Grant Agreement N° 780351

Title: *First evaluation and validation Report*

Authors: *Sergio Jiménez Gómez (INDRA), Rocío Gómez Robledo (INDRA), Jon Colado García (INDRA), Francisco Parrilla Ayuso (INDRA), Erkuden Rios Velasco (TECNALIA), Eider Iturbe Zamalloa (TECNALIA), Angel Rego Fernandez (TECNALIA), Miguel Angel Anton Gonzalez (TECNALIA), Sarah Tiphaine Ada Noye (TECNALIA), Arnor Solberg (TellU), Franck Fleurey (TellU), Uģis Grīnbergs (BOSC), Modris Greitans (EDI),Oscar Zanutto (ISRAA), Emanuela Capotosto (ISRAA),* Nicolas Ferry, (*SINTEF*) and *Wissam, Mallouli (MONTIMAGE)*

Editor: *INDRA*

Reviewers: *Nicolas Ferry, (SINTEF), Hui Song, (SINTEF) and Wissam, Mallouli (MONTIMAGE)*

Identifier: *D1.3*

Nature: *Deliverable*

Date: *31 January 2020*

Status: *Final*

Diss. level: *Public*

Executive Summary

The objective of the deliverable D1.3 is to describe the validation and verification status and procedures for the implementations performed during the ENACT project first period (M22).

As stated in the previous deliverables D1.1 and D1.2, the use cases represent the platforms to demonstrate that the ENACT tools developed in the WP2, WP3, and WP4 are functional for all of them. The use cases environment comprises all the IoT infrastructures that are needed to perform development and operation of the use cases, including IoT and edge devices, the integration platforms and the cloud services. In this deliverable, the implementations developed during the first period of the project (until month M22) are tested against the use cases requirements and KPIs stated in the D1.1 to probe that the ENACT tools are valid to be integrated into different kind of environments, in a generic manner, using the infrastructure previously described in the D1.2. The results from this report will be used as feedback to the final developments to be done in the second period of the ENACT project.

As a reminder, the three different use cases implemented in ENACT are:

- **Intelligent Transport System (ITS):** The purpose is assessing the feasibility of IoT services in the domain of train integrity control, in particular for the maintenance and logistics of the rolling stock and the on-track equipment.
- **Digital Health:** The objective of this use case is providing medical device integration and data transport capabilities to external services such as an alarm center or an electronic patient journal or stakeholders such as patients, doctors etc., exploiting the potential in IoT, edge and cloud services, adding more devices, actuation, distributed processing and better exploitation of collected data.
- **Smart Building:** The aim is generating Smart Energy Efficiency applications and Smart Elderly Care applications, which will make use of sensors, actuators and services, in order to ensure the safety of the facilities to perform energy efficiency measures and to support the caretakers in assessing the wellbeing of users.

This deliverable is the outcome of task T1.3 lead by TellU, which is responsible for the Digital Health Use Case demonstrator as a validation and verification platform. Indra and Tecnalia are responsible of assessing that the development and integration of the ITS and Smart Building demonstrators respectively with the ENACT tools fulfill the requirements stated in the D1.1. BOSC and EDI will contribute to the validation and verification tasks of the ITS case study and facilitate the rail test infrastructure and demonstrations. ISRAA will provide support for the Smart Building Use Case into the validation and verification tasks.

The deliverable is a bit delayed due to the following reasons:

- We spent more effort than expected on the implementation of the systems that are used as the bases for use case evaluation, including building the hardware environment and developing the software applications. As a result, the process of applying ENACT enablers on these systems were postponed accordingly.
- The requested revised revision of D1.1 was submitted in the end of Month 16 (first version submitted Month 6), and it was natural to include the actual state at the submission time for the revised version and adjust the evaluation focus and the concrete plans. Since we believe these

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

adjustments are important to improve the quality of the project, we decided to spend more time on the deliverable to ensure the adjustments are properly applied to all the use cases.

- After getting the feedback from the mid-term review, we spent some extra time on further revising the evaluation plans to reflect the changes of the project focuses according to the reviewers' advices.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Members of the ENACT consortium:

SINTEF AS	Norway
Beawre	Spain
EVIDIAN SA	France
INDRA Sistemas SA	Spain
FundacionTecnalia Research & Innovation	Spain
TellU AS	Norway
Centre National de la Recherche Scientifique	France
Universitaet Duisburg-Essen	Germany
Istituto per Servizi di Ricovero e Assistenza agli Anziani	Italy
Baltic Open Solution Center	Latvia
Elektronikas un Datorzinatnu Instituts	Latvia
Montimage	France

Revision history

Date	Version	Author	Comments
15.10.2019	0.0	Sergio Jiménez Gómez	Table of Contents
25.10.2019	0.1	Sergio Jiménez Gómez	First draft
26.11.2019	0.2	Arnor Solberg	TellU contribution
20.12.2019	0.3	Miguel Angel Anton	Tecnalia contribution
09.01.2020	0.4	Nicolas Ferry, Hui Song, Jon Colado, Sergio Jiménez, Francisco Parrilla	Indra last contribution and review
16.01.2020	0.5	Jon Colado, Sergio Jiménez, Francisco Parrilla, Wissam Mallouli, Nicolas Ferry, Hui Song	Edition changes and internal revision
30.01.2020	0.6	Jon Colado, Sergio Jiménez, Francisco Parrilla, Miguel Angel Anton, Arnor Solberg	Final version

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Contents

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Table of Figures

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

1 Abbreviations and Definitions

Term	Definition
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ARM	Advanced RISC Machine
CTC	Centralized Traffic Control
DevOps	Development and Operations
eGW	Enhanced Gateway
EPJ	Electronic Patient Journals
ERTMS	European Rail Traffic Management System
GNSS	Global Navigation Satellite System
GRDP	General Registry of Data Protection
GSM-R	Global System for Mobile Communications – Railway
GUI	Graphical User Interface
GW	Gateway
HLA	High Level Architecture
HVAC	Heating, Ventilation and Air Conditioning
ID	Identification
IoT	Internet of Things
ISO	International Organization for Standardization
ITS	Intelligent Transport Systems
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
L&M	Logistics and Maintenance
L&M	Logistics and Maintenance
LED	Light-Emitting Diode
MQTT	Message Queue Telemetry Transport
MW	Middleware
OPC-UA	OPC Unified Architecture
OPEX	Operating expense
OTI	On Board Train Integrity
PLC	Programmable Logic Controller
QoS	Quality of Service
RCA	Root Cause Analysis
RFID	Radio Frequency Identification
STCC	Smart Train Composition Coupling
SW	Software
TCP/IP	Transmission Control Protocol / Internet Protocol
TDS	Train Detection Systems
V&V	Validations and Verification

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

VPN	Virtual Private Network
WSN	Wireless Sensor Network

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

2 Introduction and Objectives

The main goal of this deliverable is to report on the first version evaluation of the ENACT tools into the use cases. The Intelligent Transport System, the Digital Health, and the Smart Building use cases serve as evaluation platforms from different industrial environments and business, with different trustworthiness requirements.

The requirements variety provided by these use cases is wide enough to ensure the ENACT tools can be applied to different domains. The implementations done during the first period must go throw the validation and verification tests stated by the Use Cases and established on this document. This document shows results validating the current ENACT tools developments into the use cases. Therefore, there is a basis to continue with the second ENACT project period.

The document is composed of the "Use Case Validation and Verification" section that shows the following points:

- Assesses the status of the tools with respect to the requirements and KPIs status at M22 besides their expectations at the end of the project.
- Shows the tests used to assess the KPIs and if requirements are met, following a specific validation and verification process introduced in D1.1.

Two type of tests are used to assess the KPIs and requirements:

- **Use Case tests:** Specific test to measure the use case systems capabilities to be integrated with the ENACT tools. It must be considered that these tests (defined in the D1.1 for each use case) describe adaptations to maximize the effect of the ENACT tools. These efforts are focus on improving the integration compatibility to cover a wider range of tools in a safety and secure manner, the validation and verification process of the Use Case-ENACT tools tests considers these adaptations.
- **Use Case-ENACT tools tests:** These tests measure the ENACT tools implementation degree into the Use Cases as well as the degree of fulfillment of the requirements and KPIs. All these tests are described into the annexes.

Once the objectives and the type of tests are presented, the methodology to evaluate the ENACT tools (details explained in next section 3, getting their M22 status into the Use Cases, is explained:

1. We defined requirements and some tests to assess how these requirements are addressed by the ENACT enablers.
2. We report the current status at M22 of the enablers with respect to these tests.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

3 Use Case Validation and Verification

The Validation and Verification section introduces the methodology used to generate the test and the KPIs, as it is the method to extract the objectives to be proven.

As the project is divided into three well-differentiated use cases (ITS, Digital Health, and Smart Building). All of them are designed independently besides their Validation and Verification tests keeping a common Validation and Verification methodology for all of them. Moreover, different system tests scenarios are proposed to integrate and validate the enablers that are used.

The key to understand these tests is that the ENACT tools must improve the development and operation (DevOps), including the trustworthiness aspects, of the use case without disturbing its business functionalities. The integration between the ENACT tools and the use cases is accomplished in a non-intrusive manner to avoid this mentioned conflict.

The improvements that the ENACT tools add to the use case is tackled through the procedures, constrains, and inputs. It must be considered the fact that the use case business functionalities and needs are not interfered.

The KPIs are obtained through a defined methodology. This methodology is focused on measuring the magnitudes that are relevant to show how the ENACT tools are affecting, as planned in the initial definition, the Use Cases. The methodology to get these KPIs is divided into the following three stages matched with the GQM [3] methodology:

- **STAGE 1**: Design of the Impact Assessment Roadmap for KPIs as a project level Cod Objectives Timeframe.
- **STAGE 2:** Performance of the KPI collection activities in a specific timeframe (M1-M22 in the ENACT project).
- **STAGE 3:** Evaluate the performance targets reached at the end of the project.

The first and second stages are covered in the first period of the project getting a first view of the ENACT tool progress in the use cases. During the last period of the Project (M22 to M39) the stage 3 will be reached in order to show the improvements that ENACT generates. This step involves the following sequenced objectives:

- Evaluate the results obtained in order to draft the conclusions resulting from tests execution.
- Document the results achieved as well as draft a first set of conclusions.
- Present the results of ENACT KPIs project level evaluation to the project partners.

The match that ENACT is establishing among these two points is the fact that the KPIs are divided in technical objectives following the ENACT tool groups:

- **TO1:** Support continuous delivery of trustworthy SIS.
- **TO2:** Support the agile operation of trustworthy SIS.
- **TO3:** Support continuous quality assurance strengthening trustworthiness of SIS.
- **TO4:** Leverage the capabilities of existing IoT platforms and fully exploit legacy, proprietary and off-the-shelf software components and devices.

Therefore, the KPIs and the Validation and Verification tests are focus on probing that the ENACT tools and the Use Cases can be functional once they are integrated following the ENACT project objective.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

The following sections are intended to summarize all this methodology to Verify and validate the current developments performed during the project first period, described in the deliverables D1.2 [2].

The evaluation progress stated is shown in two tables (one table for Use Case tests and another one for Use Case-ENACT tools tests). The first table shows the tests defined for the ENACT tools integration and functionality in the use case and the second shows the use case specific testing features.

The tables use the next colours to evaluate the progress degree:

- Green (Evaluation and validation is performed and KPI is fully completed)
- Yellow (Some Evaluation and validation is performed and KPI is partly completed)
- Orange (Some Evaluation and validation is performed KPI is not fulfilled)
- Red (Evaluation and validation is not started)

Considering both type of test, the general results for the validation and verification report is that the completeness level is set to 50%. This value is extracted from the fact that every use case development are almost competed except last detail. This percentage is reduced as not all the initial planned functionalities are fully complied, it is estimated that the ENACT tools into the use cases status is 40%, based on the number of objectives completed.

3.1 ITS Domain (Rail)

A set of different needs were reported to the ENACT enabler providers for the operation and development of an ITS use case that has safety and security requirements:

- Remote software update for a huge quantity of devices
- System scalability tools
- Security supervision at different layer and failures detection
- Health data systems monitoring
- Failure analysis
- Actuation measurements against security threats.

The Evaluation and Validation of ENACT results in the first period of the project applying the ITS use case has included:

- The Evaluation and Verification from the ITS use case perspective, without including the ENACT tools integration, throws a 75% of completeness level considering the adaptations made into the ENACT project framework. This estimation is based on the number of initial works planned completed. It is still necessary enabling in the ITS communication middleware the network data provision in order to monitor network security.
- Considering the ENACT tools, the tests stated at the beginning of the project for the Logistics and Maintenance scenario are not fully completed until all the developments are set in a final version. Only some of the planned ENACT tools are integrated in a partial development degree (GeneSIS and RCA), all the planned tools in a 100% status will be integrated at the end of the project.

The progress, considering both the ITS Use Case tests and the ENACT tool into the Use Case tests, is 50% based on the number of tests completed.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

An overall summary of the evaluation and validation performed applying the ITS use case is shown in the table below. The summary is provided in the context of the identified KPIs related to the eHealth use case (as stated in D1.1).

Code	KPI	Enabler involved	Status
TO1.1	**Real Time Traffic Management Plan updates on On-Board Systems.** Demonstrate the remote and continuous deployment of, at least, two cabins with OTI and a Plan update during the operation part. (i) Including at least 2 gateways and 2 IoT devices (ii) Including at least 1 cloud resource (iii) Including at least 2 deployments (iv) At least 1 software component is updated	Orchestration and Continuous Deployment Enabler	Completed in a simulated environment. At this stage the tool and the Use Cases can deal with this indicator in a virtual machine environment.
TO1.2	**SW development for the infrastructure deployed.** Agile Software deployment on the CMWs Demonstrate a valid deployment with lack of human interaction in a reduced amount of time (limited by the device) increasing the efficiency in operation time and workload. Demonstrate the remote deployment of the CMWs: (i)Including at least 2 gateways and IoT devices, (ii) Including at least 1 cloud resource	Orchestration and Continuous Deployment Enabler	Completed in a simulated environment. At this stage the tool and the Use Cases can deal with this indicator in a virtual machine environment.
TO1.2	**Elements simulation** Simulation of at least 6 different devices	Test, Emulation and Simulation Enabler	This KPIs not the focus for this period.
TO4.1	**Demonstrate the integration of the ITS SIS with the FiWARE (Orion Context Broker)**	Orchestration and Continuous Deployment Enabler	Completed. The Use Case infrastructure is connected to FIWARE
TO2.1	Demonstrate the improvement of the monitoring thanks to contextual and effectiveness analysis at operational time.	Context Monitoring and Actuation conflict management enabler	Was not the focus for this period.
TO2.2	Identify and Solve actuations conflicts at design time.	Actuation Conflict Management Enabler	Was not the focus for this period.
TO3.1	**Attacks detection and isolation** Demonstrate the detection and diagnoses different suspicious behaviours at the cloud and edge level respectively Isolate cybersecurity attacks for non-safety applications	Security and Privacy Monitoring and Control Enabler	Was not the focus for this period.
TO3.2	**Security not variable** Simulate attacks and demonstrate the level of security keeps during runtime – i.e., protection against attacks works	Security and Privacy Monitoring and Control Enabler	Was not the focus for this period.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

TO3.3	**Attacks historical** Demonstrate access to an historical log of attacks in S&P Monitoring Enabler	Security and Privacy Monitoring and Control Enabler	Was not the focus for this period.
TO3.4	**Orchestration Interface** Demonstrate that the Monitoring enabler is able to send alerts to the Orchestration and deployment enabler.	Security and Privacy Monitoring and Control Enabler	Was not the focus for this period.
TO3.5	**Failure Report** Demonstrate that the enabler is able to analyse network data and provide information about, at least, one system source of failure.	Fault-Tolerant and distributed Root-cause analysis and Prediction Mechanisms of IoT Applications	At this stage, the connection with the RCA is established

Figure 1: ITS – ENACT Tools KPIs. Source: INDRA

The former KPIs defined are matched with the following tests.

3.1.1 Evaluation and Validation DevOps Scenarios

In order to evaluate and validate the defined KPIs, it was identified a set of DevOps scenarios related to the ITS use case and these were summarized into a set of tests. These tests are intended to validate various aspects encountered by the DevOps scenarios. The DevOps scenarios and the related set of tests was described in D1.1.

The following table summarizes the status of these tests that was designed for the ITS use case.

Group of tests	Test	Description	ENACT tool	Test Update	Status
Things Data Monitoring	Test 1.1.0.1 Monitoring On-Track data	Monitoring test	Context Monitoring and Actuation Conflicts Management Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Software update/Actuation	Test 1.2.0.1 Wagon data update	Test for the wagon's identification data update	Orchestration and Continuous Deployment Enabler	The business information is separated from the ENACT tool influence to minimize the specialization per Use Case	Accomplished. The tool is able to receive business data from the ITS infrastructure to trigger a deployment. The tool is able to deploy a version of the ITS GW software
	Test 1.2.0.2 Orders prioritization.	Test to set the actuation over the system tasks prioritization	Actuation Conflict Management	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.
	Test 1.2.0.3 Train stopped.	Test to realize the movement state of the train	Orchestration and Continuous Deployment Enabler	No updates accomplished	Accomplished. The tool is able to receive business data from the ITS infrastructure to trigger a deployment. The tool is able to deploy a version of the ITS GW software
Simulation and Testing	Test 1.3.0.1 Single device simulation testing.	Simulation of a single device test	Test, Emulation and Simulation Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.
	Test 1.3.0.2 Interaction between devices simulation testing.	Simulation of the interactions between the devices	Test, Emulation and Simulation Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

	Test 1.3.0.3 Scaling procedure.	Simulation of the system scalability and problems that will be faced	Test, Emulation and Simulation Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.
Root Cause Analysis	Test 1.4.0.1 Environmental distinction.	Test to distinguish the source of a communication issue	Fault-Tolerant and distributed Root-cause analysis and Prediction Mechanisms of IoT Applications	No updates accomplished	At this stage, the connection with the RCA is established, the tool is able to receive ITS messaging. However, no functionalities are tested. This task was tested with the tool and the ITS infrastructure remotely.
Security Monitoring	Test 1.5.0.1 Penetration testing.	Test of threads	Security, Privacy and Trustworthiness Monitoring Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.

Figure 2: ITS – ENACT Tools Validation and Verification Test Progress. Source: INDRA

From the former table, it can be inferred that 40% of the progress related to the tools integration is accomplished. However, during the next iterations new tests will be added to validate and verify the new functionalities that will be added and the remaining ones defined into the ENACT first period.

3.1.2 Evaluation and Validation Application Scenarios

Regarding the specific tests for the ITS use case. The adaptations accomplished in the use case are shown and the testing status shown:

Test	Description	Status
Inauguration Process	Validation to set the wagons that form the composition	Accomplished. Validated by all the ITS participant partners (INDRA, EDI, and BOSC)

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Integrity Process	Train Integrity process with a fixed composition	Accomplished. Validated by all the ITS participant partners (INDRA, EDI, and BOSC)
Hard Reset	Train Inauguration reset process	Accomplished. Validated by all the ITS participant partners (INDRA, EDI, and BOSC)
Logistic and Maintenance Process	Logistic and maintenance report attached to the composition presented	Accomplished. Validated by all the ITS participant partners (INDRA, EDI, and BOSC)
Metadata report	Traffic metadata to evaluate the system status	Not accomplished. Use Case Functionality not implemented into the ENACT first period.

Figure 3: ITS Validation and Verification Test Progress. Source: INDRA

These tests show that the functional data required by the ENACT tools are present. However, the system metadata is not yet available as it can be adapted to the pending to integrate tools.

3.1.3 Conclusions

As it can be observed, the KPIs and the tests to Validate and Verify the developments done during the ENACT first period for the ITS Use Case (D1.2 [2]) are matched with the Technical Objectives.

The progress can be considered adequate as:

- The main integration issues are solved showing that the connection between the use case and the tools is feasible. Two ENACT tools are already integrated, with not all the functionalities available.
- GeneSIS is able to make deployments on a Safety and Secured rail infrastructure managed by the ITS use case. This represents a turning point, as it is the tool that orchestrate other ENACT tools to be used into the use case.
- FIWARE as a platform is integrated and able to store the use case data and represent it to the user.

As the initial barrier to integrate the system is already solved and the Know-How to accomplish this task is obtained, the other planned ENACT tools integration can be tackled with not further problems and the development efforts can be focus on improving the ENACT tools themselves into the ITS use case. As it was introduced, new tests will be designed as soon as the development efforts are applied improving the KPIs completeness status.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

3.2 Digital Health

Developing and operating the eHealth use case the following main needs are elaborated and provided as input to the development of the ENACT enablers:

- Abstraction of low-level embedded code (IoT edge)
- Provisioning and deployment of the gateway software
- Automate the management of a fleet of gateways
- Authentication of gateways and context aware access
- Continuous risk monitoring
- Management of security and privacy risks

The Evaluation and Validation of ENACT results in the first period of the project applying the eHealth use case has included:

- Evaluation and validation of the use case itself as a reference for the ENACT enablers both in terms of requirements and in terms of applying some state of the art DevOps tools as baseline and integration points for ENACT enablers. This includes the main needs listed above and the use case functional scenarios presented in D1.1. This part and the use case scenarios are summarised in Section 3.2.2
- First evaluation and validation of a set of enablers or part of them. These has all naturally been based on the first versions of these enablers. The maturity of these enablers varies significantly, thus, the level of evaluation also varies. For some it has been more an evaluation of the concept, while for others the evaluation is based on actual application of enablers in the use case.

An overall summary of the evaluation and validation performed applying the eHealth use case is shown in the table below. The summary is provided in the context of the identified KPIs related to the eHealth use case (as stated in D1.1).

Code	KPI	Enabler involved	Unit of measurement	Who	Status
TO1.1	**Continuous deployment across the IoT, edge and Cloud space.** (i) Include at least 2 IoT&Edge nodes and 2 cloud nodes. (ii) 10 deployments (iii) Includes orchestration, setting up interoperation with "no downtime"	Orchestration and Continuous Deployment Enabler	# of IoT, edge and cloud nodes # of deployments "Zero" downtime	Tellu, SINTEF	First evaluation performed that includes **1 IoT&edge node** (the personal health gateway (PHG)) and several cloud nodes. Evaluation based on ENACT development and baseline technology stack specified by the ENACT Enabler (e.g., Ansible, Jenkins, GitHub, Docker, Kubernetes). In particular ThingML is applied for all code development of the PHG, both for the code handling application functionality and code for managing DevOps (e.g., GW agent, monitoring probes etc). **More than 10 simultaneous deployments** are performed. It includes orchestration and setting up interoperation with **no "down time"**. Final evaluation will be performed in the last phase of the project.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

TO1.2	**Automatic change or upgrade.** Demonstrate 5 changes or upgrades of 5 independent Gateways. Performed automatically and without need of physical intervention (or minimize physical intervention)	Orchestration and Continuous Deployment Enabler	# of changes, # of Gateways # of physical interventions	Tellu, SINTEF	First evaluation performed, of **5 independent Gateways** with **several changes**, performed automatically and **without need of physical intervention**. Applying ThingML and baseline technology stack of the Orchestration and continuous deployment Enabler.
TO1.3	**Automatic Pairing of devices with Gateway after reset.** At least 5 different devices automatically paired with Gateway after reset	Orchestration and Continuous Deployment Enabler	# of different devices	Tellu, SINTEF	First evaluation performed with **3 different devices** automatically paired with Gateway after reset. Applying ThingML and baseline technology stack of the Orchestration and continuous deployment Enabler
TO1.4	**Simulation of at least 5 different devices/things**	Test, Emulation and Simulation Enabler	# of supported devices	Tellu, MI	Initial experiments with node red flows to simulate GWs
TO1.5	**Combination of simulated and physical devices** Demonstrate a combined test environment that apply at least 5 different simulated devices and at least 3 different physical devices	Test, Emulation and Simulation Enabler	# of devices	TellU, MI	Will be based on the above, remain to combine simulated devices with physical devices
TO2.1	**Root cause analysis** Support 5 archetypical situations of root cause analysis and the provision of meaningful information for those.	Run-time Quality Assurance and Root Cause Analysis Enabler	# of archetypcial root cause analysis situations	TellU, MI	Evaluation and validation not started
TO3.1	**Recovery after failures** Demonstrate the successful recovery after injected failures of two gateways	Robustness & Resilience Enabler	Y/N #Time to recovery	TellU, SINTEF	Evaluation and validation not started
TO3.2	**Configuration Rollback** Demonstrate configuration Rollback	Robustness & Resilience Enabler	Y/N	TellU, SINTEF	Initial experiments conducted. More systematic approach will be evaluated in the last phase of the project
TO3.3	**Diverse deployments** Demonstrate the automatic deployment and maintenance of at least 5 different deployments among the customers	Robustness & Resilience Enabler	# different deployments	TellU, SINTEF	Initial experiments based on the concept of fleet management. Next phase of the project will focus on evaluation of better automation as part of the fleet management concept
TO3.3	**Detection of suspicious behaviours.** Detection and Diagnoses of at least 5 different suspicious behaviours at the cloud and edge level respectively (10 in total)	Security and Privacy Monitoring and Control Enabler	# of suspicious behaviours detected	Tellu, Tecnalia	Initial experiments on general monitoring of Network traffic, memory, temperature and CPU to detect if something is abnormal.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

TO3.4	**Automate risk management** Demonstrate automatic risk management by integrating ENACT risk management functionalities into mainstream DevOps software for seamless tracking and monitoring of risks related to security, privacy and trustworthiness	Risk-Driven Decision Support Enabler	Demonstrate Automation	Tellu, BEAWRE	Initial experiments with the Risk decision support enabler applying BEAWRE framework.
TO3.5	Real-time monitoring of a set of Medical Gateways and reception of proper notifications with useful information in case of errors. This includes: (i) Demonstrate the detection (ii) Usefulness of the notifications	Security and Privacy Monitoring and Control Enabler	Y/N # End-user Satisfaction	TellU, Tecnalia	Initial experiments started based on ENACT principles and applying state of the art tools as baseline (e.g., Grafana, Prometheus)
TO3.6	**Protection of person sensitive data** Demonstrate protection against privacy attacks and incidents	CAAC, S&P Control Enabler	Y/N	TellU, EVIDIAN, Tecnalia	Not started
TO3.7	**Access control** Demonstrate the control of access to the gateway to authorised users only Demonstrate the control of access to the gateway to authorised services only	CAAC, S&P Control Enabler	Y/N Y/N	TellU, EVIDIAN, Tecnalia	Performed first evaluation of the Context Aware Access Control (CAAC), in particular the device authentication flow
TO3.8	**Secure data transmission** Demonstrate the enforcement of use of Secure protocols	S&P Monitoring Enabler	Y/N	TellU, Tecnalia	Initial experiments conducted
TO3.9	**Trustworthy communications in the sense of reliability, availability, integrity and privacy** Demonstrate the availability of enforcement controls	S&P Monitoring Enabler	Y/N	TellU, Tecnalia	Evaluation not started, some concepts are tested out
TO3.10	**Monitoring, Diagnose information and failure detection** Detection and Diagnosis of suspicious behaviours at cloud level	S&P Monitoring Enabler	Y/N	TellU, Tecnalia	Not started
TO3.11	**Full end-to-end security**	Secure protocols, S&P Monitoring Enabler, S&P Control Enabler, CBA	It is the collection of the KPIs above.	TellU, Tecnalia	Initial experiments conducted.

Figure 4: Digital Health – ENACT Tools KPIs. Source: TELLU

3.2.1 Evaluation and Validation DevOps Scenarios

In addition to evaluate and validate according to defined KPIs, it was identified a set of DevOps scenarios related to the eHealth use case and these were summarized into a set of tests. These tests are

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

intended to validate the various the aspects encountered by the DevOps scenarios. The DevOps scenarios and the related set of tests was described in D1.1.

The following table summarizes the status of these tests that was designed for the Digital Health Use Case.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Group of tests	Test	Description	ENACT tool	Test Progress at M22+Status
Initial Software deployment and GW onboarding	Test 2.1.0.1 Development/Specification of Generic image Test 2.1.0.2 Initial deployment of generic image Test 2.1.0.3 GW onboarding	Test of ENACT development/specification toolchain, initial deployment and GW onboarding,	Orchestration and Continuous Deployment Enabler	Test of ENACT development/specification toolchain, initial deployment and GW onboarding, applying ThingML and baseline technology stack of the enabler (Ansible, GitHub, Jenkins, Kubernetes, Docker). Next validation will include more components and features of the ENACT GeneSIS enabler
Software deployment and evolution in the Gateway	Test 2.2.0.1 Software deployment and evolution in the Gateway	Test of software deployment and software evolution across edge and cloud (GW and cloud server).	Orchestration and Continuous Deployment enabler & Security and Privacy Monitoring and Control enabler	Test of software deployment and software evolution across edge and cloud (GW and cloud server). Applying ThingML and baseline technology stack of the enabler (Ansible, GitHub, Jenkins, Kubernetes, Docker). Next validation will include more components and features of the ENACT GeneSIS enabler
Gateway recovery and factory reset in the field	Test 2.3.0.1 Gateway recovery and factory reset in the field	Tests the capability of remote recovery and reset functionality of potentially huge number of edge components accounting for both their physical and virtual nature.	Orchestration and Continuous Deployment enabler & Security and Privacy Monitoring and Control enabler	Basic tests of the capability of remote recovery and reset functionality of edge components accounting for both their physical and virtual nature.
Software testing on the Gateway	Test 2.4.0.1 Software testing on the Gateway	Test of ENACT capability of performing efficient testing across IoT, edge and cloud space	Test, Emulation and Simulation enabler & Risk-Driven Decision Support enabler	Test will be conducted in next phase of the project

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Continuous integration of gateway modules and versioning	Test 2.5.0.1 Continuous integration of gateway modules and versioning	Test of DevOps continuous integration capabilities and versioning across IoT, edge and cloud	Test, Emulation and Simulation enabler, Robustness & Resilience enabler & Orchestration and Continuous Deployment enabler	Test will be conducted in next phase of the project
Trustworhiness of the medical system	Test 2.6.0.1 Trustworhiness of the medical system	Test of ENACT capabilities of efficient support for ensuring proper security and privacy of the full DevOps process according to customer requirements as well as regulations and laws	Security and Privacy Monitoring and Control enabler, Risk-Driven Decision Support enabler Robustness & Resilience enabler	Test of ENACT Context Aware Access Control as well as the risk management support conducted

Figure 5: Digital Health – ENACT Tools Validation and Verification Test Progress. Source: TELLU

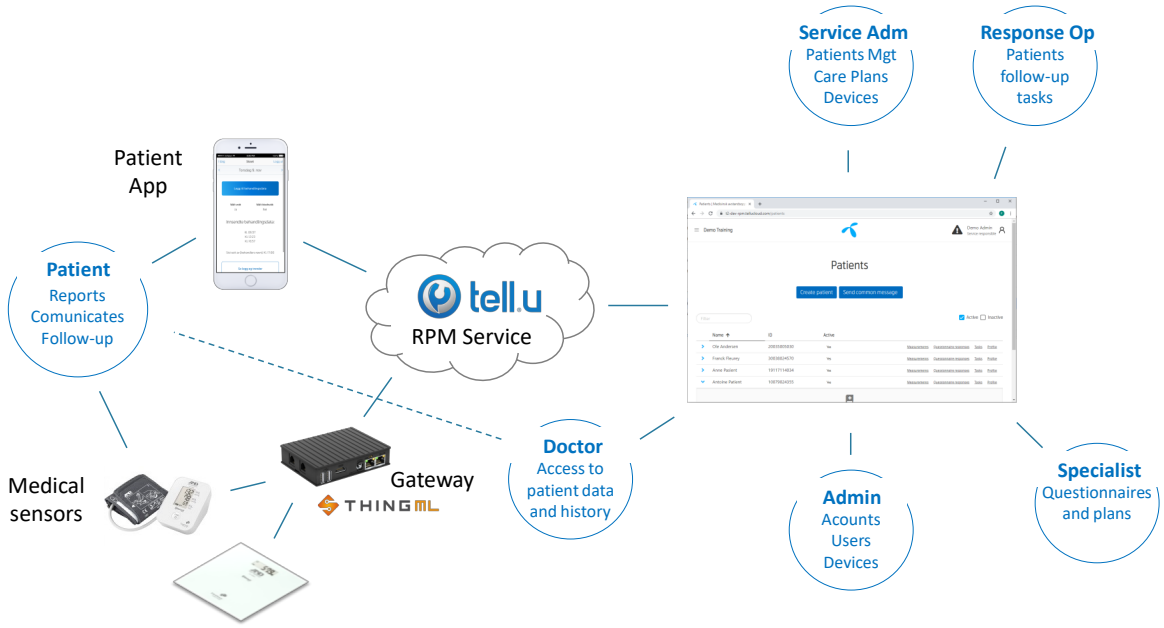
During the next iterations it may be added new tests to validate and verify more functionalities of ENACT enablers. Details of the tests and scenarios are provided in the appendix

3.2.2 Evaluation and Validation Application Scenarios

A main effort in the first period of the project has also been the evaluation and validation of the use case itself, to ensure it can function as a reference for the ENACT enablers both in terms of requirements and in terms of applying some state of the art DevOps tools as baseline and integration points for ENACT enablers.

The overall set up of the eHealth use case is depicted in the figure below. **This set up is now complete and 100% functional.** It includes the Personal Health Gateway, which is one central edge computing node that can manage a set of medical devices (IoT layer). Each patient is typically provided with a package of devices and a configuration set up that match the patient individual needs. The patient app is provided in the patient mobile phone and/or tablet, these are other edge computing nodes that need to be present. The RPM service is a SaaS running in the cloud. A cloud-based web application is provided to serve the Service administrator and response operator roles configuring the system and perform patient follow up tasks respectively. These set up is meant to be able to scale to thousands of homes.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).



3.2.3 Conclusions

The above shows how the KPIs and the test to Validate and Verify the developments done during the ENACT first period for the eHealth Use Case are matched by Technical Objectives and describe the progress against the final objectives specified for the Use Case. Main achievements in terms of application and validation of ENACT enablers during the first period includes:

- The integration, application and first evaluation of the Context Aware Access Control is performed having tight and agile interaction with the Enabler development team. The CAAC is an innovation that is very relevant for Tellu in order to do authorisation management at the Gateway level
- The ENACT deployment and orchestration base tooling is integrated, applied for the software deployment on the edge and IoT level and evaluated
- ThingML is extensively applied for the development of the Personal Health Gateway and evaluated throughout the use case.
- The Risk Management enabler is applied and evaluated both in terms of handling risks related to security and privacy. Continuous feedback has been provided in order to ensure direction of the development according to Use case requirements.

The plan for the next period is to follow up on already integrated and applied Enablers as well as applying and evaluating a few additional Enablers, in particular, the Security and Privacy Enabler, the Run-time Quality Assurance and Root Cause Analysis Enabler.

3.3 Smart Building

In the Smart Building use case, several ENACT tools were successfully tested to enable continuous deployment, solve of actuation conflicts and ensure security and privacy of the communications. In addition, the integration and adaptation of the remaining tools to identify risks at design time, correct

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

behavioural drifts, enforce security and privacy and self-optimizing controller that will be tested during the final period of the project evolve as planned. The completeness status of the evaluation and validation of ENACT tools in the use case for this period is near the 50% of the whole project.

The Evaluation and Validation of ENACT results in the first period of the project applying the Smart Building use case has included:

- Evaluation and validation of the use case itself as a reference for the ENACT tools and a set of the first versions of those tools in a series of DevOps scenarios. The tested DevOps scenarios focused on the concurrent operation of multiple energy efficiency and user comfort IoT applications with orchestration of components, actuation conflicts, trustworthiness, security and privacy risks.
- First evaluation and validation of the first versions of a set of enablers. During this period, three ENACT tools were tested: the Orchestration and Continuous Deployment Enabler (GeneSIS), the Actuation Conflict Manager (ACM) Enabler and the Security and Privacy Monitoring Enabler.

An overall summary of the evaluation and validation performed applying the Smart Building use case is shown in the table below. The summary is provided in the context of the identified KPIs related to the Smart Building use case (as stated in D1.1).

Code	KPI	Enabler involved	Status
TO1.1	**Interface with Risk-Management Enabler.** The Orchestration enabler interacts with the Risk-Management Enabler to provide it with the list of devices selected as part of the SIS.	Risk-Management Enabler	Not measured as it was not the focus of this period
TO4.1	**Integration with SOFIA/SMOOL.** Demonstrate the integration of the Orchestration and deployment enabler with the SMOOL platform: (I) Demonstrate the continuous deployment of SMOOL client and their automatic integration with SMOOL broker. (II) Demonstrate data exchange of the deployed components via SMOOL.	Orchestration and Continuous and Deployment Enabler	Completed. The SMOOL middleware was integrated via the Orchestration and deployment enabler for continuous deployment and data exchange
TO1.2	**Deployment of the use case applications** Demonstrate the continuous deployment of the two smart building applications. Including actuation conflict managers and S&P monitoring probes.	Orchestration and Continuous and Deployment Enabler	Completed. Energy efficiency and user comfort IoT applications were continuously deployed including actuation conflict managers and security and privacy monitoring
TO2.1	**Direct conflicts when acting on shared actuators.** The Conflicts Enabler identifies and avoid conflicts in commands send to shared actuators by one or two IoT Systems or applications.	Context Monitoring and Actuation Conflict Management Enabler	Completed. The Actuation Conflict Manager (ACM) Enabler was implemented for thermal comfort

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

			control when two different IoT applications sent different temperature setpoints to the same heating actuator
TO2.2	**Indirect conflicts when acting on shared physical entities.** The Conflicts Enabler identifies and avoid conflicts when acting on shared physical entities, e.g. lighting, by two or more IoT Systems even if actuators are not directly shared.	Context Monitoring and Actuation Conflict Management Enabler	Partially completed. A demo was implemented with two different IoT applications that indirectly affected to lighting physical entity. However, the Behavioural Drift Analysis (BDA) Enabler is not yet available to evaluate how much the observed behaviours of the IoT System are different from the expected ones
TO2.3	**Online learning and user comfort and energy consumption** Demonstrate a positive impact on the user comfort and energy consumption by comparing how the system evolve with or without the online learning	Context-Aware Self-Adaptation Enabler	Not measured as it was not the focus of this period
TO2.4	**Online Learning and Actuation conflict management** Demonstrate integration between the online learning tool and the actuation conflict management enabler	Context-Aware Self-Adaptation Enabler	Not measured as it was not the focus of this period
TO3.1	**Risk level.** Risk Enabler indicates a risk level in a particular configuration of the IoT system	Risk-Management Enabler	Not measured as it was not the focus of this period
TO3.2	**Risk level analysis combining legacy systems.** Risk Enabler to analyze possible threats brought by new system parts and its combination with legacy systems.	Risk-Management Enabler	Not measured as it was not the focus of this period
TO3.3	**Risk Management to minimize risks.** Risk-Management Enabler support selection of the best IoT system elements that minimize risks according to the user requirements.	Risk-Management Enabler	Not measured as it was not the focus of this period

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

TO3.4	**Risk level change alert.** DSS Enabler raises alerts whenever a change in the IoT system (e.g., a redeployment) modifies the risk level.	Risk-Management Enabler	Not measured as it was not the focus of this period
TO3.5	**Demonstrate configurability of alarm threshold.** Demonstrate that the S&P Monitoring enabler enables the user to set the desired thresholds to raise cybersecurity alarms.	Security & Privacy Monitoring Enabler	Completed. The Security and Privacy Monitoring Enabler was used to raise cybersecurity alarms when messages were sent by unauthorized nodes and abnormal network traffic was detected
TO3.6	**Security enforcement. Demonstrate reaction to attacks and incidents.** Demonstrate that the S&P Monitoring enabler work together with S&P Control Enabler which helps reacting to attacks or incidents.	Security & Privacy Control Enabler	Not measured as it was not the focus of this period

Figure 6: Smart Building – ENACT Tools KPIs. Source: TECNALIA

The former KPIs are matched with the following tests.

3.3.1 Evaluation and Validation DevOps Scenarios

In addition to evaluate and validate according to defined KPIs, it was identified a set of DevOps scenarios related to the Smart Building use case and these were summarized into a set of tests. These tests are intended to validate the various the aspects encountered by the DevOps scenarios. The DevOps scenarios and the related set of tests were described in D1.1.

The following table summarizes the status of these tests that were designed for the Smart Building use case.

Group of tests	Test	Description	ENACT tool	Test Update	Status
Thermal comfort control – Heating design tests	Test 3.1.0.1 Risk-driven heating design	Test risk during the design of thermal comfort application design	Risk-Driven decision support Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

	Test 3.1.0.2 ThingML models integration in the SMOOL middleware	Test to verify if ThingML models integrated in SMOOL are automatically deployed	Orchestration and Continuous Deployment Enabler	The Orchestration and Continuous Deployment Enabler was added to the heating design tests	Accomplished. ThingML models were correctly deployed with GeneSIS due to their integration in SMOOL middleware
Thermal comfort control – Conflict in heating actuator use test	Test 3.2.0.1 Actuation conflicts on the use of the same actuator	Test of the capability of the ACM tool to resolve the sending of conflicting orders by different IoT flows to the same heating actuator	Actuation Conflict Manager (ACM) Enabler	Enabler No updates accomplished	Accomplished. The ACM Enabler rewrote the IoT flows for an adequate behaviour
	Test 3.2.0.2 Integration of the Actuation Conflict Manager Enabler with GeneSIS	Test to verify if the ACM tool is correctly deployed	Orchestration and Continuous Deployment Enabler	The Orchestration and Continuous Deployment Enabler was added to the conflict in heating actuator use test	Accomplished. The ACM tool was correctly deployed with GeneSIS
Thermal comfort control – Conflict in temperature actuation tests	Test 3.3.0.1 Actuation conflicts on the same physical variable	Test actuation conflict on the same control parameter	Behavioural Drift Analysis (BDA) Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.
	Test 3.3.0.2 Integration of the Behavioural Drift Analysis Enabler with GeneSIS	Test to verify if the BDA tool was correctly deployed	Orchestration and Continuous Deployment Enabler	The Orchestration and Continuous Deployment Enabler was added to the conflict in temperature actuation tests	Accomplished. The BDA tool was correctly deployed with GeneSIS

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Smart building alerts for user comfort tests	Test 3.4.0.1 Trustworthy smart building alerts	Test of the capability of the Security and Privacy Monitoring tool to check if the messages were maliciously modified or sent by unauthorized nodes	Security and Privacy Monitoring Enabler	EnablerNo updates accomplished	Accomplished. The Security and Privacy Monitoring Enabler detected abnormal network traffic and ensured the integrity of the data
	Test 3.4.0.2 Improved security in smart building alerts	Test to implement proactive security measures	Security and Privacy Control Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.
Thermal comfort control – Self-optimizing controller design	Test 3.5.0.1 Self-optimizing controller	Self-optimization test of HVAC operation	Online Learning Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.
	Test 3.5.0.2 Integration of the Online Learning Enabler with GeneSIS	Test to verify if the Online Learning tool was correctly deployed	Orchestration and Continuous Deployment Enabler	No updates accomplished	Not accomplished. Tool not integrated into the ENACT first period.

Figure 7: Smart Building – ENACT Tools Validation and Verification Test Progress. Source: TECNALIA

As it can be seen in the above table, approximately half of the ENACT tools have been tested in the first version of the implementation of the use cases. New tests will be performed during the second period of the project to validate the new versions of the ENACT tools in the Smart Building domain.

3.3.2 Evaluation and Validation Application Scenarios

In the first period of the project, the Energy Efficient Building application was implemented as benchmark for ENACT tools and is now 100% operational. The Energy Efficient Building application performs thermal comfort control of the building using environmental sensors, motorized blinds, electric heaters and HVAC actuators. This application will allow to test the following tools developed in the ENACT project: Orchestration and Continuous Deployment Enabler, Actuation Conflict Manager (ACM) Enabler, Behavioural Drift Analysis (BDA) Enabler, Online Learning Enabler, Risk-Driven Decision Support Enabler, Security and Privacy Monitoring Enabler, Security and Privacy Control Enabler.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

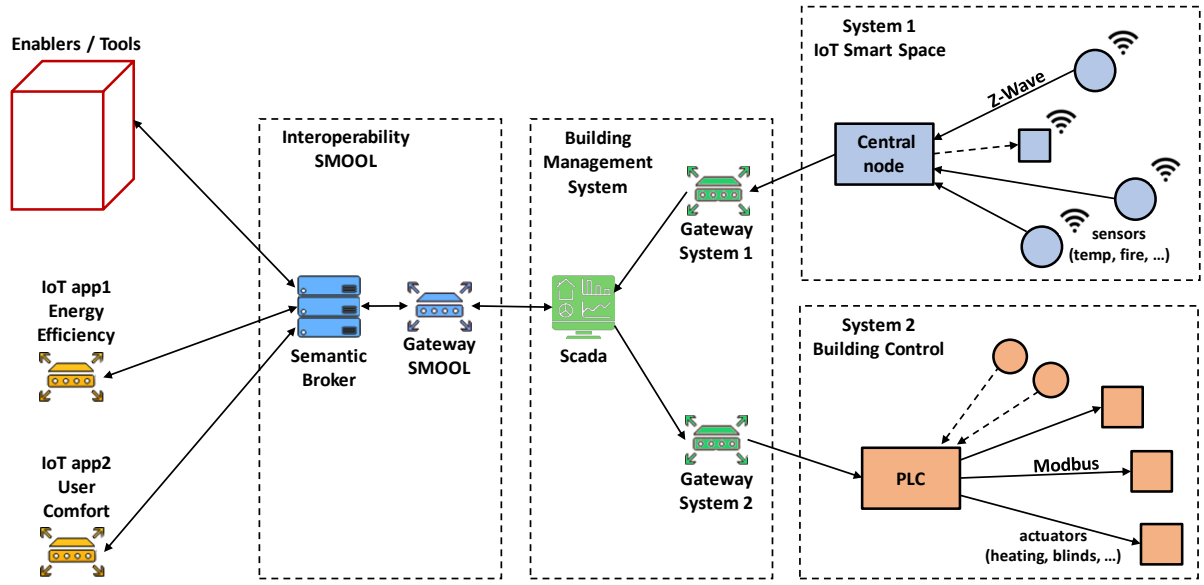


Figure 8: Smart Building – Schema of the Energy Efficient Building application. Source: TECNALIA

3.3.3 Conclusions

The KPIs measured during the ENACT first period for the Smart Building use case correspond with the ENACT tools that were integrated in the use case for the first period of the project:

- The Orchestration and Continuous and Deployment Enabler (GeneSIS) is also used by most of the ENACT tools such as he Actuation Conflict Manager (ACM) Enabler and is able to make deployments on of IoT applications in the Smart Building environment.
- SMOOL platform (SOFIA) is fully integrated with ThingML models and is used as communication middleware in the Smart Building use case.
- The Actuation Conflict Manager (ACM) Enabler is fully operational and resolve the sending of conflicting orders by different applications to the same actuator.
- The Security and Privacy Monitoring Enabler actively checks messages and monitors network traffic.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

4 Conclusion

The Evaluation and Validation of ENACT results in the first period of the project applied to the different use cases has included the validation of the ENACT tools over the Use Cases and the adaptations made into the Use Case themselves.

The Use Cases Evaluation and Validation themselves as a reference for the ENACT enablers, both in terms of requirements and in terms of applying some state-of-the-art DevOps tools as baseline and integration points, enabler have been completed at 50% approximately:

- ITS Use Case: All the tests have been completed but the Metadata report that will be finish on the third period as a source for the Security Monitoring and Control enabler.
- Digital Health: Most of the DevOps scenarios associated tests have been completed (4/6) but Software testing on the Gateway and Continuous integration of gateway modules and versioning that will be done on the third period.
- Smart Building: The testing of the most fundamental ENACT tools for the use case have been performed with successful results. In the second period, the pending tools for the use case will be fully implemented and tested.

Concerning the First evaluation and validation of a set of enablers or part of them, these has all naturally been based on the first versions of these enablers. The maturity of these enablers varies significantly, thus, the level of evaluation also varies. We can summarize grouping by enablers that:

- **Orchestration and Continuous Deployment enabler**: It has been completed for ITS, Digital Health and Smart Building: Real Time Traffic Management Plan updates on On-Board Systems.
- **Test, Emulation and Simulation enabler**: some simulation of different devices and a combination of simulated and physical devices test have been made for Digital Heath Use Case but it is not completed yet.
- **Context Monitoring and Actuation conflict management enabler**: It is almost completed from Smart Building domain being able to identify and to avoid the direct and indirect conflicts when acting on shared actuators and physical entities.
- **Context-Aware Self-Adaptation enabler**: not started during this first period
- **Security and Privacy Monitoring and Control enabler**: This enabler is not started yet in the ITS domain. However, in Digital Health Use Case some initial experiments have been made and the enabler is able to monitor in real time and at the same time create alarms and detect suspicious behaviours. In Smart Building first period tests are fully completed. It is demonstrated the configurability of alarm threshold.
- **Run-time Quality Assurance and Root Cause Analysis enabler:** The Root Cause analysis evaluation and validation test is has not being started from Digital Health Use Case.
- **Robustness & Resilience enabler**: Evaluation of this enabler has been started in the Digital Health Use Case. Initial experiments have been done regarding the configuration rollback and the automation of several deployments in different devices.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Risk-Driven Decision Support enabler**: This enabler is planned to be implemented in the Digital Health and Smart Building. Initial experiments have been tackled in the Digital Health Use Case.
- **CAAC and S&P Control enablers**: These enablers have been experimented both in the Smart Building and the Digital Health Use Cases. Several initial experiments are done related with the authentication flow, secure transmission, and full end security besides the risk level report tests.
  .

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

5 References

[1] D1.1 Use case definition and requirements, validation and evaluation
[2] D1.2 Case studies implementation
[3] The Goal/Question/Metric Method: a practical guide for quality improvement of software Development - THE McGRAW-HILL COMPANIES - Rini van Solingen and Egon Berghout

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

A Evaluation and Validation Plan

A.1 ITS Domain (Rail)

A.1.1.1 Use Case – ENACT Tools Tests

A.1.1.1.1 Software update/Actuation test

ENACT references

Enabler reference: Orchestration and Continuous Deployment Enabler.

Enabler tool: Orchestration and deployment

Test 1.2.0.1 Wagon data update

- **Enablers involved**

Enabler
Orchestration and Continuous Deployment Enabler
Context Monitoring and Actuation Conflict Management Enabler

- **Description**

The wagons require having a specific identification to be part of a predefined composition. This identification is needed for the Train Integrity and the rest of the procedures are only accomplished once the Train Integrity is correct.

It is requested to update the rolling stock composition data in several trains at the same time. Comparing the data file that contains the data composition and the identifications written on the rolling stock, the Things validate that the parameters are updated and, as a consequence, the software update process is correct.

- **Interfaces of the test**
  - **The Running System**: The system to be update.
  - **Orchestration & Deployment engine**: Manage the software updates deployed in the system.
  - **Orchestration, Deployment, Security**: Generate the software updates based on the monitored inputs.
  - **Monitoring & Analytics Platform**: Storage of the business and health information.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Testing Inputs**
  - **Infrastructure Health Information**: Information obtained from the status of the deployed devices. This functionality is based on the Context Monitoring and Actuation Conflict Management Enabler.
  - **Business information**: Information related with the applications of the Use Case: Train Integrity, Wagon Composition Discovery, Logistics, and former SW updates/configurations. This functionality is based on the Context Monitoring and Actuation Conflict Management Enabler.
- **Testing Outputs**
  - A notification reported back and stored at the Cloud to be visualized by the Rail operator that contains:
    - The updated information.
    - The updated Things.
    - Log Status.
    - The information path until reaching the Things.
- **Test Procedures**
  - **Initialization**
    - Simulated input data is queued for the test.
    - Inputs are treated in the Orchestration and Deployment tool in order to generate the file and the address of the Things.
    - The Orchestration and Deployment tool checks the status of the Things to check the feasibility of the update.
  - **Development**
    - The Orchestration & Deployment engine tool manage the transmission of the data to the GWs after the Rail Operator order the SW update/configuration.
    - The SW is updated on the Things and the report file is generated and sent.
  - **Termination**
    - The file is evaluated on the monitoring platform.
    - The file is compared with the inputs to validate and verify the test.
- **Testing Constraints**
  - There are not further constraints than those ones that involve the personnel, except the time limitation for the SW updated or configuration. The operation time is considered low.
  - The personnel activates the test 1.2.0.3 Train stopped and the necessary resources to accomplish the test are available.
- **Control**
  - The test is considered semi-automatic. The Rail Operator starts the SW update/configuration operations. Moreover, the same entity checks the Things reports to validate and verify the test during the update process.
  - The process is tracked in the results report and during the test by the GWs infrastructure. The inputs are stored to compare these data with the update data to validate and verify the test.
- **Test Relationships**
  - The tests that interact with the Test 1.2.0.1 Wagon data update are:
    - Test 1.1.0.1 Monitoring On-Track data
    - Test 1.2.0.3 Train stopped

Test 1.2.0.2 Orders prioritization.

- **Enablers involved**

Enabler
Orchestration and Continuous Deployment Enabler
Context Monitoring and Actuation Conflict Management Enabler

- **Description**

Several orders can be applied and they may have a non-logical overlapping in the infrastructure. It is expected form the enabler to schedule and to establish a prioritization of these orders.

Several orders may be applied to the rolling stock communications infrastructure during the Train Discovery process, only the necessary ones will be applied. The orders, which are not applicable in the Train Discovery procedure, are scheduled to be applied after the composition confirmation phase.

It is expected in every case a proper order of the instructions provided.

- **Interfaces of the test**
  - The Running System: The system to be interacted.
  - Monitoring & Analytics Platform: Monitors the results of the deployment.
  - Orchestration & Deployment engine: Manage the order to be deployed on the rolling stock communications infrastructure.
  - Orchestration, Deployment, Security: Edits the file with the order to be deployed.
- **Testing Inputs**
  - Rail Operator instructions: The instructions that are executed on the Rail Infrastructure.
  - Rail Operator confirmation: The actions executed by the Rail Operator.
- **Testing Outputs**
  - Actuation of the instruction executed.
  - A results report.
- **Test Procedures**
  - **Initialization**
    - Several orders to the same affected devices are queued to execute.
    - The Rail Operator orders the execution of the instructions.
  - **Development**
    - The orders are prioritized. They are also evaluated to avoid overlapping in the actuation.
  - **Termination**
    - The actions are finished. The Rail Operator visualizes the results reported.
- **Testing Constraints**
  - No further constrains found at this stage.
- **Control**
  - The operation is considered manual. The Rail operators has to authorise manually the operations.
- **Test Relationships**
  - The tests that interact with the Test 1.2.0.2 Orders prioritization are:
    - Test 1.1.0.1 Monitoring On-Track data
    - Test 1.2.0.3 Train stopped

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Test 1.2.0.3 Train stopped.

- **Enablers involved**

Enabler
Orchestration and Continuous Deployment Enabler
Context Monitoring and Actuation Conflict Management Enabler

- **Description**

The execution of SW updates/configurations is accomplished in wagons that are stopped. This is justified due to safety reasons to avoid potential accidents.
It is required to validate that no orders are executed in a running rolling stock. The Integrity Things report the location, and the velocity state, of the rolling stock.

- **Interfaces of the test**
  - The Running System: The system to be interacted with.
  - Orchestration & Deployment engine: Manage the signal to request the rolling stock movement status.
  - Orchestration, Deployment, Security: Edits the request file to request the rolling stock movement status.
  - Monitoring & Analytics Platform: Storage the rolling stock respond regarding movement status.
- **Testing Inputs**
  - Movement status request: The Rail Operator requests the movement status of the rolling stock.
- **Testing Outputs**
  - Reception of the movement status of the rolling stock.
- **Test Procedures**
  - **Initialization**
    - A software deployment or actuation is prepared to be deployed in the rolling stock. Therefore, the rolling stock must be stopped and a movement status is requested.
    - The movement status request is queued.
  - **Development**
    - The Orchestration & Deployment engine tool manages the transmission of the data to the On-Board GW.
    - The accelerometers, integrated in the Train Integrity Things, report the information on the speed variations.
  - **Termination**
    - The Rail Operator receives the accelerometers information to verify that the train is stopped.
    - The report is compared with the initially known movement state of the rolling stock.
  - **Testing Constraints**
    - No further constrains found at this stage.
- **Control**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- The operation is considered automatic.

- **Test Relationships**
  - The tests that interact of the Test 1.2.0.3 Train stopped are:
    - Test 1.1.0.1 Monitoring On-Track data
    - Test 1.2.0.1 Wagon data update
    - Test 1.2.0.2 Orders prioritization

Verification Plan

The verification is accomplished using the internal tests designed for the Software updated/Actuation scenario 4 and the Test 1.1.0.1 Monitoring On-Track data. The representation is shown in the next figure:

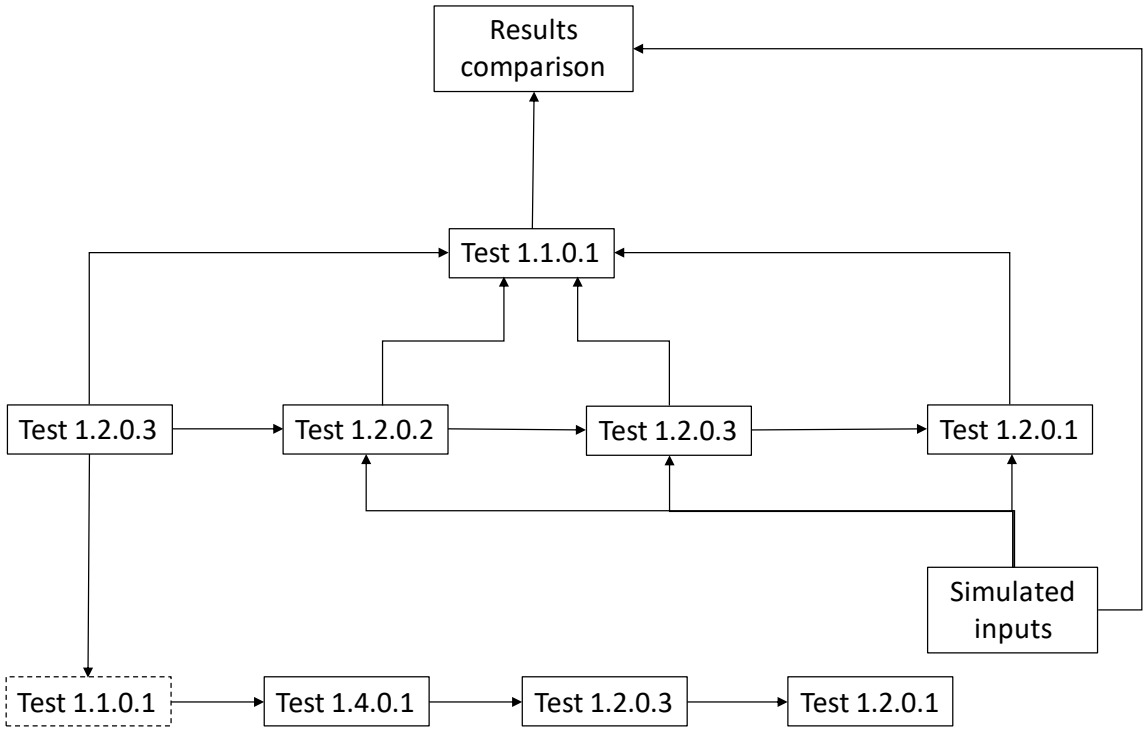


Figure 9: SW Update/configuration Verification Scheme. Source: INDRA.

The verification is accomplished at the depot following the steps:

- **Step 1:** It is tested that the train is stopped. This process is repeated before every actuation is done on the rolling stock, but it is not applied to the fixed infrastructure (information panels).
- **Step 2:** Two Rail Operations entities provide different commands to update the data of the same wagon. The Tests 1.2.0.2 Prioritization Orders throws a voting to establish the priority.
- **Step 3:** The planed delays from the Test 1.4.0.1 Arrivals planification are deployed. It is checked that the train is stopped for further actions.
- **Step 4:** The data is updated and the results reported to be checked with the simulated inputs.

A.1.1.1.2 Root Cause Analysis tests

ENACT references
Enabler reference: Run-time Quality Assurance and Root Cause Analysis Enabler.
Enabler tool: Root Cause Analysis.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Test 1.4.0.1 Environmental distinction.

- **Enablers involved**

Enabler
Context Monitoring and Actuation Conflict Management Enabler

- **Description:**

The environmental conditions for the rolling stock communication infrastructure is considered harsh. These conditions may cause degradation in the performance of the Things and/or their communications. It is expected to test the RCA to validate the capacity to distinguish failures caused by environmental issues from the ones caused by the Thing issues. The objective is to avoid false alarms in critical applications such as Train Integrity.

- **Interfaces of the test**
  - **The Running System:** The system to be interacted. Report the information to test the RCA in rail domains.
  - **Monitoring & Analytics Platform:** Storage of the results.
- **Testing Inputs**
  - **Infrastructure Health Information:** Information obtained from the status of the deployed devices. This functionality is based on the Context Monitoring and Actuation Conflict Management Enabler.
  - **Business information:** Information related with the applications of the Use Case: Train Integrity, Wagon Composition Discovery, Logistics, and former SW updates/configurations. This functionality is based on the Context Monitoring and Actuation Conflict Management Enabler.
  - **Rail models:** The environmental models are provided to establish a comparison with the results.
- **Testing Outputs**
  - A report of the errors distinguishing the detection error against the measured simulated inputs.
- **Test Procedures**
  - **Initialization**
    - Detection of a failure.
    - Inputs of the business and health information are simulated. The inputs are provided by the monitoring platform.
  - **Development**
    - The monitoring platform, based on the models, tests the simulated inputs received.
  - **Termination**
    - The results are compared against the inputs to observe the error rate, detecting the environmental influences.
- **Testing Constraints**
  - No further constrains were found at this stage.
- **Control**
  - This process is considered automatic.
- **Test Relationships**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- The relationships are limited to the Things Data Monitoring DevOps scenario 1. The related tests are listed**:**
  - Test 1.1.0.1 Monitoring On-Track data

Verification Plan

The verification plan will involve the interactions with the monitoring and the orchestration Enablers and the tests themselves. The general procedure will be tackled following the scheme:

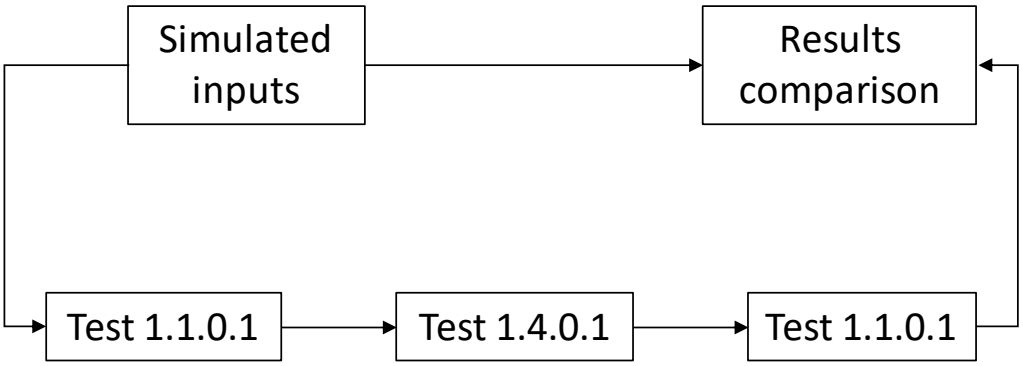


Figure 10: RCA Verification Scheme. Source: INDRA.

The steps performed in the Context-Aware Self-Adaptation are:

- **Step 1:** The Test 1.1.0.1 reports the information necessary for the RCA.
- **Step 2:** The Test 1.4.0.1 is performed to detect simulated environmental failures. The results are reported back to the monitoring platform to be stored for further operations.
- **Step 3:** The detection error rate is calculated comparing the inputs and the detections.

A.1.1.1.3 Security Monitoring tests

ENACT references
Enabler reference: Security and Privacy Monitoring and Control Enabler.
Enabler tool: Security & Privacy Monitoring.

Test 1.5.0.1 Penetration testing.

- **Enablers involved**

Enabler
Security and Privacy Monitoring and Control Enabler

- **Description:**

The security threats that can affect a rail scenario are various and the consequences are severe. It is expected from the system to develop capabilities to detect these attacks and generate attacks historical data. These attacks historical data must be improved from new attacks detected in the infrastructure and in other systems. This information could also come from third parties as CISC.
The system will be tested simulating On-Track and On-Board attacks and the system must be able to monitor the threats and report them.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Interfaces of the test**
  - **The Running System:** The system to interact.
  - **Monitoring & Analytics Platform:** Storage of the results.

Testing Inputs

- **Infrastructure Health Information:** Information obtained from the status of the deployed devices. This functionality is based on the Context Monitoring and Actuation Conflict Management Enabler. However, administrative information such as logins will be reported.
- **Business information:** Information related with the applications of the Use Case: Train Integrity, Wagon Composition Discovery, Logistics, and former SW updates/configurations. This functionality is based on the Context Monitoring and Actuation Conflict Management Enabler. However, administrative information such as logins will be reported.

Testing Outputs

- A report with the attacks detection rate.
- Alarm in case of attack.

Test Procedures

Initialization

-Simulated On-Board and On-Track attacks are started.

Development

-The Monitoring platform treats the detection of the attacks

Termination

-A report with the detected attacks is compared with the inputs simulated. The objective is obtaining the attacks detection rate**.**

Testing Constraints

No further constrains were found at this stage.

Control

This process is considered automatic.

Test Relationships

The relationships involve the attacks detection and the learning capabilities.

Verification Plan

The verification plan will involve the interactions with the security monitoring and the learning capabilities. The general procedure will be tackled following the scheme:

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

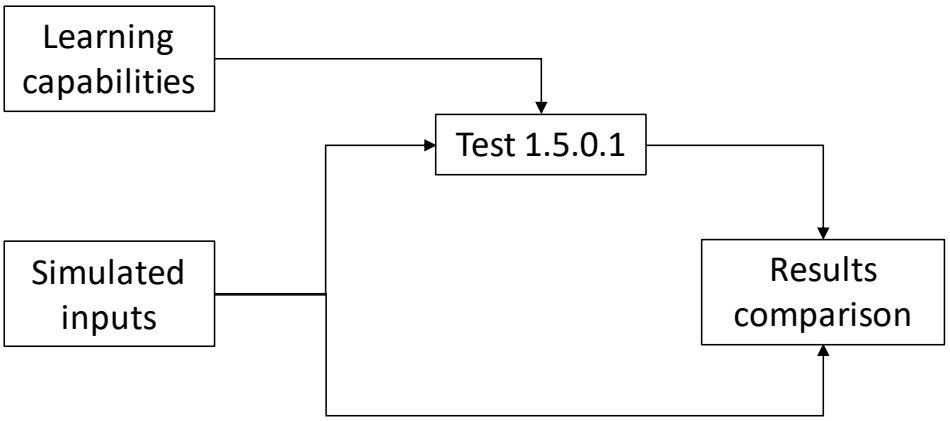


Figure 11: Security Verification Scheme. Source: INDRA.

The steps performed in the Security Monitoring are:

- **Step 1:** The stored attacks are incorporated to the attacks validation test.
- **Step 2:** The Test 1.5.0.1 validates the attack detection and the historical is updates in case of positive detection.
- **Step 3:** The results are compared.

A.1.1.2 Use Case Tests

A.1.1.2.1 Inauguration Process

Description

The objective of this test is perform the integrity process in order to know if the composition of the train is composed by wagons defined by the inauguration process

Interfaces of the test

- Interface CMW to WSN coordinator
- Interface CMW to DMI

Testing Inputs

This section show the inputs to perform the test, the inputs are listed.

- The driver sends to the CMW a train integrity asynchronous request.

Testing Outputs

This section show the outputs obtained from the test, the inputs are listed.

- The integrity process has been successfully carried out.

Test Procedures

Initialization

The initial conditions of the test are listed.

- Communication Middleware (CMW)

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- CMW device on

- WSN coordinator
  - Sensors on

Development

The procedure in order to perform the tests are:

- Step 1: The J101100 is send to the WSN Coordinator
- Step 2: Once the J101100 is received, the WSN Coordinator send the train consist data through the J101103. The process is performed into a 5 seconds timer.

Termination

The tests end with next termination conditions.

- The driver sends an asynchronous request in order to end the mission.

Testing Constraints

N/A

Control

In this section the automatic condition of the test is detailed

Test Relationships

Related tests:

- Test 1.2. Integrity Process

A.1.1.2.2 Integrity Process

Description

The objective of this test is perform the integrity process in order to know if the composition of the train is composed by wagons defined by the inauguration process.

Interfaces of the test

- Interface CMW to WSN coordinator
- Interface CMW to DMI

Testing Inputs

This section show the inputs to perform the test, the inputs are listed.

- CMW sends a request to start the train integrity.
- The driver sends to the CMW a train integrity asynchronous request.

Testing Outputs

This section show the outputs obtained from the test, the inputs are listed.

- The integrity process has been successfully carried out.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Test Procedures

Initialization

The initial conditions of the test are listed.

- Communication Middleware (CMW)
  - CMW device on
- WSN coordinator
  - Sensors on

Development

1. **Case 1**

The procedure in order to perform the tests are:

- Step 1: The CMW send the J100100 in order to request to the WSN coordinator the train integrity data.
- Step 2: The WSN coordinator send the J100106 each 0.25 seconds.
- Step 3: The CMW send to each WSN coordinator the J100101 indicating the stop of publishing train integrity data.
- Step 4: Each WSN coordinator send the J100102 to confirm the stop gathering information from the nodes.

2. **Case 2**

The procedure in order to perform the tests are:

- Step 1: The CMW send the J100100 in order to request to the WSN coordinator the train integrity data.
- Step 2: The WSN coordinator send the J100106 each 0.25 seconds, the packet is reported by an intruder.
- Step 3: The packet cannot be published into the system and the Train Integrity is broken.

3. **Case 3**

The procedure in order to perform the tests are:

- Step 1: The CMW send the J100100 in order to request to the WSN coordinator the train integrity data.
- Step 2: The WSN coordinator send the J100106 each 0.25 seconds, but a node is missed.
- Step 3: The Train Integrity is broken and the train stopped. It is connected again and the Train Integrity recovered.
- Step 4: The CMW send to each WSN coordinator the J100101 indicating the stop of publishing train integrity data.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- Step 5: Each WSN coordinator send the J100102 to confirm the stop gathering information from the nodes.

Termination

The tests end with next termination conditions.

- The driver receives a notification of the end the mission.

Testing Constraints

N/A

Control

In this section the automatic condition of the test is detailed

Test Relationships

Related tests:

- Test 1.1. Inauguration Process

A.1.1.2.3 Hard Reset

Description

The objective of this test is re-establish the default values of the WSN coordinator in order to correct unexpected failures with not feasible solution.

Interfaces of the test

- Interface CMW to WSN coordinator
- Interface CMW to DMI

Testing Inputs

This section show the inputs to perform the test, the inputs are listed.

- Request from the driver side to start the reset process

Testing Outputs

This section show the outputs obtained from the test, the inputs are listed.

- Execute the reset operation

Test Procedures

Initialization

The initial conditions of the test are listed.

- Communication Middleware (CMW)
  - CMW device on
- WSN coordinator
  - Sensors on

Development

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

The procedure in order to perform the tests are:

- Step 1: The driver requests the reset process. Therefore, the CMW send the J103101 to the WSN coordinator.
- Step 2: The WSN coordinator send the 103102 to the CMW.
- Step 3: The WSN send a J103102, which contains the report about the reset process.

Termination

The tests end with next termination conditions.

- The driver receives the notification of the sensors has been correctly reset.

Testing Constraints

N/A

Control

In this section the automatic condition of the test is detailed

Test Relationships

Related tests:

- Test 1.1. Inauguration Process
- Test 1.2. Integrity Process
- Test 1.3. Maintenance Process

A.1.1.2.4 Logistic and Maintenance Process

Description

The objective of this test is verified that the maintenance data gathered is correctly send to the CMW and transmit to the Cloud.

Interfaces of the test

- Interface CMW to WSN coordinator
- Interface CMW to Cloud

Testing Inputs

This section show the inputs to perform the test, the inputs are listed.

- The driver throws an order to request the maintenance information form the WSN coordinator.

Testing Outputs

This section show the outputs obtained from the test, the inputs are listed.

- The maintenance information has been successfully gathered.

Test Procedures

Initialization

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

The initial conditions of the test are listed.

- Communication Middleware (CMW)
  - CMW device on
- WSN coordinator
  - Sensors on
- Cloud

Development

4. **Case 1:**

The procedure in order to perform the tests are:

- Step 1: The CMW send the J102100 to start the gathering data of the WSN for the Maintenance.
- Step 2: The WSN coordinator send the J102119 and J102120 to the CMW.
- Step 3: The CMW receives the maintenance data and transmit it to the Cloud.
- Step 4: The CMW throws an order to stop collecting maintenance data to the WSN coordinator (J102102)
- Step 5: The WSN coordinator answers with a J102104 to confirm about to stop the Maintenance data transmission.

5. **Case 2:**

- Step 1: The CMW send the J102100 to start the gathering data of the WSN for the Maintenance.
- Step 2: The WSN coordinator send the J102119 and J102120 to the CMW.
- Step 3: The CMW receives the maintenance data and transmit it to the Cloud from a different coordinator not identified in the LDAP server.
- Step 4: The data is not represented at the Grafana,
- Step 5: The CMW throws an order to stop collecting maintenance data to the WSN coordinator (J102102).

Termination

The tests end with next termination conditions.

- The driver sends an asynchronous request in order to end the mission.

Testing Constraints

N/A

Control

In this section the automatic condition of the test is detailed

Test Relationships

Related tests:

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- Test 1.1. Inauguration Process

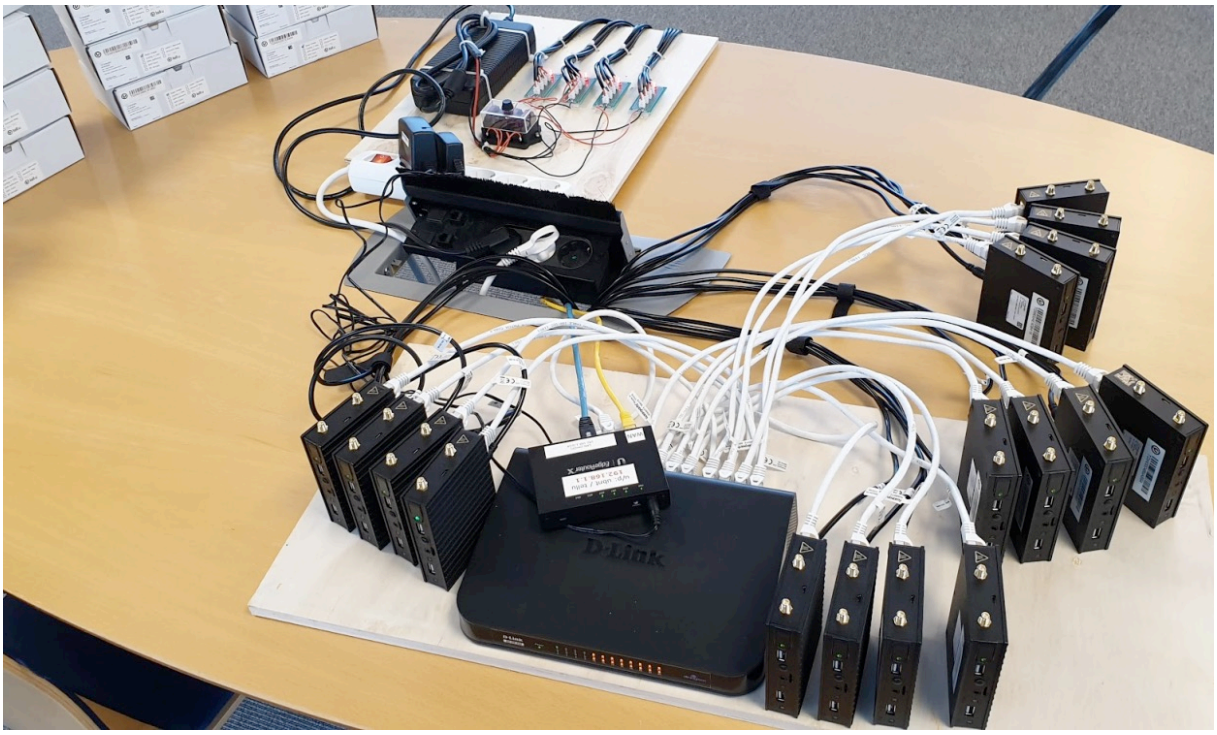
A.2 Digital Health

Enabler tool: Orchestration, deployment and security, ThingML, Security & Privacy Monitoring, Orchestration and deployment engine, Test and Simulation Environment.

A.2.1.1 Initial Software deployment and GW On-Boarding tests

Test 2.1.0.1 Development/Specification of Generic image (for initial deployment)

Status: This test has been conducted during the first evaluation and validation based on the currently available tools of the Orchestration and Continuous Deployment Enabler. In particular we have used the ThingML editor of the Enabler for specification and we have set up a testenvironment for providing initial deployment of generic image on the GW, which typically is performed before the GW's are shipped to the customer. The baseline software stack of the enabler is applied for the initial deployment (e.g., ansible). The picture below shows the testsetup for the initial deployment of up to 16 GWs in parallelle



- **Enablers involved**

Enabler

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Orchestration and Continuous Deployment Enabler

- **Description**

  The developer of the generic image for medical gateway want to get efficient support for specification of GW image and deliver specification models that can be verified and checked before deployment (for example according to certification standards). Moreover, the developer wants efficient support for specification of component orchestration and deployment. ENACT tool will be used for specification and derivation of code to be deployed on the medical GW as well as specification of the actual configuration of the image, the deployment and security set up. Moreover, it will be used to ensure provision of models that can be verified and checked.

- **Interfaces of the Enabler Scenario**
  - Orhcestration and Continuous Deployment Enabler - ThingML editor has been used for the specifications.
- **Testing Inputs**
  - The actual code level generic image for the medical gateway as well as configuration files and scripts for deployment.
- **Testing Outputs**
  - A ThingML specification that derive the compliant (to the manual code) code as well as specifications for orchestration, deployments and security. Moreover, a set of models that can be validated and checked.
- **Test Procedures**
  - **Initialization**
    - Code of the generic image for the GW as well as configuration files and scripts for deployment
  - **Development**
    - Specification of the compliant model in the ENACT tools and the derivation of code, as well as orchestration, deployment and security configuration files and scripts. Provision of models that can be validated and checked
  - **Termination**
    - The derived code, configurations and scripts are compliant with the manual developed code, configurations and scripts.
- **Testing Constraints**

  No further constrains found at this stage.

- **Control**

  The operation is partly manual (specification) and partly automatic (derivation).

- **Test Relationships**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

None.

Verification Plan

- **Key Performance Indicator (KPI)** (Orchestration and Continuous Deployment Enabler):

KPI	Evaluation
Efficiency in specification	We found ThingML efficient for doing specification. It has a faniliar high level programming language like syntax and is based on standard UML finite state machines principles. Thus, it is easy to learn, and the specification is easy to grasp. The way the low level details are managed by the built in abstractions makes it efficient to work with. We believe it will be important to have smooth integrations of the editors of the Orchestration and Continuous Deployment Enabler (e.g., the ThingML and the Node Red based editors) for increase usability. There are missing some libraries for interfacing with more IoT technologies and frameworks (e.g., MQTT, REST etc)
Derived code	It is appreciated that the ThingML derived code is readable, in our experience many code generation tools generates code that is hard to read. The code derived from ThingML can easily be further evolved in a standard IDE. We will investigate further the quality of the derived code in the next phase of the project, especially with respect to efficiency and memory management (this is particularly important when targeting C and C++ code.
configurations and scripts are correct	For the test we have conducted so far, all the configurations and scripts have been correct.

Test 2.1.0.2 Initial deployment of generic image (in case the GW is remote)

Status: At this stage we are not able to do initial deployment on the GW remotely. This is partly due to lack of automation and partly because of lack of tools. We expect that ENACT will contribute to make this possible.

- **Enablers involved**

Enabler
Orchestration and Continuous Deployment Enabler

- **Description**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

In some cases the Medical GW will be shipped with the generic image deployed. In some cases the GW may already been shipped and the generic image need to be remotely deployed. The developer wants to execute the specified deployment in the remote case.

- **Interfaces of the Enabler Scenario**
    - Orchestration and deployment engine
- **Testing Inputs**
    - The derived code as well as the orchestration, deployment and security scripts.
- **Testing Outputs**
    - Successful deployment of a generic image on a remote GW
- **Test Procedures**
    - **Initialization**
        - GW that are remote
    - **Development**
        - Execution of the orchestration and deployment
    - **Termination**
        - The generic image is successfully deployed on a remote GW
- **Testing Constraints**

  No further constrains found at this stage.
- **Control**

  The operation is automatic.
- **Test Relationships**

  None.

Verification Plan

- **KPI** (Orchestration and Continuous Deployment Enabler): The generic image is successfully deployed on a remote GW.

Test 2.1.0.3 GW onboarding

Status: Initial tests has been conducted applying both ThingML and baseline DevOps tooling and technologies such as GitHub, Ansible, Jenkins, Docker, Kubernetes and SSH.

- **Enablers involved**

Enabler
Orchestration and Continuous Deployment Enabler

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Security and Privacy Monitoring and Control Enabler

- **Description**

  The GW should identify to a server with its Gateway ID that the server provides. If the Gateway ID is OK/Confirmed the GW is expected to retrieve credentials and certificates, to allow secure connection. GW identifies to a server through a secure connection and retrieves specific configuration and SW components for the particular GW. It is expected to get support to monitor the execution of the components and reports to DevOps backend.

- **Interfaces of the Enabler Scenario**
  - Orchestration and deployment engine
  - Security and Privacy Monitoring
- **Testing Inputs**
  - GW with generic image deployed and ready to start GW on-boarding.
- **Testing Outputs**
  - GW is On-Boarded and executes or operator has been notified on exceptions
- **Test Procedures**
  - **Initialization**
    - Trigger to start GW on-boarding of a GW with generic image deployed
  - **Development**
    - GW identifies to a server with its Gateway ID that the server provides
    - If the Gateway ID is OK/Confirmed the GW retrieve credentials and certificates, to connect to Virtual Private Network (VPN) backend.
    - GW identifies to a server in the VPN and retrieves configuration and SW components.
    - DevOps agent monitors the execution of the components and reports to DevOps backend.
  - **Termination**
    - The GW is securely On-Boarded
- **Testing Constraints**

  No further constrains found at this stage.

- **Control**

  The operation is automatic.

- **Test Relationships**

  None.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Verification Plan

- **KPI** (Orchestration and Continuous Deployment Enabler, and Security and Privacy Monitoring and Control Enabler):

KPI	**Evaluation**
The whole process is secure	For the initial test the security concerns are handled applying the baseline software stack. In the next phase of the project larger part of the Orchestration and Continuous Deployment Enabler and Security and Privacy Monitoring and Control Enabler will be tested to make the process more efficient and improve to the security handling. At this stage it includes some manual tasks, that we want to be able to automate and perform on remote GWs. For example, to distribute the original software image and the certificates used to connect to the management backend.
Continuous deployment support	The initial test has been conducted applying ThingML and the baseline software stack. In the next phase of the project larger part of the Orchestration and Continuous Deployment Enabler will be applied to make the process more efficient and to be able to scale to large fleets of GWs
Framework should provide meaningful information to the operator to diagnose the status	N/A (Not part of the initial test conducted in the first part of the project)

A.2.1.2 Software deployment and evolution in the Gateway

ENACT references

Enabler reference: Orchestration and Continuous Deployment Enabler, Security and Privacy Monitoring and Control Enabler.

Enabler tool: Security & Privacy Monitoring, Orchestration and deployment engine,

Test 2.2.0.1 Software deployment and evolution in the Gateway

Status: Initial tests has been conducted applying both ThingML and baseline DevOps tooling and technologies such as GitHub, Ansible, Jenkins, Docker, Kubernetes and SSH.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Enablers involved**

Enabler
Orchestration and Continuous Deployment Enabler
Security and Privacy Monitoring and Control Enabler

- **Description**

The Developer and operator want to enable continuous Software deployment and evolution in the Gateway. This includes change of any of the services running on the GW or and upgrade of an existing component.

- **Interfaces of the Enabler Scenario**
    - Orchestration and deployment engine
    - Security and Privacy Monitoring
- **Testing Inputs**
    - Software updates and specification of orchestration and deployment
- **Testing Outputs**
    - GW is updated or operator has been notified of any exceptions
- **Test Procedures**
    - **Initialization**
        - Initiate a trigger to start an update. I.e., a change of any of the services running on the GW or and upgrade of an existing component
    - **Development**
        - GW downloads new versions of the service components from the Binary Repository (continuous process)
        - Operator triggers the change or update of a component (which is locally available) (Coming from the Developer providing the software updates/Change)
        - GW stops components to be updated and deploy the new configuration (only redeploy the components that are changed)
        - GW starts the new configuration
        - Backend check that the new configuration is running as specified by the operator
    - **Termination**
        - The GW is securely updated
- **Testing Constraints**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

No further constrains found at this stage.

- **Control**

  The operation is automatic.

- **Test Relationships**

  It is a relationship to the Test 2.1.0.1, as a similar process is needed to specify the software code as well as the orchestration, deployment and security set up.

Verification Plan

- **KPI** (Orchestration and Continuous Deployment Enabler, and Security and Privacy Monitoring and Control Enabler):

KPI	**Evaluation**
The whole process is secure	For the initial test there security concerns are handled applying the baseline software stack. In the next phase of the project larger part of the Orchestration and Continuous Deployment Enabler and Security and Privacy Monitoring and Control Enabler will be tested to make the process more efficient and improve to the security handling
Continuous deployment support	The initial test has been conducted applying ThingML and the baseline software stack. In the next phase of the project larger part of the Orchestration and Continuous Deployment Enabler will be applied to make the process more efficient and to be able to scale to large fleets of GWs
Framework should provide meaningful information to the operator to diagnose the status	N/A (Not part of the initial test conducted in the first part of the project)

A.2.1.3 Gateway recovery and factory reset in the field

ENACT references

Enabler reference: Robustness & Resilience Enabler, Orchestration and Continuous Deployment Enabler, Security and Privacy Monitoring and Control Enabler.

Enabler tool: Robustness&Resilience, Security & Privacy Monitoring, Orchestration and deployment engine.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Test 2.3.0.1 Gateway recovery and factory reset in the field

Status: This test will be performed the next phase of the project.

- **Enablers involved**

Enabler
Robustness&Resilience enabler
Orchestration and Continuous Deployment Enabler
Security and Privacy Monitoring and Control Enabler

- **Description**

  The operator wants to do a gateway recovery and factory reset in the field, typically in situations where the GW is reassigned to another user or is malfunctioning and cannot be fixed by normal redeployment.

- **Interfaces of the Enabler Scenario**
  - Robustness and Resilience
  - Orchestration and deployment engine
  - Security and Privacy Monitoring
- **Testing Inputs**
  - Remote Gateway with existing software deployed
- **Testing Outputs**
  - GW is recovered and On-Boarded
- **Test Procedures**
  - **Initialization**
    - Trigger to start the factory reset
  - **Development**
    - The image of the GW is restored to its Factory image
    - On-Boarding procedure is followed as it was a new GW
  - **Termination**
    - The GW is securely recovered and On-Boarded

  No further constrains found at this stage.

- **Control**

  The operation is automatic.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Test Relationships**

  This test includes the Test 2.1.0.3.

Verification Plan

- **KPI** (Resilience and robustness, Orchestration and Continuous Deployment Enabler, and Security and Privacy Monitoring and Control Enabler):
  - Ability to trigger remotely low level reset function (without relying on the upper level software stack and DevOps framework to be operational

A.2.1.4 Ensuring Software quality on the Gateway

ENACT references

Enabler reference: Test, Emulation and Simulation Enabler, Risk-Driven Decision Support Enabler

Enabler tool: Test & Simulation Environment, Risk Driven development process.

Test 2.4.0.1 Ensuring Software quality on the Gateway

Status: Initial testing of the Risk Driven Decision support Enabler is performed

- **Enablers involved**

Enabler
Test, Emulation and Simulation Enabler
Risk-Driven Decision Support Enabler

- **Description**

  Developer has made a new version of the Software that needs to be risk mitigated and properly tested before it is made available in the Production Repository.

- **Interfaces of the Enabler Scenario**
  - Test, Emulation and Simulation Enabler,
  - Risk Driven DevOps process
- **Testing Inputs**
  - GW software that are not yet risk mitigated and tested
- **Testing Outputs**
  - GW software is validated through risk analysis and realistic testing (realistic implies typically some actual devices/sensors and proper simulation and/or emulation of IoT devices.)
- **Test Procedures**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Initialization**
  - Trigger to conduct risk analysis and start the test
- **Development**
  - Risk analysis and mitigations are conducted
  - Code is deployed on a test infrastructure that consists of both actual devices/sensors and simulated devices/sensors (either by mocks or by using emulators)
  - Application specific test suite is executed.
  - Not all tests can be automated and manual tests have to be run. For example for Blood pressure measurement one would like to pair the device and measure sending/transport, (hard to have a "patient" in the loop in automatic and continuous testing)
  - Generic Monitoring probes are used to verify that the components are running as expected.
  - Results of the tests are provided to the Developer
  - Relevant risks are monitored.
- **Termination**
  - The test is executed and test results are reported

- **Testing Constraints**

  No further constrains found at this stage.

- **Control**

  The operation is automatic.

- **Test Relationships**

  No relationships

Verification Plan

- **KPI** (Test, Emulation and Simulation Enabler, Risk-Driven Decision Support Enabler):

KPI	**Evaluation**
A Risk DevOps process is applied to include improved coverage and testing of critical functions and qualities	For the initial test the architecture of the use case has been modelled in the tool and an initial risk analysis is performed. We analysed the risk for each communication channel and for each data path. In particular, we want to look for impact related to information being tampered, or information being injected after a given GW have been compromised.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Test results are provided in a way that is easy exploitable by the developer (e.g., highlighting regressions and giving easy access to trace of the failing tests)	N/A (Not part of the initial test conducted in the first part of the project)
Need to be able to run and rerun an arbitrary selection of tests.	N/A (Not part of the initial test conducted in the first part of the project)
Efficiency for example in terms of methods and guidelines to some extent be able to create mocks for physical/manual processes (like device pairing)	N/A (Not part of the initial test conducted in the first part of the project)

A.2.1.5 Continuous integration of gateway modules and versioning

ENACT references

Enabler reference: Test, Emulation and Simulation Enabler, Robustness & Resilience Enabler, Orchestration and Continuous Deployment Enabler

Enabler tool: Test & Simulation Environment, Orchestration and deployment engine, Robustness and Resilience.

Test 2.5.0.1 Continuous integration of gateway modules and versioning

Status: Initial tests has been performed using a combination of build scripts and GitHub release tags. There are still a few manual steps that we would like to automate with contribution form ENACT

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Enablers involved**

Enabler
Test, Emulation and Simulation Enabler
Robustness & Resilience Enabler
Orchestration and Continuous Deployment Enabler

- **Description**

  The developer wants to have continuous integration of gateway modules and versioning when he commits new code and/or tag a new release.

- **Interfaces of the Enabler Scenario**
  - Test & Simulation Environment,
  - Orchestration and deployment engine,
  - Robustness and Resilience.
- **Testing Inputs**
  - Code that is ready to be committed
- **Testing Outputs**
  - New software is integrated and new release is configured and validated
- **Test Procedures**
  - **Initialization**
    - trigger by committing new code and/or tag a new release
  - **Development**
    - The code is pulled by the continuous integration server
    - Modules are distributed to be built on appropriate executors (e.g. ARM gateways need code to be built on ARM servers)
    - Binary modules are pushed to distribution server, (similar to a Maven repository for Java). This should support versioning and distinguish between snapshots and releases.
    - Binaries are synchronised by test gateways
    - A test suite is executed
    - Validated binaries are made available for production gateways within the Distribution Repository

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

  - **Termination**
    - Binaries are made available, and they are only made available if they pass the integration tests

- **Testing Constraints**

  No further constrains found at this stage.

- **Control**

  The operation is automatic.

- **Test Relationships**

  No relationships.

Verification Plan

- **KPI** (Test, Emulation and Simulation Enabler, Robustness & Resilience Enabler, Orchestration and Continuous Deployment Enabler):

KPI	**Evaluation**
Manage the various configurations and versions of modules which have to be deployed together to make a working configuration (as they need to have compatible versions)	Initial solution is working OK using tags to indicate compatibility between components.
Test results are provided in a way that is easy exploitable by the developer (e.g., highlighting regressions and giving easy access to trace of the failing tests).	N/A (Not part of the initial test conducted in the first part of the project)
Efficiency, e.g., the process is fully automated (or at least	N/A (Not part of the initial test conducted in the first part of the project)

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

advancing the automation compared to current state of the art).	

A.2.1.6 Trustworthiness of the medical system

ENACT references

Enabler reference: Security and Privacy Monitoring and Control Enabler, Risk-Driven Decision Support Enabler, Robustness & Resilience Enabler

Enabler tool: Security and Privacy Monitoring and Control Enabler, Test & Simulation Environment, Risk driven development process, Robustness and Resilience.

Test 2.6.0.1 Trustworhiness of the medical system

Status: Initial implementation of device authentication flow using the Context Aware Access Control tool of the Security and privacy monitoring and Control enabler.

- **Enablers involved**

Enabler
Security and Privacy Monitoring and Control Enabler
Risk-Driven Decision Support Enabler
Robustness & Resilience Enabler

- **Description**

  The developer and operator want to have good support for ensuring proper security and privacy of the full DevOps process according to customer requirements as well as regulations and laws.

- **Interfaces of the Enabler Scenario**
  - Test & Simulation Environment,
  - Risk driven development process,
  - Robustness and Resilience.Test & Simulation Environment.
- **Testing Inputs**
  - DevOps framework, a set of operational GWs and trustworthiness test configurations
- **Testing Outputs**
  - Reports on the state of trustworthiness aspects

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Test Procedures**
  - **Initialization**
    - Execute specific trustworthiness test configuration
  - **Development**

    The Trustworthiness tests consists of applying a selection of test templates related to following aspects
    - Test the feasibility applying of risk analysis methods proposed by ENACT to identify threats in general related to trustworthiness and management of sensitive data in particular for the digital health domain.
    - Test security of all communication (e.g., that all communication with back end is encrypted)
    - Test security of Encryption keys (e.g., that they can not be shared between GWs and that each of them can be tied to specific GW IDs)
    - Test that it is not possible to set up connections directly between GWs
    - Test GW privacy management (e.g., that the GW is not storing or retrieving any identifiable patient data, and that GW are not storing log of sensitive data etc).
  - **Termination**
    - Trustworthiness test configuration(s) are executed and test reports are provided
- **Testing Constraints**

  No further constrains found at this stage.
- **Control**

  The operation is automatic.
- **Test Relationships**

  There are some relations with some of the other tests described related to IoT test infrastructure etc, still, Trustworthiness test-configurations can be executed independently.

Verification Plan

- **KPI** (Security and Privacy Monitoring and Control Enabler, Risk-Driven Decision Support Enabler, Robustness & Resilience Enabler):

• **KPI**	**Evaluation**
Efficient set up of Trustworthiness test configurations (e.g., based on templates)	Initial solution is working OK using tags to indicate compatibility between components.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

End to end security.	Device authentication flow is prerequisite to implement end to end encryption of the patient measurements. The initial test of the CAAC worked satisfactory. We need the GW to communicate the authentication code to the patient, we need to implement this in the next phase of the project.
Support to isolate GWs	Initial experiments with revokable Gateway specific certificates both at the management level (VPN) and at the application level (MQTT, RabbitMQ)
Efficient development and maintenance of a Risk Analysis for the trustworthiness of an IoT based system in a system lifecycle perspective.	N/A (Not part of the initial test conducted in the first part of the project)

A.3 Smart Building

Enabler tool: Orchestration and Continuous Deployment (GeneSIS), ThingML, SMOOL, Actuation Conflict Manager (ACM), Security and Privacy Monitoring.

A.3.1.1 Thermal comfort control – heating design tests

ENACT references

Enabler reference: Risk-Driven Decision Support Enabler.

Enabler tool: Risk-Driven Design Planning and Selection support.

Test 3.1.0.1 Risk-driven heating design

- **Enablers involved**

Enabler
Risk-Driven Decision Support Enabler
Orchestration and Continuous Deployment Enabler

- **Description**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

The developer of the Thermal control IoT application wants to use a smart thermostat to improve overall comfort and give more control to user. ENACT will be used to help understand cybersecurity risks in the IoT system and decide which countermeasures are needed and which security controls should be monitored at runtime by the Security and Privacy Monitoring and Control Enabler.

- **Interfaces of the Enabler Scenario**
    - **The thermal control energy efficiency IoT application**: The IoT application that has the thermal control strategy.
- **Testing Inputs**
    - **Business information**: Information related to the temperature values of the thermostat, energy consumption values of heating, related environmental sensorized values such as humidity and other contextual information, e.g. room location.
    - Critical assets to protect.
    - Criticality in data exchanged.
- **Testing Outputs**
    - The list of required security controls in the IoT application.
- **Test Procedures**
    - **Initialization**
        - IoT System architecture model (e.g. ThingML model).
    - **Development**
        - The thermal control system specification analysis with respect to potential risks (intentional and unintentional threats) is supported by the Risk Enabler.
    - **Termination**
        - The recommended countermeasures (security controls) to include in the IoT application so they can be tracked at runtime for ensuring secure behaviour.
- **Testing Constraints**

  No further constrains found at this stage.
- **Control**

  The operation is considered automatic.
- **Test Relationships**

  None.

Verification Plan

- KPI (Orchestration Enabler): Efficiency in deployment of IoT application.
- KPI (Risk Enabler): Understandability of threats identified.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- KPI (Risk Enabler): Understandability of recommendations for security controls required in the IoT application.

A.3.1.2 Thermal comfort control – conflict in heating actuator use test

ENACT references

Enabler reference: Context Monitoring and Actuation Conflict Management Enabler.

Enabler tool: Application and Context Monitoring and Actuation Conflicts Handling.

Test 3.2.0.1 Actuation conflicts on the use of the same actuator

- **Enablers involved**

Enabler
Actuation Conflict Manager (ACM) Enabler
Orchestration and Continuous Deployment Enabler

- **Description**

  The IoT energy efficiency application sends actuation orders to the HVAC system of the room according to sensorized temperature data. However, at some point the logic of the IoT application is sending colliding instructions to actuator(s) acting on the same physical variable.

- **Interfaces of the Enabler Scenario**
  - **Users' thermal comfort IoT application**: The IoT application that keeps the temperate according to users' preferences.
  - **The energy efficiency IoT application**: The IoT application that minimizes energy consumption.
- **Testing Inputs**
  - **Business information**: Information related to the temperature values of the thermostat, energy consumption values of heating, related environmental sensorized values such as humidity and other contextual information, e.g. room location.
  - The Finite State Machines (or other type of behavioural models) of both applications needs to be created.
- **Testing Outputs**
  - A detected conflict in the use of the same actuator and recommendation to avoid it in the design of the IoT applications.
- **Test Procedures**
  - **Initialization**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

        - The user comfort application and the thermal control energy efficiency IoT applications want to be deployed in the same IoT environment so they can co-exist. They may be sending contradictory orders on temperature status by acting simultaneously on the same actuator that controls the temperature. The designers need to analyse possible conflicts.
        - A unique model of desired behaviour in the smart environment is developed to be the input of the Context Monitoring and Actuation Conflict Management Enabler.
    - **Development**
        - The thermal control energy efficiency application and User comfort applications try to send contradictory orders on the same temperature actuator.
        - The Context Monitoring and Actuation Conflict Management Enabler at Dev time analyses context (environment) related variables that provide clues about the potential conflicts in actuation orders.
        - The Enabler identifies the exact point of conflict.
    - **Termination**
        - Both applications' designs are corrected with recommendations from the conflict handling mechanisms in the Enabler.

- **Testing Constraints**

  No further constrains found at this stage.

- **Control**

  The operation is considered automatic.

- **Test Relationships**

  None.

Verification Plan

- KPI: Efficiency in actuation conflict detection.
- KPI: Understandability and easiness of design update recommendation.

A.3.1.3 Thermal comfort control – conflict in temperature actuation tests

ENACT references

Enabler reference: Context Monitoring and Actuation Conflict Management Enabler.

Enabler tool: Application and Context Monitoring and Actuation Conflicts Handling.

Test 3.2.0.2 Actuation conflicts on the same physical variable

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Enablers involved**

Enabler
Behavioural Drift Analysis (BDA) Enabler
Orchestration and Continuous Deployment Enabler

- **Description**

  An IoT thermal comfort application controls the heating in the room according to the elderly patients' preferences. For example, in winter the room should be maintained to 23 ºC.

  Another IoT energy efficiency application in the thermal control of the building has a setting of stopping the heating when it reaches 20 ºC in winter. When the sensed temperature is between 20 and 23 degrees the orders over the heating collide.

  In operating the building, both applications run simultaneously, though there has been specified the priority for IoT user comfort application in winter and for the IoT energy efficiency application in summer. The ENACT context Monitoring and Actuation Conflict Management Enabler is used to manage actuation conflicts.

- **Interfaces of the Enabler Scenario**
  - **The users' thermal comfort IoT application**: The IoT application that keeps the temperate according to users' preferences.
  - **The energy efficiency IoT application**: The IoT application that minimizes energy consumption.
- **Testing Inputs**
  - **Business information**: Information related to the temperature values of the thermostat, energy consumption values of heating, related environmental sensorized values such as humidity and other contextual information, e.g. room location.
  - The Finite State Machines (or other type of behavioural models) of both applications needs to be created.
- **Testing Outputs**
  - A detected conflict in the use of the same actuator and recommendation to avoid it in the design of the IoT applications.
- **Test Procedures**
  - **Initialization**
    - The user comfort application and the thermal control energy efficiency IoT applications want to be deployed in the same IoT environment so they can co-exist. They may be sending contradictory orders on temperature status by acting simultaneously on the different actuators that control the temperature. The designers need to analyse possible conflicts.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

        - A unique model of desired behaviour in the smart environment is developed to be the input of the Context Monitoring and Actuation Conflict Management Enabler.
    - **Development**
        - The thermal control energy efficiency application and User comfort applications try to set contradictory temperature values by sending orders to different actuators that impact the temperature (same physical variable).
        - The Context Monitoring and Actuation Conflict Management Enabler at Ops time senses context (environment) related variables that provide clues about the potential conflicts in actuation orders.
        - The Enabler identifies the exact point of conflict.
    - **Termination**
        - Both applications' designs are corrected with recommendations from the conflict handling mechanisms in the Enabler.
- **Test Procedures**
    - **Initialization**
        - The user comfort application and the thermal control energy efficiency IoT applications want to be deployed in the same IoT environment so they can co-exist. They may be sending contradictory orders on temperature status by acting simultaneously on the same actuator or on different actuators that control the temperature. The designers need to analyse possible conflicts.
    - **Development**
        - The Context Monitoring and Actuation Conflict Management Enabler detects contradictory actuation orders sent to the same actuator (or different actuators impacting the same physical variable) by two different IoT applications.
        - The Context Monitoring and Actuation Conflict Management Enabler recommends conflict resolution strategies in the design of both IoT applications.
    - **Termination**
        - The most suitable actuation control is performed.
- **Testing Constraints**

  No further constrains found at this stage.
- **Control**

  The operation is considered automatic.
- **Test Relationships**

  None.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

Verification Plan

- KPI: Efficiency in actuation conflict detection.
- KPI: Understandability and easiness of design update recommendation.

A.3.1.4 Smart building alerts for user comfort tests

ENACT references

Enabler reference: Security and Privacy Monitoring and Control Enabler.

Enabler tool: Security and Privacy Monitoring and Control.

Test 3.3.0.1 Trustworthy smart building alerts

- **Enablers involved**

Enabler
Security and Privacy Monitoring Enabler
Security and Privacy Control Enabler

- **Description**

  In the smart building, the IoT user comfort alerts application is used to ensure the safety of the occupants of the building. The IoT user comfort alerts application is continuously monitoring the values of the sensors for avoiding safety problems in the building. When one of the alarm events is detected, an alarm is sent to a care taker, including the nature of the alert and a level of priority.

  It is important to avoid false positives in alarm situations, so the integrity and security of the data processed is key. The integrity and confidentiality of the information that reaches to the care takers should be guaranteed. In case incidents or attacks (tampered of the data) are detected the application will be informed and security enforcement measures applied.

- **Interfaces of the Enabler Scenario**
  - **The users' thermal comfort IoT application**: The IoT application that triggers safety alerts in the smart building.
- **Testing Inputs**
  - **Business information**: Information related with the status of the building safety alerts, communication security parameters, and IoT system configuration.
  - IoT application user groups and their permissions.
- **Testing Outputs**
  - Alerts and reaction recommendations whenever a security incident occurs.
- **Test Procedures**

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).

- **Initialization**
  - The metrics to monitor in the Security and Privacy Monitoring Enabler are set by the IoT system operator together with their alarm thresholds.
- **Development**
  - The Security and Privacy Monitoring Enabler is constantly checking the security of the communications in the building and the protection of data exchanged between the devices and the cloud.
  - The Security and Privacy Monitoring Enabler controls access to the services in the IoT systems.
  - An alarm is set and the Security and Privacy Monitoring Enabler evaluates the whole status of the communications.
- **Termination**
  - The confidentiality of the communications is controlled.
  - The notification on the detected security incident is sent/showed to the operator of the IoT application.

- **Testing Constraints**

  No further constrains found at this stage.

- **Control**

  The operation is considered automatic.

- **Test Relationships**

  None.

Verification Plan

- KPI: Efficiency in incident detection.
- KPI: Understandability of alerts.
- KPI: Easiness of enforcement/adaptation recommendations.

The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement n° 780351 (ENACT).