelines haven't been used at this stage of the project, and will be configured for the next phase using GitLab and Kubernetes.

VI.3.1. Generic Deployment Steps

In order to deploy the integration toolkit, ICE gathered the platform requirements for Kubernetes and Rancher installation, and an estimation of CPU & memory utilization by ProTego components. Those requirements were sent to the users of the use-cases in order for them to provide the infrastructure needed, i.e., virtual machines with the necessary amount of cpu and memory.

Figure VI.3-2 depicts the architecture of the integration toolkit.

The general architecture is comprised of the Rancher cluster, a K3s based Kubernetes cluster meant to deploy Rancher, and the application cluster, a RKE based Kubernetes cluster to deploy the ProTego components from WP 4-5.

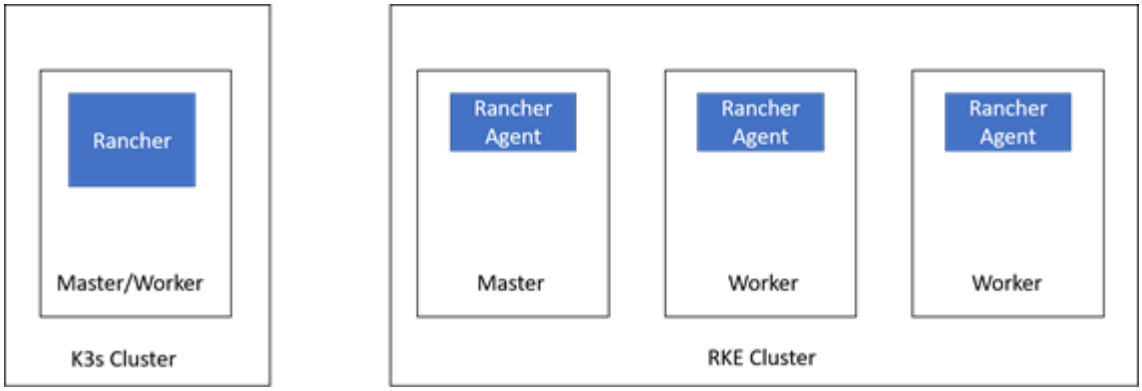


Figure VI.3-2 Integration Toolkit Architecture Diagram

General steps to deploy the Integration Toolkit are:

- Create the installation user and install requirements (Ansible)
- Install docker if it is not installed (Ansible)
- Install the K3s cluster (Ansible)
- Deploy Rancher to the K3s cluster (Ansible)
- Install the RKE cluster (Ansible)
- Register the RKE cluster in Rancher (Rancher)
- Setup Rancher with the catalog to deploy the technical components from WP4 and WP5 (Rancher)
- Deploy the ProTego components from WP4-5 (Rancher)

In addition to this, deployment for each case study was different since OSR deployment was on-premise while the Marina Salud one was on-cloud using AWS.

General support was provided to both case studies in order to deploy the integration toolkit and the components. Support was also provided at the component level to help partners from WP4 and WP5 design the Helm charts and deploy the components to Kubernetes.

In the following sections a description of the specific deployment for each case study is presented.

VI.3.2. OSR

This deployment of the integration toolkit was fully on-premise. Requirements were sent to the admin users who provided 4 nodes (virtual machines) with the specified cpu and memory resources.

Detailed instructions were provided by ICE so that admin users from the hospital were able to deploy the Kubernetes cluster and Rancher themselves, using the Ansible playbooks.

The architecture diagram for this use-case is the same than the one shown in Figure IV.4-2 Integration Toolkit Architecture Diagram.

The installation with the Ansible playbooks performed the tasks described in the list of general steps in VI.3.1, above.

VI.3.2.a. Issues

Regarding the deployment and use of the integration stack, only one issue related to networking arose.

After deploying the first components to the cluster, there were some communication issues in accessing the components. There was a bug in one component, which was fixed. But communication issue persisted, so we had to recreate the RKE cluster, setting the specific network interface for cluster communication. That solved the issue.

VI.3.2.b. Support

Some support was required in order to help troubleshooting some issues and start deploying components.

Additional help and instructions were provided on how to use Rancher in order to deploy the components and monitor them.

Some support was also provided to use Kubernetes command line utility kubectl.

Help in deploying and integrating ProTego components from WP 4-5.

Some instructions on how to use basic Rancher functionality were provided.

VI.3.3. Marina Salud

Deployment at Marina Salud was fully on-cloud using Amazon AWS.

For this deployment, an AWS EC2 instance (virtual machine) was provisioned by Marina Salud, then ICE deployed the K3S cluster and Rancher using the Ansible playbooks. After that, Marina Salud provisioned an EKS Kubernetes Cluster as the application cluster, and registered the cluster in Rancher.

For the Rancher cluster the steps are the same as the general ones but on an AWS EC2 instance:

- Create the installation user and install requirements at the EC2 instance (Ansible)
- Install docker (Ansible)
- Install the K3s cluster (Ansible)
- Deploy Rancher to the K3s cluster (Ansible)

For the application cluster, AWS EKS, the managed Kubernetes service from AWS, was used. Marina Salud provisioned such cluster and registered it in Rancher themselves. The deployment steps were slightly different to the general ones:

- Provision the application cluster (AWS EKS)
- Register the cluster in Rancher
- Setup Rancher
- Deploy components

VI.3.3.a. Issues

There was a communication issue in order to access the cluster network from an external private network that was preventing access to the ProTego components from WP 4-5 deployed in that cluster. That got solved setting some network configuration at the AWS networking level by Marina Salud.

VI.3.3.b. Support

Initial help was provided for deployment and integration of the Data/Query Gateway component from WP5.

Some instructions on how to use basic Rancher functionality were provided.

VII. REFERENCES AND INTERNET LINKS

VII.1. Internet Links

[1] K. Wiegers, and J. Beatty, “Software requirements,” Pearson Education, 2013.

[2] S. Covey, “The 7 Habits of Highly Effective People,” New York: Free Press, 2004.

[3] K. Nicieja, “Writing Great Specifications: Using Specification by Example and Gherkin,” Manning Publications, 2017.

[4] A. William, and T.Tullis, “Measuring the user experience: collecting, analyzing, and presenting usability metrics,” Newnes, 2013.

[5] J. R. Lewis, “Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ.” SIGCHI Bull. 23, 1, 1991.


ProTego