This project has received funding from the European's Union Horizon 2020 research innovation program under grant agreement No. 769355

D7.1 – Integration Report v1

Deliverable No.	D.7.1	**Due Date**	31/10/2019
Type	Report	**Dissemination Level**	Public
Version	1.0	**Status**	Release 1
Description	This document will report about the integration activities in two versions. It will include test plans (individual, module and integration), the different integration environments (lab, real scenario, real data). It will report of all the pilot integration activities including technical, organizational and operational aspects. Associated task: T7.1, T7.2, T7.3, T7.4, T7.5 and 7.6		
Work Package	WP7		

Authors

Name	Partner	e-mail
Benjamin Molina	P01 UPV	benmomo@upvnet.upv.es
Eneko Olivares	P01 UPV	enolgor@teleco.upv.es
Miguel Ángel Llorente	P02 PRO	mllorente@prodevelop.es
Ismael Torres	P02 PRO	itorres@prodevelop.es
José Antonio Clemente	P02 PRO	jclemente@prodevelop.es
Flavio Fuart	P03 XLAB	flavio.fuart@xlab.si
Erwan Simon	P05 CATIE	e.simon@catie.fr
Romain Quéraud	P05 CATIE	r.queraud@catie.fr
Charles Garnier	P05 CATIE	c.garnier@catie.fr
Gilda De Marco	P04 INSIEL	gilda.demarco@insiel.it
Luka Traven	P08 MEDRI	luka.traven@medri.uniri.hr
Tamara Corsano	P09 SDAG	t.corsano@sdag.it
Andrea Chinese	P09 SDAG	a.chinese@sdag.it
Cinzia Ninzatti	P09 SDAG	c.ninzatti@sdag.it
Stefano Bevilacqua	P12 ASPM	s.bevilacqua@monfalconeport.it

History

Date	Version	Change
10-June-2019	0.1	ToC and task assignments
01-Sep-2019	0.2	ToC updated and first version of the Testing techniques description
21-Sep-2019	0.3	Energy Management trial strategy
02-Oct- 2019	0.4	Pixel Platform Description
10-Oct-2019	0.5	Testlink tool description and tool available
01-Nov	0.6	Test plan and schedule of pilots
20- Nov	1.0	Final version to be submitted

Key Data

Keywords	Use Case, Ports, Test Plans, Validation
Lead Editor	Ismael Torres - P02 PRO
Internal Reviewer(s)	Eirini Tserga - P10 THPA , Deborah Mille - P07 CREO

Abstract

The present document is the first version of the “integration report” and the main objective of this document is to guarantee the successful integration of the Pixel platform, the development of the four use cases, the application of the Port environmental Index (PEI) in all of them and the inter-pilot integration. Moreover, it also validates the Pixel architecture including the different components and modules developed in WP4, WP5 and WP6.

Each of the use cases is associated with one of the ports of the project and tries to solve a real need of the port.

The energy management trial will be implemented in the Port of Bordeaux. The objective of the pilot is to understand the energy needs and the impact of the different port activities in terms of energy consumption. The Intermodal transport trial will be implemented in the Port of Monfalcone. The objective of the pilot is to improve intermodal transport, pushing connection by rail between Port of Monfalcone and SDAG to increase efficiency of logistics. The Port-City integration trial will be implemented in the Port of Piraeus and Thessaloniki. The pilot will focus on the planning and optimization of urban logistics.

To validate the technical developments made in this project, including the pilots, there are a lot of testing approaches, techniques and tools. In order to do the validation, we have been following ISTQB guidelines based on the best practices and testing standards. The three main testing techniques used in the project are: Individual test (unit test), focussed on the validation specific functionalities, Module test, focused on the validation of modules or components, and last but not least, we have used integration test that allow us to validate the integration of different components and use cases. There are more techniques and approaches to validate software components, but the selected ones are the most common and are considered adequate to have a good validation.

The management of test cases and their execution is not an easy task, because the test cases are in evolution during the development of the project. As for the management of the tasks, it was decided to use a task management tool "Jira". To facilitate the management of tests, the web tool “TestLink” is going to be used, that is one of the most widely used tools to carry out the management of software tests.

Statement of originality

Table of contents

List of tables

List of figures

List of acronyms

Acronym	Explanation
AIS	Automatic Identification System
ARPA	Regional Environmental Agency of Friuli Venezia Giulia
ASPM	CCIAA DI GORIZIA - AZIENDA SPECIALE PER IL PORTO DI MONFALCONE
ATD	Actual Time of Departure
BOD	Biological Oxygen Demand
BTEX	Benzene, Toluene, Ethylbenzene and Xylene
CO	Carbon Monoxide
DAL	Data Acquisition Layer
D&N	Dashboard and Notification
ETD	Estimate vessel’s time of departure
FEL	Facilitation of International Maritime Traffic
GHG	Green House Gases
GPMB	Grand Port Maritime de Bordeaux - Port of Bordeaux
IH	Information Hub
IoT	Internet of Things
IS	Information System
KPI	Key Performance Indicators
NOx	Nitrogen Oxides
O3	Ozone
OT	Operational Tools
PAS	Port Activity Scenario
PEI	Port Environmental Index
PIXEL	Port IoT for Environmental Leverage
PMIS	Port Management Information System
PM10	Particulate Matter 10 micrometres or less in diameter
TDS	Total Dissolved Solids
ThPA	Thessaloniki Port Authority
REHO	Regional Environment and Health Observatory
SO2	Sulfur Dioxide
KPI	Key Performance Indicators
SDAG	Stazioni Doganali Autoportuali Gorizia S.p.A.
S&P	Security and Privacy
WP	Work Package

1. About this document

The scope of the deliverable is to report about the integration activities. It will include test plans (individual, module and integration) and the different integration environments. It will report on all the pilot integration activities including technical, organizational and operational aspects. First version of this proposal will be key to gather requirements and test plans before the execution of the pilots, being a useful instrument to report the state of the WP and the integration activities

This deliverable will have a second version on month 27 with the initial results of the validations and the updated versions of the test plans and test cases.

Testing work done in this deliverable, not only helps to verify that the software meets the requirements, but it also helps to clarify what is to be developed.

1.1. Deliverable context

Table 1: Deliverable content

Keywords	Lead Editor
Objectives	*Objective 1:* Definition of evaluation methodology to be applied for the technical developments of the projects including the pilots.
	Objective 2: *Definition of the test plan and test cases to be applied for the pilots and for the different integration.*
Work plan	This deliverable reports about the works performed in task 7.1, 7.2, 7.3, 7.4, 7.5 and 7.6. The results reported in this document are also important for the work done in WP4, WP5 and WP6, due to work of wp7 is based on the developments of the mentioned packages
Milestones	Direct contribution to Indirect contribution to
Deliverables	Detected inputs: • D6.1 and D6.2 describe the Pixel architecture to be validated • D4.2 and D4.3. Models are fully described with their requirements for integration in D4.2. Moreover, predictive algorithms are being developed and a first description is available in D4.3. • D3.2 and D3.4 describe the requirements of the pixel architecture and the pilots to be developed and evaluated in the WP7. Detected outputs: • D7.2 and D7.3 will provide an updated information this deliverable

Risks	This deliverable helps mitigating the risks

1.2. The rationale behind the structure

The deliverable has been structured in three main sections, which explain the different elements to be tested, which testing techniques are going to be used and the definition of the specific test cases to execute.

The PIXEL platform and use cases test plans section, contains an overview of the integration activities and pilots to be carried out in this work package with the initial test plan.

The test techniques section, explains which testing techniques are going to be applied to validate the different integrations and pilots. In addition, it introduces "TestLink", a test management tool that is going to be used, in order to define the test cases.

The pilot and component test cases description section, define the specific test cases created and the relation with the requirements that are going to be evaluated. Test cases have been defined though the TestLink tool, while an example of this activity has been included as an annex. In order to have a complete description of the test cases, the TestLink should be accessed.

The last but not least, is the conclusion section, summarising the deliverable with the most important conclusions and objectives to be achieved, at the end of this work package.

1.3. Version-specific notes

This document is the first version of the deliverable "Integration Report" (D7.1). This document must be exhaustive in the definition of test plans and test cases of the different modules / components and pilots to be developed during the execution of the project and to be used as the basis, in order to validate the quality of the artefacts developed.

This document provides an initial version of the test plans and test cases, which can evolve throughout the project, so they are defined by a web application, that allows its modification and evolution.

The second and final version of the document (D7.2), with due date in month 27, will contain any modification performed in the test plans and test cases. It will also include the initial execution reports with the test cases that have been passed and those that remain to pass, in order for the developments to be finalized.

2. Motivation

In order to develop useful and quality software, the first step is to clearly define the problem that requires a solution, the way (methodology) that is going to be solved and which specific functionalities of the solution will have to be provided to the user. For the specification of the problem, techniques of elicitation of requirements, that consist of describing as clearly as possible the problem to be solved, are commonly used. Among these techniques, the definition of use cases is quite popular. These tasks are usually carried out both in product development for customers and in research projects, as in this case. In Pixel, the definition of the project requirements, were a part of Work Package 3..

On the other hand, to ensure the quality of a software, the definition of concrete test cases is required , so as to verify that the software meets the requirements, not only the functional requirements, but the non-functional as well. Quality assurance is a bit forgotten in the world of software development, where not all software development companies have a testing phase in their development cycles to ensure the quality of the products they develop. And if we focus on the world of research projects, the situation is even worse and rarely are tests defined. Testing work not only helps to verify that the software meets the requirements, but the benefit of testing goes beyond verification, if applied from the early stages, assisting in clarifying what is to be developed, involving end users or customers. It also provides extra valuable information for developers, who have a better vision of the work to be done.

In this project, software quality is considered a top priority and therefore, in WP7 an important task will be developed, defining test plans and test cases to ensure that the software components and pilots, have at the end of the project the desired quality and thus facilitate the exploitation of them, once the project is finished. It is very important that during the development of integration activities and pilots, the plans and guidelines defined in this deliverable, are followed.

3. PIXEL Platform and use cases Test Plan

3.1. PIXEL Platform

WP6 is in charge of integrating all the modules developed in WP6 PIXEL Platform", applying properly the models described in WP4 and applying the methodology and techniques of WP5 to calculate the PEI. All these elements will we used to develop the pilots.

This section summarizes how these software components are going to be validated before applying them to the pilots.

3.1.1. PIXEL Platform Overview

The PIXEL platform mainly refers to a set of open components aimed at harmonizing both the IT and operational worlds within ports, narrowing the gap towards the Port of the Future. The problem about the lack of IT and operational integration is common in many companies, and the current industrial digitalization movement tends to reduce internal silos and facilitate internal cooperation and synergies to be more effective and productive. Ports are probably a paradigmatic case because different companies work together in the logistics chain, and therefore both the current limitations as well as the potential synergies become clearer.

Bringing heterogeneous companies together is quite a big challenge and requires a lot of effort. Fortunately, technology development is speeding up such process, creating a trusted and secured framework for exchanging information and services among stakeholders. The PIXEL platform comes into scene here, by providing an open environment that gets data from IoT platforms, merging them into a harmonized hub, building analytic models on top of that, and finally presenting relevant information in a dashboard, to help operational management and decision making.

The architecture of the PIXEL platform has been extensively described in deliverable D6.1 and is to be updated in deliverable D6.2. However, a brief summary is included in this chapter for a better understanding. Figure 1 depicts a diagram of the architecture with the following modules (bottom-top approach):

- **Data Acquisition Layer (DAL)**: it is responsible for connecting with the different data sources available in the port: sensors, legacy systems and other services. It represents the ground level of an IoT platform, but it is very important as the number of heterogeneous sources in ports can be potentially huge.
- **PIXEL Information hub (IH)**: data from the sources are merged and stored in the information hub in an intelligent way (e.g. by using convenient data formats), so that they can be later accessed to check historical data or feeding analytics engines.
- **Operational Tools (OT)**: the operational tools are mainly responsible for executing models and predictive algorithms able to work with the existing data in the Information Hub and provide relevant results to monitor or help in the decision-making process. Such models and predictive algorithms (e.g. traffic, energy, environment) are specific for each port and will be subject of special assessment.
- **Dashboard and Notification (D&N)**: The dashboard represents the UI presented to the port operator to visualize the results, configure the values and notifications, run and schedule simulations, etc.
- **Security and Privacy(S&P)**: the whole environment needs to be secured affecting all previous components. Identity management and access policies are part of this cross-layer component.

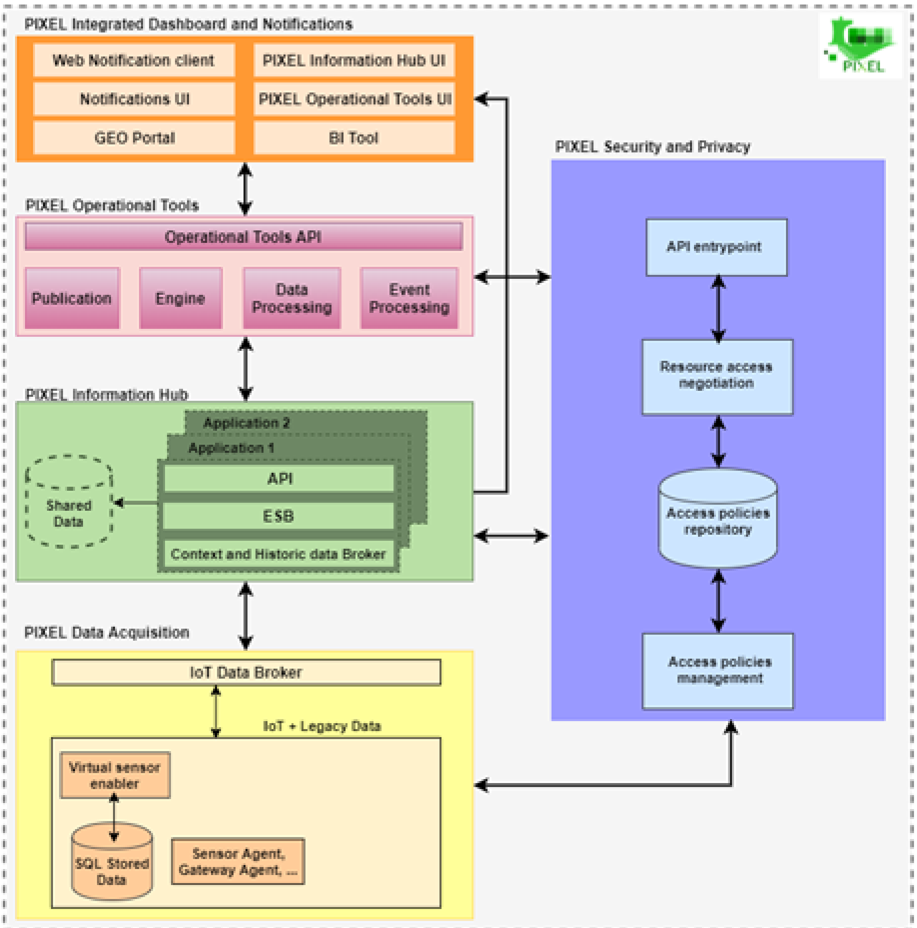


Figure 1: Global architecture of the PIXEL platform

3.1.2. Algorithms and Models description

PIXEL's models (developed/used in WP4) aim to define and implement port's activities models to be used in order to manage the port in an efficient way and adapted to the environmental needs, including energy demand and production, hinterland multimodal transport, atmospheric and noise pollution. All of these models are fully described with their requirements for integration in D4.2. Moreover, predictive algorithms are being developed and a first description is available in D4.3.

Port and City Environmental Management Model

The knowledge and modelling of the supply chain and port's activities (type of machine, duration of use, position in the port) enables the compilation of activity scenarios that are used to identify the energy sources, local emissions of pollutants, but also to estimate the flow of cargoes entering or leaving the port. Using this approach, the modelling scenario will be used by the ports, as a support for decision making. This model has been developed by PIXEL partners and utilizes in a JSON files system. Readers will find a full description in section 3.2.4 of deliverable D4.2. Then an application has been written in Python 3.6 with "pipenv" [1], in order to manage the different files. The main test for this PIXEL model will consist to check that the data format is fully respected and that the application in python runs well. Moreover, UI and dashboard have to be tested in real conditions in WP7.

Energy demand prediction

For operational actors in ports, energy efficiency is an important issue, both from the environmental and economic perspectives. In order to optimize the energy flows inside the port, a prerequisite is to manage and quantify energy consumption and production in the context of complex industrial processes. PIXEL enables such quantification for small and medium ports by focusing on the three complementary elements:

- energy consumption (fossils and electric) associated to cargo transition across the port,
- energy produced by local sources such as photovoltaic panels,
- balance between energy consumption and production.

This approach enables a predictive estimation if the port activities scenario is known (e.g. maximum net energy consumption during next week). The main test for this PIXEL model, will consist to check that the data format defined in WP4 is fully respected and that the link between the PAS and the energy model works well. Similarly, to Port and City Environmental Management Model, , UI and dashboard have to be tested in real conditions in WP7 as well.

Hinterland multimodal transport model

This model will assist in understanding whether a different traffic management has a positive impact on congestion issues, citizens risk on the road and environmental impacts. The main result that will be reached, is an effective decision support tool, to optimize the re-routing of trucks from the Port towards SDAG or other infrastructures, and so facilitate the staff working in both entities (between ASPM and SDAG). The model has been split in two sub models, one related to intermodality way of transport (applied on slabs traffic between ASPM and Aussa Corno industrial district) and the other one, related to traffic and parking inside port premises and the possible rerouting to SDAG. The first one is going to be called "intermodality sub model" and the second one "traffic sub model". The models were developed based on the reality of port activities and as close as possible to the practices. Thus, the modeling steps use common sense and business expertise. The modeling choices that have been made, have therefore been validated by the port partners. The main test and validation will consist in the correct parameterization of the models, according to the use-case considered and the data available.

Environmental dispersion models

Cargo handling within ports, and to/from the hinterland loading and unloading cargo, undeniably involves emissions of pollutants. For example, loading or unloading a cargo of dry bulk can cause a significant amount of dust in the atmosphere and generate a lot of noise. Thus, the sources of emissions generated by the supply chain of cargo, are not negligible and have direct effects on the port environment. Based on existing measurements and port activity data, the air and noise model should simulate the air and noise pollutant dispersion, caused by various activities and operations taking place within the port. By using historical data, it will simulate the current state of the ambient air/noise in the port and with the help of weather prediction models, it should be able to simulate the possible future air/noise quality. These simulations can assist the port manager/operator in the decision-making process, in order to optimize various activities within the port and minimize their impact on the environment. For both, air and noise modelling state-of-the-art software have been used. This software is fully validated so the main point here will be to tune them correctly according to the use case considered.

Predictive algorithms

The following predictive algorithms were identified in D4.3.

- **Prediction of vessel call data from FAL forms and other sources**: In this task, internal data about vessel calls is utilized to predict vessel calls and their durations. General statistical analysis and visualizations are also performed. Vessel call data is available in every port as is obtained from FAL forms which are legally required, thus making this task generally applicable to every port at a low cost.

- **Use of AIS data:** AIS data is widely used in maritime domain and is becoming extremely useful for data analytics tasks, especially because of its quantity. In this task we visualize and analyse the data around the ports, provide port congestion indicators out of AIS data and to some extent ETA prediction for the incoming ships.
- **Use of satellite imagery**: Obtaining operational insights from remote sensing imagery presents an emerging field, offering the ports increased situational awareness by giving them the ability to monitor their port from Space and compare it in a global perspective, to understand their unique differentiators in the global market.
- **Analysis and prediction of road traffic conditions with connection to port operations**: In this task, hinterland multimodal transport requirements in the Port of Monfalcone, Port of Piraeus and Port of Thessaloniki are addressed. A common task of short-term traffic volume prediction has been identified. Predictions will be correlated with port operations to provide estimates on the impact that congestion has on them.
- **Prediction of renewable energy production**: In this task, ports are provided with the ability to estimate the potential of renewable energy production for different time resolutions. Task is focused on the Port of Bordeaux use case, but these methods are general and applicable for any port. Different open data sources are identified about the weather and the measured photovoltaic power. Live, as well as historical data sources are presented, with initial results based on this data.

3.1.3. Platform and modules Test Plan description

The main objective of this deliverable is to identify integration requirements, in all of the modules of the aforementioned architecture, in order to be able to run the pilots smoothly. As there is a second version, the main aim in this version relates to building an integration plan for each pilot, listing needed requirements and propose test plans to ensure readiness before the start of the pilots, focusing mainly in the technical approach. The plan will cover not only the technical dimension, but also the organizational and operational one:

- The **technical/technological dimension** refers to those needs (software, hardware) related to being able to deploy successfully the PIXEL platform and connecting it with the systems (hardware, software) already available in each port. It may also include sensors that have been bought for a particular port.
- The **organizational dimension** refers to aligning ports' internal and external factors with the PIXEL platform deployment
    - Internal factors include the management style and how things are organised internally. For example, if the hardware deployment of the PIXEL platform takes place in port premises, it should follow a certain methodology and policies set by the port, if the PIXEL platform is mainly deployed in the cloud, more flexibility is gained.
    - External factors include local or national economic conditions, specific regulations or sociocultural trends influencing the environment, in which the port operates. This may determine, for example, when to start the pilot, which days, for how long, etc.
- The **operational dimension** refers to those aspects that ensure the performance of the system under operation, monitoring errors and managing service quality. It relates more with business process and decision strategies. Some generic examples could be:
    - Trigger detection: what are the events that trigger a business process (model/ predictive algorithm)?
    - How is data feeding the business process (model/ predictive algorithm)? Is data complete/incomplete/reliable/predicted?
    - Is the output data properly visualized to operators to help in decision making process?
    - Is the decision being made automatically on the output result of a model/predictive algorithm or does it require human confirmation?
    - What kind of other controls are available for the business process (e.g. notifications)?

3.2. Energy Management Trial

3.2.1. Scenario and scope

GPMB use-case aims to use PIXEL in order to:

1) ***Understand the energy needs and the impact of the different port activities in terms of energy consumption;***
2) ***Be able to optimize and well dimension renewable energy networks.***

These two questions will be addressed one after the other, since we first need to understand energy needs before suggesting an optimisation strategy for renewable energy. Today, there is no tool in GPMB to understand energy consumption of port activities. Energy models and the PIXEL platform will therefore be used by GPMB as a tool, in order to have a better understanding and develop optimized strategies for energy saving.

In deliverable D3.4 "Use cases and scenario manual V2", GPMB has described eight different scenarios to test and integrate with the PIXEL platform:

- Statistics manager scenario (GPMB-StM-1) which aims to provide a data analysis of vessels calls, by connecting VIGIEsip to Pixel Platform, to obtain historical data and then use small analytics tools. This means the integration of the data acquisition layer of PIXEL developed in WP6 with the PMS of GPMB, to acquire and then analyse vessels' calls.
- Energy manager scenario (GPMB-EM-1) which aims to understand the energy consumption of the port, based on vessels' calls and supply chains description. To do so, the port activity scenario and the energy models developed in WP4, will be integrated, parametrized and used.
- Energy manager scenario (GPMB-EM-2) which aims to connect smart electrical sensors to the PIXEL platform. Today, these data have not been analysed and used for decision-making. This means the integration of the data acquisition layer of PIXEL developed in WP6 with the 13 industrial programmable electric automatons.
- IT manager scenario (GPMB-IT-1) which aims to connect old sensors of the port to the PIXEL platform.
- Environmental manager scenario (GPMB-EnvM-1) which aims to use PIXEL and PEI as a support for environmental decisions.
- Port manager scenario (GPMB-PM-1) which aims to use PIXEL as a support for investment in green energy. For this point, an investigation will be conducted on how old sensors can be included in the PIXEL platform and whether new system connections have to be developed/deployed.
- Software editor scenario (GPMB-SE-1) which refers to the integration of PIXEL platform in VIGIEsip (GPMB's PMS). This means that internal works must be done in VIGIEsip in order to be able to communicate with PIXEL.
- Port Agent scenario (GPMB-PA-1) which refers to transferring or extending some of the previous functionalities/models to the port ecosystem (e.g. terminal operator). It is therefore extrapolated from the scenarios GPMB-EM-1 and GPMB-PM-1.

3.2.2. Pilot Plan

The **¡Error! No se encuentra el origen de la referencia.** describes the responsibilities with the d evelopment of the Energy management pilot and the dependencies with other work packages.

Table 2: Energy management trial - Roles

Title:	Energy Management Trial: Port of Bordeaux			
Partners:	Number	Name	PM	ROLE
	4	CATIE	8	Coordination of the intergation activity. Test and validation of PAS and energy models,
	13	GPMB	2	Use-case location. Provide all necessary data to test energy models and Implement the standard interface between PIXEL and PCS in VIGIEsip
	6	ORANGE	2	Provide knowledge in data manageent and communication to carry out the pilot
	1	UPV	2	Provide knowledge in use of operational tools and aspects related with information interchange. Provide all necessary inputs to integrate PIXEL platfrom in GPMB
	2	PRO	3	Provide knowledge in use of PIXEL Dashboard and aspects related with information interchange. Provide all necessary inputs to integrate PIXEL platfrom in GPMB
Start:	01/09/2019			
End:	30/01/2021			
Related WPs:	WP6	Enabling ICT infrastructure framework	**Batch 1: Data acquisition & Information Hub** **Batch 2: Operational Tools & Dashboard**	
	WP7	Pilot trials integration, deployment and evaluation		
	WP8	Assessment and expansion plan		
Related Deliverables:	D3.2	PIXEL Requirements Analysis	Informational	
	D3.4	Use cases and scenarios manual v2	Use cases inventory	
	D7.1	**Integration with existing systems**	**Initial status of software / plan to follow**	
	D8.1	Evaluation plan	What will be measured	
Other prerequisites:	**T7.1 - Phase 1**	**Connect and test all the software components developed in PIXEL in a testbed environment**		
	T7.1 - Phase 2	Effectively integrate the PIXEL Information System resources to the existing data sources in the different pilot scenarios		

The initial Gantt diagram for the development of this pilot is:

Year	Who ?	2019				2020												2021
Month		9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1
Phase 1: connect and test all the software components developed in PIXEL in a testbed environment																		
Connection with VIGIEsip for vessels calls (API, NGSI Agent)	Orange and GPMB																	
Connection with electrical sensors and wheather station	CATIE, Orange, GPMB																	
Definition of equipment needed to deloyed PIXEL in GPMB premise	UPV, PRO, CATIE, GPMB																	
Definition of sensors to deploy for the energy management trial	CATIE, GPMB, UPV, PRO																	
Test of Information Hub and data acquisition layer for energy trial	UPV, PRO, ORANGE																	
Test of the energy demand models																		
Energy model parametrisation: supply chain definition, machine specification, building, ...	CATIE, GPMB																	
Preparation of an input data models of vessels calls (based on past data)	Orange, GPMB, CATIE																	
Technical test and validation of energy demand model with past data (comparison with bills or expert knowledge) using operational tools	CATIE, UPV, GPMB																	
Link energy demand model with NGSI agent and test of operationnal tools and information hub	CATIE, UPV, Orange																	
Test with end-users and feedback on the dashboard	GPMB, PRO, CATIE																	
Understand the energy need / Impact of the different port activities																		
Data analysis of energy consumption (peak, seasonality, temporality, ...)	CATIE, GPMB																	
Be able to optimize and well dimensioning renewable energy networks																		
PVGIS simulation to obtain estimation of prodcution	CATIE, GPMB																	
Comparison of production and consumption	CATIE, GPMB																	
Propostion of a stratagy of energy optimisation	CATIE, GPMB																	
Pilot test in real condition																		
Phase 2: Effectively integrate the PIXEL Information System resources to the existing data sources in the different pilot scenarios																		
WP6 1st BATCH Information Hub and data acquisition layer																		
WP6 2nd BATCH																		
SOFTWARE AND HARDWARE INTEGRATION	Tentative																	
VALIDATION	Tentative																	
TECHNICAL IMPACT ASSESSMENT	Tentative																	
PROOF OF CONCEPT	Tentative																	

Figure 2: Energy management trial - Gantt

The pilot is divided in two phases:

(i) phase one is devoted to connect and test all the software components developed in PIXEL useful for the GPMB use-case. In this phase, energy models will also be tested and validated. This first phase is composed of:

- Connection of the different data sources endpoints to the PIXEL Platform.
- Test and validation of the energy demand model in real conditions.
- Data analysis of the energy consumption with real data.
- Dimensioning of renewable energy (potential) network.

(ii) phase two regards the deployment of the full PIXEL platform in GPMB premise. This second phase will consist in the integration of all PIXEL components in order to be able to perform a pilot test during the last four months of this task.

According to the previous chart, the estimated date of finalising the pilot is September-December 2020. During these four months, data will be captured, and models applied in the real environment, providing real information about the performance of data. This period will be used to provide useful feedback for the technical impact assessments, especially for the quality in model use.

The most important deadlines for the integration plan are:

- End of December 2019: all data-sources endpoints tested and the communication with DAL validated.
- End of April 2020: Energy models have been tested and GPMB have been able to collect all the data needed to fully describe the different supply chain.
- September 2020: PIXEL platform deployed in GPMB and starting of the pilot test in real conditions.

This is a preliminary Gantt chart and schedule that may evolve, since it could be subject to some delays due to integration work and WP6 developments. An updated version will be provided in the D7.2.

3.2.3. Software integration

GPMB has developed its own port management system called VIGIEsip. That is why the main software integration actions are directly linked with it. Other software integrations are linked with air quality and weather conditions in the area around the port. The following table lists the different needs about software integration for GPMB. First insights on how and when the integration will occur, are also provided. This list is subject to changes depending on the progress of the PIXEL project and new integration paths that may emerge.

Table 3: Energy management trial software integration

Scenario	Software integration	How	When
GPMB-StM-1 *GPMB-EM-1*	Integration of VIGIEsip in order to collect information about ship call.	GPMB, trough VIGIEsip will push data every hour and a NGSI agent will acquire and transform them into the data model defined for vessel calls (This data model has been proposed by the port activity scenario model in WP4).	This is already an on-going work and it is planned to be tested and validated before mid-November 2019.
GPMB-EM-1	Integration of supply chains and machine specification.	This is not a direct software integration but there is a need of interface in order to allow a port agent to provide the different specifications of supply chains and engines	In a first phase (until January 2020), we will consider two specific supply chains and data will be entered manually. In a second phase (after April 2020), GPMB will contact every terminal operator to obtain data. An interface will have to be developed for this.
GPMB-EnvM-1	Integration of open-data coming from ATMO Nouvelle-Aquitaine for air quality monitoring.	ATMO Nouvelle Aquitaine provides webservices [2] PIXEL data acquisition layer could use existing FIWARE data model for air quality [3] An NGSI Agent to import those open data will be developed.	This work will be done between January and March 2020.
GPMB-EM-2	Integration of weather data.	Data can be acquired via an API https://developer.sencrop.com/ An NGSI agent will be developed to acquire those data.	This is still an exploratory work and it is planned to be tested and validated before December 2019.

3.2.4. Hardware integration

Some hardware integration has to be done, so as to gather data that are directly linked with the electrical energy used by GPMB. The two other hardware integration items currently identified, are related with weather data and connection with old sensors. The following table lists the different needs about hardware integration for GPMB. First insights about how and when the integration will occur, are also provided. This list is subject to change, depending on the progress of the PIXEL project and new integration paths that may emerge.

Table 4: Energy management trial hardware integration

Scenario	Hardware integration	How	When
GPMB-EM-2	Integration of electrical sensors for measuring electrical consumption of the port.	GPMB will push data every ten minutes. The description of the data is documented. PIXEL data acquisition layer could use existing FIWARE data model for energy [3] with some adjustments An NGSI Agent is in development in order to allow the GPMB solution to push data on PIXEL.	This is already an on-going work and it is planned to be tested and validated before mid-November 2019.
GPMB-EM-2	Integration of a weather station	GPMB is purchasing a weather station. When it becomes available, we will define the endpoints for data acquisition. Then the FIWARE data model for weather could be used [4]	This integration will start when the weather station will be purchased by GPMB. It is planned to have it between January and March 2020.
GPMB-IT-1	Connection of old sensors to the PIXEL platform with new standards	This point still needs to be deeper investigated by GPMB, CATIE and Orange.	This integration is planned to be deeper investigated after January 2020.

3.2.5. PIXEL modules/tools to be validated in the use case

GPMB is the PIXEL use-case related with energy management. That is why, a huge focus is placed on the test and validation of energy models. The main PIXEL features that are going to be tested and validated are the following:

- PIXEL Models
    - Port activity scenario (PAS) for the supply chain modelisation.
    - Energy demand model to estimate the energy consumption needed for un/loading vessels.

    - Energy production model to estimate the potential of PV system in GPMB.
- PIXEL Predictive Algorithms
    - Prediction of vessel calls from FALL forms and other sources.
    - AIS-driven predictive algorithms.
    - Prediction of renewable energy production.
- Port Environmental Index (PEI)
- PIXEL Platform
    - Data acquisition layer: test and validation with GPMB data sources.
    - Information Hub: to store and manage electrical sensors measurement, ship calls, supply chains and engines specification, weather data, open data of air quality measurements.
    - Operational tools: to test and validate energy models, predictive algorithms and PEI.
    - Dashboard: to test and validate the dashboard and the notifications linked with energy management and PEI.

Independently of the use case, core modules of the architecture need to be deployed and tested on each site. Therefore, a set of common testing methodology is envisioned for the core, which is described in Section 5.

Table 5 Energy management trial- Energy demand - needs

Model: Energy demand. Energy needs for loading and unloading cargo [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Build any kind of port activity scenario, based on past data and on predicted vessel calls.	Test, validation and integration of the port activity scenario builder to generate inputs for the energy model.	Vessels' call available through DAL and IH Supply chains and machines description available through DAL and IH Vessels calls prediction (pattern detection) available in IH Port activity scenario model previously published and deployed in OT.
Use of the port activity scenario as an input for the energy model	A defined vessel call list and supply chains and machines description generate a port activity scenario and link it with the energy models.	Previous features validated. Energy demand model previously published and deployed in OT.
Test and validation of energy model	For a specific supply chain, the results of the energy demand model will be validated using expert knowledge.	Full description of a specific supply chain with its associated machines

Model: Energy demand. Energy needs for loading and unloading cargo [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Characterize the energy demand generated by recurrent port activities	For a vessel calls list representative of a typical year the energy demand generated by recurrent activities will be studied.	Past data of vessels calls available through DAL and IH. A typical year of vessel calls is available.
Specify the energy needs in terms of frequencies, duration and power	For a vessel calls list representative of a typical year the energy needs will be investigated.	Past data of vessels calls available through DAL and IH. A typical year of vessel calls is available.
Predict the energy demands and peaks.	For a predictive vessel calls list, we will study the energy needs.	Predictive data of vessels calls available through DAL and IH. Supply chains and machines description available through DAL and IH
Predict the consumption associated with a port activity	For a predictive vessel calls list, the energy needs will be investigated	Predictive data of vessels calls available through DAL and IH. Supply chains and machines description available through DAL and IH
Predict the port activity environmental impacts.	For a predictive vessel calls list the environmental impacts using emissions factors will be studied	Predictive data of vessels calls available through DAL and IH. Supply chains and machines description available through DAL and IH

Table 6: Energy management Trial- Energy demand- production

Model: Energy demand. Energy production using PV [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Use of PVGIS to evaluate potential PV production	By using PVGIS and considering a typical year, an hourly estimation of the potential PV production, will be provided.	Type of PV system to use is defined Localisation for installing PV system is defined

Table 7: Energy management trial - Energy balance model

Model: Energy demand. Energy balance model and optimization strategies [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Data analysis of energy demand and consumption	Provide a full data analysis of production and consumption of energy inside the port. Analyse if self-consumption is possible and if there is a need for storage.	Estimation of PV potential Estimation of energy demand due to port activities
Recommendations for an optimisation strategy for energy savings	Provide guidelines to help GPMB to make decisions about energy management.	Data analysis of energy demand and consumption
Recommendations for energy production (storage, self-production possibility, …)	Provide guidelines to help GPMB to make decisions about energy management.	Data analysis of energy demand and consumption

Table 8: Energy management trial- Prediction of vessel calls

Predictive algorithm. Prediction of vessel calls from FAL forms and other sources [see D4.3]		
Features to be evaluated	Scenario	Assumptions/Dependencies
ETD model deployed and published through OT.	Integrate ETD prediction algorithm into PIXEL platform	Core architecture modules are already deployed.
Execution of ETD predictive algorithm.	When new vessel call data from FAL forms is available, predictive algorithm will estimate vessel's time of departure (ETD) based on data from FAL form.	Vessels' call is available through DAL and IH. ETD predictive algorithm is deployed.
Evaluation of ETD predictive algorithm.	Upon vessel's departure, ETD will be compared to	Previous ETD features available.

	actual time of departure (ATD).	

Table 9: Energy management trial - Use of AIS

Predictive algorithm. Use of AIS data [see D4.3]		
Features to be evaluated	Scenario	Assumptions/Dependencies
ETA predictive algorithm deployed and published through OT.	Integration of ETA prediction algorithm into PIXEL platform.	Core architecture modules are already deployed.
Execution of predictive algorithm.	When vessel is inside ETA prediction area, a predictive algorithm will estimate time of arrival.	AIS data is available through DAL and IH. ETA predictive algorithm is deployed.
Evaluation of ETA predictive algorithm.	Upon vessel's arrival to the terminal, previous estimations will be evaluated, by computing error metric.	Previous features available.

Table 10: Energy management trial - prediction of renewable energy

Predictive algorithm. Prediction of renewable energy production [see D4.3]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Test and validate the methodology based on data coming from an external source (GPMB is not equipped yet with a PV system)	Use the predictive algorithm for energy production in real time conditions	Predictive algorithm has been developed.
Adapt the developed methodology to GPMB use-case	Provide guidelines to adapt the predictive algorithms to the GPMB use-case when a PV system will be installed.	Data of production for a PV system installed in GPMB

3.3. Intermodal transport Trial

3.3.1. Scenario and scope

The use case of the Port of Monfalcone consists in taking advantage from the PIXEL solution, in order to handle the freight traffic in urban and surrounding area, through the creation of an IoT platform to share data and make interoperable different types of information and sensor systems. The use case will help understanding and measuring the benefits coming from a new model of logistic solutions and the related impact on the environment.

The use-case focuses in solving the main issue related to the lack of a regional logistic platform in FVG Region, which already made some investments on the matter, but could not make possible a joint coordination between ports and inland ports.

In summary, as described in D3.4, in order to address the challenges, the objectives of this Use Case are the following:

- Data sharing between the Port of Monfalcone and SDAG, in order to reduce bottlenecks and congestion in port areas and to promote the use of railways or inland terminals where necessary;
- Reinforce the safety related to ADR transport through the operability of data with regional stakeholders and in connection with SILI;
- Collect data on Regional logistic flows to support the activities of the Regional Environment and Health Observatory (REHO). Here the objective is to give the REHO the possibility to monitor the flows on the road. Within the ARPA (Regional Environmental Agency of Friuli Venezia Giulia), the REHO structure has the aim to put in relation health population status and environmental factors. Moreover, there are air data pollution in open data format available on the ARPA website;
- Collect and analyse data on Regional logistic flows to support the activities of the Regional Government (another public entity), to evaluate specific programs, to promote rail logistic solutions, as well as the implementation/expansion of other logistic infrastructures. This objective refers especially to the side study of slabs traffic from the port of Monfalcone to the regional iron mills. If the externalities related to the slabs transport by truck are measured, it is possible to promote the rail transport mode for this traffic and specific economic bonus for operators that use rail instead of trucks.

The main result expected is to minimize congestion in the port area. Common end users, mainly truck companies and agencies, will benefit from decreased congestion in terms of time, fuel consumption and from a better planning of the logistic solution to be used (promotion of rail transportation, stop planning at SDAG). The Port, as an end user, will benefit from minor externalities related to heavy traffic and overcrowded streets and area, in terms of safety and environmental emissions.

So, the use case is focusing on understanding and measuring the benefits coming from a new model of logistic solutions and the related impact on the environment.

3.3.2. Pilot Plan

The following table describes the responsibilities with the development of the Intermodal Transport Pilot and the dependencies with other work packages.

Table 11: Intermodal transport trial Roles

Code:	***T.7.3***			
Title:	***Intermodal transport trial: Port of Monfalcone***			
Partners:	Number	Name	PM	ROLE
	4	Insiel	8	implementing the the pilot of Monfalcone (Port and SDAG)
	9	SDAG	6	actor of the pilot
	12	ASPM	6	actor of the pilot
	2	PRO	17	support for the integration and testing
Start:	01/09/2019			
End:	30/01/2021			
Related WPs:	WP6	Enabling ICT infrastructure framework		**Batch 1: Data acquisition & Information Hub** **Batch 2: Operational Tools & Dashboard**
	WP7	Pilot trials integration, deployment and evaluation		**Preparation of infrastructure, integration of information systems (data providers)**
	WP8	Assessment and expansion plan		
Related Deliverables:	D3.2	PIXEL Requirements Analysis		Informational
	D3.4	Use cases and scenarios manual v2		Use cases inventory
	D7.1	**Integration with existing systems**		**Initial status of software / plan to follow**
	D8.1	Evaluation plan		What will be measured
Other prerequisites:	**T7.1 - Phase 1**	**Connect and test all the software components developed in PIXEL in a testbed environment**		
	T7.1 - Phase 2	Effectively integrate the PIXEL Information System resources to the existing data sources in the different pilot scenarios		

The Initial Gantt to develop the Intermodal transport trial is:

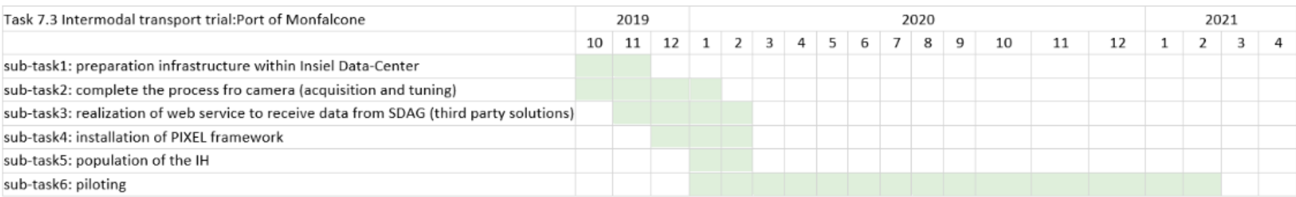


Figure 3: Energy management trial - Gantt

The pilot can be divided in three main phases:

1. Prepare all data to be used (task2, task3, task5)
    - Data coming from SILI (on Regional road and Monfalcone Gate) are already available;
    - Data coming from new camera to know in real time the occupancy of parking area of Monfalcone Port will be available in 3 months (end of January 2020);

- Data coming from SDAG will be available at the most by January 2020;
- Data coming from PMIS (vessels calls) are already available on the Monfalcone website, but we are working to have an integration via web-services in order to collect these data directly from PMIS to Pixel IH;
- Weather data is already available as an open data format.

2. Preparation of the infrastructure and installation of Pixel Framework within Insiel Data Center (task1, task5)
3. Apply models (piloting) (intermodal transport of slabs and Monfalcone parking area management) to manage the synergies between Monfalcone Port and SDAG (already developed including web services to bring data from IH and pass result to Operational Tools).

According to the previous chart, the estimated date of finalising the pilot is February of 2021. By this time, data will be gathered, and models applied in the real environment, providing real information about the performance of data.

This is a preliminary Gantt chart and schedule that may evolve since it could be subject to some delays due to integration work and WP6 developments. An updated version will be provided in the D7.2.

3.3.3. Software integration

The software that will be integrated within PIXEL for the ASPM-SDAG use case are related to the existing SDAG and ASPM information systems, existing information system of other authorities at regional and national level and the new software, purchased for PIXEL scope.

SDAG will mainly use its truck parking management software in order to fulfil the requirements set in PIXEL, and it is based on a physical server available at SDAG facilities.

This software allows to check the entrances and exits of trucks from the parking area, and it also makes available the payment of the ticket. This software allows to manage both the administrative and the maintaining part of the area. The system is accessed by user identification and the various functions are divided by access groups. Printing functions are also available for controlling parking access flows and managing archives.

Actually, the software requires an update to extract the parking area data needed by Pixel. This action is scheduled by the company provider (CAME) by the end of 2019.

The data that will be used in the pixel project are the parking accesses and the history of area usage.

Concerning ASPM within PIXEL will be integrated the following existing software:

- the SILI system, that is the software to manage the requests to access the Monfalcone port and that manages all transits in and out of the port, moreover this system manages the trucks transits in some point of the regional territory (data used in the predictive algorithms);
- the PMIS, that is the software of the National Port Authority, that manages the scheduled plan of the vessel's arrivals to the Monfalcone Port. In this case, the National Authority allows each port to obtain the list of the vessel arrivals, through a national interoperability standard;
- the open data on air pollution available by the website of the ARPA (Regional Environmental and Protection Agency of Friuli Venezia Giulia);
- the software of the smart camera, the new software that will be adopted by Monfalcone within PIXEL, able to produce data on the parking area availability and gate congestion.

3.3.4. Hardware integration

In the ASPM-SDAG use case, the whole PIXEL infrastructure will be replicated and hosted inside the Insiel Data Center. In terms of direct integration of hardware in the use case there is any direct connection between hardware and PIXEL platform. All data that will be used in PIXEL will be extracted from local database in order to be sent to the IH.

3.3.5. PIXEL modules/tools to be validated in the use case

The Pilot will start on January 2020 and approximately will end in 12 months.

The use case will focus on testing the following components of the PIXEL platform, according to the set of requirements expressed by both Monfalcone Port and SDAG:

- Data Acquisition Layer;
- Dashboard and Notification Layer;
- Information Hub;
- Predictive Algorithms;
- Models;
- Operational Tools.

The use case can be seen as a comprehensive use case, covering most of the functional aspects of the PIXEL platform. It represents a good test bench to assess effectiveness and efficiency of the proposed architecture.

In particular the following models will be exploited:

- Hinterland Multimodal Transport model;
- Environmental Management model.

Traffic prediction algorithm will be tested in order to evaluate its effectiveness according to the Hinterland Multimodal Transport model and its information needs.

Table 12: Intermodal transport trial - Hinterland multimodal transport model

Model: Hinterland Multimodal Transport [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Build any kind of port activity scenario based on past data concerning trucks and on vessel call data.	Test, validation and integration of the port activity scenario builder to generate inputs for the energy model.	Vessels' call acquired by PMIS available through DAL and IH Truck historical and future data acquired by SILI and available through DAL and IH Data about parking lot occupancy acquired by the smart camera. Results of truck arrivals prediction available in IH

Model: Hinterland Multimodal Transport [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
		Port activity scenario model previously published and deployed in OT.
Use of the port activity scenario to predict congestions, which could take place both inside the port and at the port's entrance.	For a defined vessel call list and a set of incoming trucks (predicted or scheduled) predict if a congestion will take place and how long it will involve the port's entrance	Previous features validated.
Test and validation of the model	Model will be validated using expert knowledge and data describing previously recorded congestions	Historical data about congestions and related causes.
Evaluate if SDAG should be involved, in order to host incoming trucks	For a defined vessels call list and a set of incoming trucks (predicted or scheduled) that will generate a congestion, evaluate if SDAG could be used to host some of the incoming trucks, by reducing both the size of the queue at the entrance of Port of Monfalcone and the impact on local highways	Previous features validated. Data concerning SDAG's parking lot availability through DAL and IH

Table 13: Intermodal transport trial- Predictive algorithm - Use of traffic

Predictive algorithm. Use of traffic related data [see D4.3]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Traffic volume prediction model deployed and published through OT	Integration of traffic volume prediction algorithm into PIXEL platform	Core architecture modules are already deployed
Execution of predictive algorithm	Predict traffic volume at specific time and location.	Traffic data from SILI is available through DAL and IH. Traffic volume prediction model is deployed.
Evaluation of predictive algorithm.	Compare predictions from previous feature to actual values.	Previous features available.

3.4. Port City integration trial - Port of Thessaloniki

3.4.1. Scenario and scope

The use case of the Port of Thessaloniki focuses on the interoperability of city and port in freight traffic and on the pollution impact towards the city. The expectations from THPA for the project and the execution of its use case are the following:

1. To optimize the traffic between the city and the port area and thus alleviate congestion or bottlenecks, caused by its operations.
2. To measure air and noise pollution thus being able to establish mitigation actions to reduce its environmental impact while at the same time, strengthen its relationships with the city of Thessaloniki.
3. To integrate existing systems and devices (legacy and/or new) with the PIXEL platform.

Moreover, the Port of Thessaloniki aims to gain insight about:

- Inbound and outbound traffic flows
- Overall environmental footprint of the port

In D3.4 (v2) ThPA has defined three use cases / scenarios related to the use of the PIXEL features:

1. Terminal Operator scenario (ThPA-TO-1), which aims to bring insight on the traffic related to port activities and ultimately help make operational decisions. This involves the use of the Data Acquisition Layer and Information Hub to obtain historical data on vessel calls and traffic data; and predictive models (from the Operational Tools module) to get estimations on future traffic related to imminent ship calls.
2. Environmental Manager scenario #1 (ThPA-EM-1), which aims to estimate the air pollution impact related to bulk and cargo handling activities. This involves the Data Acquisition Layer and Information Hub, to get weather and air pollution data, as well as ship calls data. Moreover, predictive models from the Operational Tools will be utilised, to get estimations

on the effect of bulk & cargo operations to the air quality in the nearby areas. These may trigger some operational decisions.

3. Environmental Manager scenario #2 (ThPA-EM-2), which aims to estimate the noise impact related to bulk and cargo handling activities. This involves the Data Acquisition Layer and Information Hub, to get noise data, as well as ship calls data. Moreover, predictive models from the Operational Tools will be utilised, to get estimations on the noise that will be generated from imminent bulk & cargo operations. These may trigger some operational decisions.

3.4.2. Pilot Plan

The Table 14: Port City integration trial - RolesTable 14 describes the responsibilities for the development of the Port-Citi integration trial for the Ports of Pireaus and Thessaloniki.

Table 14: Port City integration trial - Roles

Code:	**T7.4**			
Title:	**Port-City integration trial: Ports of Piraeus and Thessaloniki**			
Partners:			PM	ROLE
	1	PEOPLE	5	1. Task leader 2. Customise the generic PIXEL solution in order to be ready for two Pilot Trials
	2	UPV	1	Present in integration tasks and in the PEI and energy trials, conducting, namely, aspects related with the information interchange.
	3	PRO	3	Work package coordinator, collaborating in all the pilots and promoting the collaboration among the participant ports
	4	ThPA	2	Contribute to the integration of the PIXEL Information System to the existing port data sources, as part of the port-city integration and the transversal trials.
	5	PPA	-	integrate the PIXEL Information System into its port system for testing the mobility management measures and participate into inter-pilot Integration, collaboration and transversal trials
	6	CERTH	3	Contribute to the port-city integration trial at the port of Thessaloniki.
Start:	01/09/2019			
End:	30/01/2019			
Related WPs:	WP6	Enabling ICT infrastructure framework	**Batch 1: Data acquisition & Information Hub** **Batch 2: Operational Tools & Dashboard**	
	WP7	Pilot trials integration, deployment and evaluation		
	WP8	Assessment and expansion plan		
Related Deliverables:	D3.2	PIXEL Requirements Analysis	Informational	
	D3.4	Use cases and scenarios manual v2	Use cases inventory	
	D7.1	**Integration with existing systems**	**Plan to follow**	
	D8.1	Evaluation plan	What will be measured	
Other prerequisites:	**T7.1 - Phase 1**	**Connect and test all the software components developed in PIXEL in a testbed environment**		
	T7.1 - Phase 2	Effectively integrate the PIXEL Information System resources to the existing data sources in the different		

The initial Gantt diagram of the trial is:

1	Year	Who ?	2019		2020												2021
2	Month		11	12	1	2	3	4	5	6	7	8	9	10	11	12	1
3	***Phase 1: connect and test all the software components developed in PIXEL in a testbed environment***																
4	Connection with ThPA IT System (Ship Calls Data)	UPV, PRO, ThPA															
5	Connection with Weather Station (Wind Data)	UPV, PRO, ThPA															
6	Definition of Traffic Volume Data source	ThPA, CERTh															
7	Connection with Traffic Volume Data source	UPV, PRO, ThPA, CERTh															
8	Connection with Air Pollution Data source (fixed points on Port)	UPV, PRO, ThPA															
9	Definition of sensors / devices to deploy for Noise Data Collection	PRO, PEOPLE, ThPA, CERTh															
10	Optimise noise sensor placement	PRO, PEOPLE, ThPA, CERTh															
11	Connection with Noise Data source	UPV, PRO, ThPA															
12	Definition of City Noise Data source (if applicable)	PEOPLE, ThPA, CERTh															
13	Connection with City Noise Data source (if applicable)	UPV, PRO, ThPA															
14	Definition of equipment needed to deloy PIXEL (ThPA premises or other)	UPV, PRO, PEOPLE, ThPA															
15	Test of Information Hub and Data Acquisition Layer	UPV, PRO, PEOPLE, ThPA															
16	Definition of level of detail for PEI calculation of the port related data (equipment, supply chains) for connection of PAS with PEI	UPV, PEOPLE, ThPA															
17	Collection of port related data (equipment, supply chains) for connection of PAS with PEI.	UPV, PEOPLE, ThPA															
18	**Test of the Traffic Prediction Model**																
19	Traffic Prediction Model parametrisation	PRO, PEOPLE, ThPA, CERTh															
20	Preparation of input data models (Traffic Prediction Model)	PRO, PEOPLE, ThPA, CERTh															
21	Technical test and validation of Traffic Prediction Model with past data (where applicable) using Operational Tools	PRO, PEOPLE, ThPA, CERTh															
22	Link Traffic Prediction Model with NGSI agent and test of Operational Tools and Information Hub	UPV, PRO, PEOPLE, ThPA															
23	**Test of the Pollution Model**																
24	Pollution Model parametrisation	PRO, PEOPLE, ThPA															
25	Preparation of input data models (Pollution Model)	PRO, PEOPLE, ThPA															
26	Technical test and validation of Pollution Model with past data (where applicable) using Operational Tools	PRO, PEOPLE, ThPA															
27	Link Pollution Model with NGSI agent and test of Operational Tools and Information Hub	UPV, PRO, PEOPLE, ThPA															
28	**DashBoard and User Feedback**																
29	Test with end-users and feedback on the dashboard	PRO, PEOPLE, ThPA															
30	**Pilot test in real condition**	UPV, PRO, PEOPLE, ThPA															

Figure 4: Port City integration trial Thessaloniki – Gantt

The pilot is divided in four phases:

(i) phase one is devoted to connect and test all the software components developed in PIXEL useful for the Thessaloniki trial, including the integration of the data sources provided by the port.

(ii and iii) phases two and three will consist in the application of the traffic prediction model and the pollution model.

(iv) phase four regards the visualizing and monitoring all the information generated by the models used for the trial.

This is a preliminary Gantt chart and schedule may evolve since it could be subject to some delays due to integration work and WP6 developments. An updated version will be provided in the D7.2.

3.4.3. Software integration

For the purposes of the PIXEL project, THPA has developed an API, in order to share its operational data without exposing its infrastructure (as all operational servers and databases are inside the Port's LAN and not visible to the "outside world"). The added benefit has been that data shared by THPA through the API to the Data Acquisition Layer is formatted in JSON (a common format for data exchange), and accessible through simple web calls (URLs).

Data exposed through the API are coming from a number of different THPA data sources, that have been mentioned in previous documents - mainly, the Statistics application, the TOS (FRETIS) and the in-house application for gates' traffic. The API can be seen as a broker, so henceforth, any existing systems integration (in terms of data sharing) will be dealing with the API and not the application directly.

Table 15: Port City integration trial Thessaloniki - Software integration

Scenario	Software Integration	How	When
ThPA-TO-1	API --> ship calls API -> traffic (gates) Open data (see D4.2 Appendix B.1.4)	PIXEL agent connects to URL, retrieving data in JSON	Already available
ThPA-EM-1	API -> wind data Open data (see D4.2 Appendix B.1.4)	PIXEL agent connects to URL, retrieving wind data in JSON	Already available
ThPA-EM-2	API -> ship calls	PIXEL agent connects to URL, retrieving data in JSON	Already available

3.4.4. Hardware integration

At the moment of writing the deliverable, the only available sensor is the weather sensor (wind data), on Gantry Crane #3. During the phase 1 of the project plan, it will be decided if more sensors are needed.

Table 16: Port City integration trial Thessaloniki - Hardware integration

Scenario	Hardware Integration	How	When
ThPA-EM-1	API -> wind data	PIXEL agent connects to URL, retrieving wind data in JSON	Already available

3.4.5. PIXEL modules/tools to be validated in the use case

THPA is a PIXEL use-case related with environmental management. Therefore, a huge focus is placed, on the test and validation of environmental models. The main PIXEL features that are going to be tested and validated are the following:

- PIXEL Models
  - Port Activity Scenario for the supply chain modelisation
  - Air pollution model
  - Noise dispersion model
- PIXEL Predictive Algorithms
  - Prediction of road traffic
- Port Environmental Index (PEI)

- PIXEL Platform
  - Data acquisition layer: test and validation with THPA data sources.
  - Information Hub: to store and manage weather station data, traffic volume, air pollution data and noise data.
  - Operational tools: to test and validate Traffic prediction model, pollution model, PAS and PEI.
  - Dashboard: to test and validate the dashboard linked with traffic prediction, pollutions measurements and PEI.

Table 17: Port City integration trial Thessaloniki - Port Activity Scenario

Model: Port Activity Scenario (PAS) [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Build any kind of port activity scenario, based on past data and on predicted vessel calls.	Test, validation and integration of the port activity scenario builder.	Vessels' call available through DAL and IH Supply chains and machines description available through DAL and IH Results of vessels calls prediction available in IH Port activity scenario model previously published and deployed in OT.
Use of the port activity scenario as an input for environmental model (PEI) through emissions quantifications	A defined vessel call list and supply chains and machines description, generate a port activity scenario and link it with the environmental model (PEI).	Previous feature validated.

Table 18: Port City integration trial Thessaloniki . Air pollution model

Model: Air pollution model [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Air pollution model deployed and published through OT	Integration of models into the PIXEL platform	Core architecture modules already deployed
Execution of the model. Output storage and visualization	ThPA-EM-1. Environmental manager for air pollution	Input data provided by port Previous features validated.
Define different possible scenarios for air modelling, to evaluate worst/best case situations and provide recommendations for port/city	ThPA-EM-1. Environmental manager for air pollution	Worst/best case situations provided by port

Table 19: Port City integration trial Thessaloniki - Road traffic prediction

Predictive Algorithm: Road traffic prediction [see D4.3]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Visualization of current traffic status	Verification of integration of heterogeneous (open & private) data in PIXEL	Information available in IH
Prediction algorithm deployed and published through OT	Integration of predictive algorithms into the PIXEL platform.	Core architecture modules already deployed Predictive Algorithm trained, validated and packaged

Predictive Algorithm: Road traffic prediction [see D4.3]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Execution of the predictive algorithm. Output retrieval according to agreed data format.	Prediction of traffic at the port gates in the port. THPA-TO-1: Impact of the current inbounds/outbounds flow of vehicles entering/exiting the port, considering the actual traffic nearby	Previous features validated (deployed through OT) Input data available through IH
Predicted speed/volume visualization for different horizons	THPA-TO-1: Impact of the current inbounds/outbounds flow of vehicles entering/exiting the port, considering the actual traffic nearby	Previous features validated Input (updated) data available through IH
Visualization of "CGI" semaphore-like indicator	THPA-TO-1: CGI validation	Previous features validated The user can interact with the tool
User setting scales for triggering congestion condition	THPA-TO-1: CGI triggering	Previous features validated

3.5. Port City integration trial - Port of Piraeus

3.5.1. Scenario and scope

The use case of the Port of Piraeus focuses on Port-city integration: implementation of sustainable and friendly environmental measures, regarding transport demand between port and nearby city, focussing on passenger. In particular, the Port of Piraeus overall use case objective is to create a development strategy that meets the demands on transport in and around the port area, while at the same time mitigating the negative impact on the environment and on specific social groups. The PIXEL use case will lead PPA to efficiently implement sustainable, corruptible and environmentally friendly measures regarding transport demand around the port area and mobility. The use case will make use of the PIXEL Hub integrating data from the different data sources, to improve synchronization of mobility services and improve the awareness of the general public. It will enhance the PPA visibility and awareness through the PIXEL platform operational tools and will improve collaboration and logistics operators with the City and authorities.

Support air quality sensors

To be able to use air quality sensors data to forecast how noise and pollution generated by cruise and passengers ships will affect the city.

PIXEL must be able to read/receive data from air quality sensors such as:

- BTEX,
- CO,
- NOx,
- SO2,
- O3,
- PM10

Available in real-time or in batch mode (e.g.: by using TXT files exported by the air quality data logger).

Support water quality data

PIXEL must be able to store water quality data. Data are inserted manually according to the set of samplings executed every year on 40 sampling points. The analysed parameters include pH, turbidity, salinity, BOD, COD, enterococci, E. coli, total coliforms, TDS and heavy metals.

Integration with the PMIS SPARC N4

PIXEL should be able to acquire data provided by the Port Management Information System (PMIS), aimed at:

- Operation planning, such as yard planning (dynamic storage area allocation).
- Berth planning (berth allocation to vessel and crane allocation to berths).
- Ship planning (loading & unloading plan optimization, with respect to cargo compatibility and stability limitations).
- Rail-terminal operations planning.
- Dangerous cargo monitoring.
- High value cargo monitoring.

PIXEL should use information provided by SPARC N4 in order to monitor vessels and operations (both in container and coastal areas) in real time and to model impact on city and coastal areas (in terms of pollution and noise).

Integration with the Waste Management Information System

PIXEL should be able to acquire data provided by the Waste Management Information System, in order to evaluate how such system can reduce emissions of pollution and noise.

Estimate pollution impact of cruise and passengers ships related activities

PIXEL must be able to estimate the pollution impact of cruise and passengers ships related activities (bus, taxi, etc.) to the city, due to specific/bad forecasted weather conditions, for the next day. Model used for estimation must consider:

- weather data (coming from sensors or third party);

- air quality data (coming from sensors or third party);
- traffic data;
- vessels data.

PIXEL must provide, as output, a clear picture of pollution impact generated by the cruise and passengers ships related activities on the city.

Measure pollution impact of cruise and passengers ships related activities

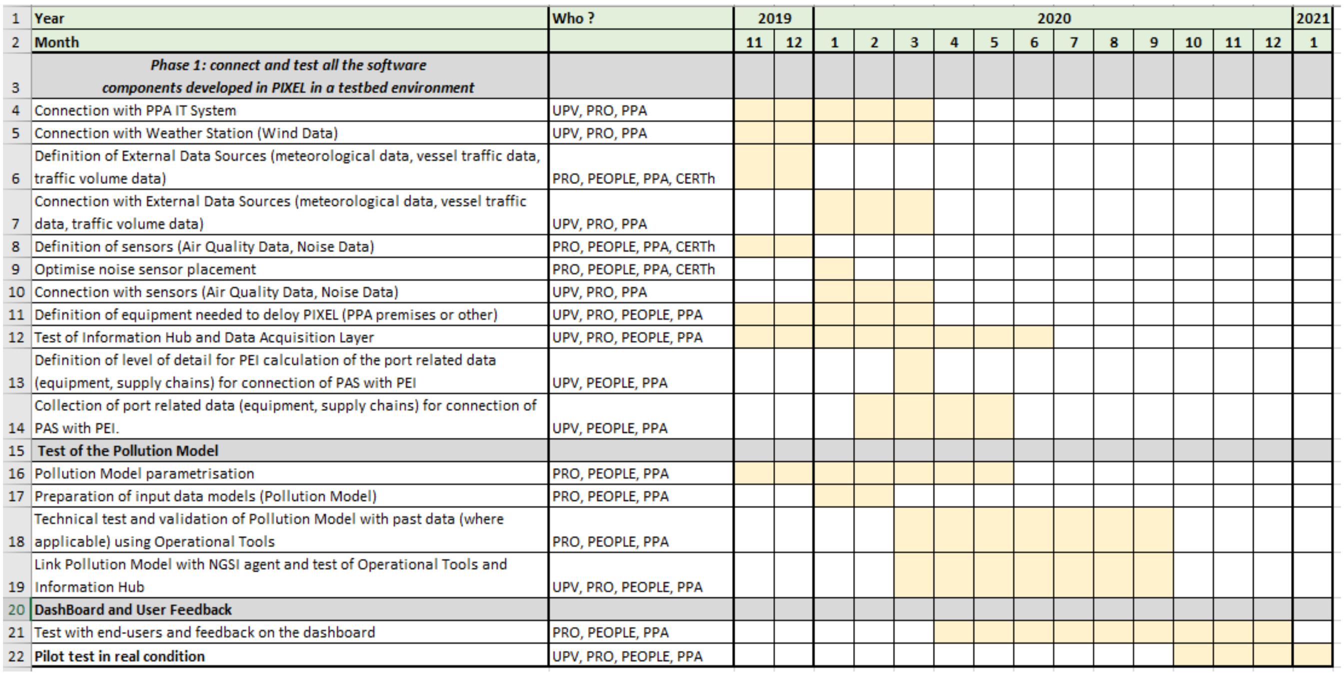
PIXEL must be able to collect data related to pollution impact of cruise and passengers ships related activities (bus, taxi, etc.) to the city, in terms of nitric oxide (NO), nitrogen dioxide (NO2), sulphur dioxide (SO2).

Such requirement allows PPA to identify the impact of cruise and passengers ships related activities and to refine forecasting models. In fact, while there are a variety of emissions sources in the PPA region, most maintain relatively constant schedules over time, with the exception of cruise ships and their associated bus traffic.

3.5.2. Pilot Plan

Table 14: Port City integration trial - RolesTable 14 (included in the Piraeus trial section) describes the responsibilities for the development of the Port-City integration trial for the Ports of Pireaus and Thessaloniki.

The following table shows the initial schedule to develop the trial.



1	Year	Who ?	2019		2020												2021
2	Month		11	12	1	2	3	4	5	6	7	8	9	10	11	12	1
3	*Phase 1: connect and test all the software components developed in PIXEL in a testbed environment*																
4	Connection with PPA IT System	UPV, PRO, PPA															
5	Connection with Weather Station (Wind Data)	UPV, PRO, PPA															
6	Definition of External Data Sources (meteorological data, vessel traffic data, traffic volume data)	PRO, PEOPLE, PPA, CERTh															
7	Connection with External Data Sources (meteorological data, vessel traffic data, traffic volume data)	UPV, PRO, PPA															
8	Definition of sensors (Air Quality Data, Noise Data)	PRO, PEOPLE, PPA, CERTh															
9	Optimise noise sensor placement	PRO, PEOPLE, PPA, CERTh															
10	Connection with sensors (Air Quality Data, Noise Data)	UPV, PRO, PPA															
11	Definition of equipment needed to deloy PIXEL (PPA premises or other)	UPV, PRO, PEOPLE, PPA															
12	Test of Information Hub and Data Acquisition Layer	UPV, PRO, PEOPLE, PPA															
13	Definition of level of detail for PEI calculation of the port related data (equipment, supply chains) for connection of PAS with PEI	UPV, PEOPLE, PPA															
14	Collection of port related data (equipment, supply chains) for connection of PAS with PEI.	UPV, PEOPLE, PPA															
15	**Test of the Pollution Model**																
16	Pollution Model parametrisation	PRO, PEOPLE, PPA															
17	Preparation of input data models (Pollution Model)	PRO, PEOPLE, PPA															
18	Technical test and validation of Pollution Model with past data (where applicable) using Operational Tools	PRO, PEOPLE, PPA															
19	Link Pollution Model with NGSI agent and test of Operational Tools and Information Hub	UPV, PRO, PEOPLE, PPA															
20	**DashBoard and User Feedback**																
21	Test with end-users and feedback on the dashboard	PRO, PEOPLE, PPA															
22	**Pilot test in real condition**	UPV, PRO, PEOPLE, PPA															

Figure 5: Port City integration trial Piraeus - Gantt

The pilot is divided in three phases:

(i) phase one is devoted to connect and test all the software components developed in PIXEL useful for the Piraeus trial, including the integration of the data sources provided by the port.

(ii) phase two consist in the application of the pollution model, using real data.

(iii) phase three regards visualizing and monitoring all the information generated by the models used for the trial.

This is a preliminary Gantt chart and schedule that may evolve, since it could be subject to some delays due to integration work and WP6 developments. An updated version will be provided in the D7.2.

3.5.3. Software integration

The integration of the PMIS of Port of Piraeus (NAVIS) with PIXEL will be determined together with the technology partners later in the project, when the exact scope and methodology of integration is decided.

The measurements coming from the existing sensors and potential new ones should be sent to PIXEL platform. Data exchange protocols need to be defined and then implemented.

PPA terminals have to plan their service activities based on many interrelated (and in many cases conflicting) factors. The physical operational procedures are rather complex and requires extensive information flows.

The cruise terminal provides an excellent example of the integration of the various Information Systems and Control Engineering applications in an overall Port Information System (PIS) architecture that incorporates the vessel traffic management, the sea yard and freight station planning operations, the administrative and financial management, the management and control of the handling activities, the cargo consolidation/warehouse services, the gate inspections and maintenance equipment.

3.5.4. Hardware integration

To reduce GHG by means of rationalizing the traffic in the cities of Piraeus and Athens around. 3% when fully deployed. This will be achieved by delivering plans to address the impact of city events and tourism, i.e. the arrival of passenger ships and cruisers, improving logistics efficiencies, minimising idle times, GHG emissions and operational costs. Real time monitoring of air quality directly operated by the port with weekly, monthly and annual reports:

- 2 sensors already implemented measuring the following parameters: quality indices (NO_2 and O_3).
- 1 sensor already implemented in testing phase measuring PM10
- 1 sensor to be implemented measuring the NO_2 and SO_2 parameters The main objectives of the monitoring are: to ensure periodic supervision, to check the tendencies, to intervene in case of alerts, to realize punctual studies generated from the raw data collected, such as Periodic air quality study and modelling by the national agency in charge of air quality monitoring.

3.5.5. PIXEL modules/tools to be validated in the use case

In the use case, PPA will demonstrate how an environmental measurement, such as PEI, can rapidly build a system and a data network using the PIXEL platform, to enable information exchange for passengers and travellers and the planning of road itineraries, allowing both logistics providers and travellers to collaborate and to plan the dispatching of orders and deliveries, maximizing the assets utilization and at the same time reducing emissions and costs.

The port of Piraeus will use PIXEL technology to set up a mobility use case. PPA will utilize the Port Environmental Index, the PIXEL Platform and its associated interface (HMI), to self-monitor and appraise of different mitigation measures, in order to define and apply effective mobility management measures in the port surrounding area, aiming at the improvement of the air quality, energy consumption, noise, relationship with local community, port development (land related) and dust KPIs. In particular, the port wishes to model, simulate or perform data analysis on the noise and air emissions from the container terminal and passenger /cruise terminal activities respectively. The noise distribution at the port of Piraeus needs to be addressed via modelling, for the following activities:

- Co-evaluate in combination with the results of LAeq indicator measurements, in order to have adequate data for noise mapping in the residential area.
- Estimate the noise level impact of the Container Terminal in the residential nearby area.
- Determination of the main sources that dominate the noise emitted during port operations.
- Overview of the possibilities to attenuate the noise sources with technical solutions.
- Assess the key noise source in the area.
- Determine the noise levels per activity and their fluctuation, during the period of study.
- Estimate the influence of the air emissions related with the port activities (cruise and passenger terminals) on the city.

The Port of Piraeus will analyse with technical partners of the Consortium, the appropriate IoT platform to be deployed, by considering the existing equipment and devices. In particular, PIXEL will enable the end-to-end visibility of the port's operations that relate to cruises operations and capacity resources, mainly focusing on passengers and barge transportation plans and execution monitoring, delivering state of the art Planning as a Service, facilitating online access to schedules and allowing PPA to accurately plan, and inform on goods and capacities, their existing and new clients. Furthermore, it will enable the development of simulation modelling tools for detailed analysis and predictions on passengers flows (per transport mode), including their emissions, port turnaround times, emissions in port operation.

PPA is a PIXEL use-case related with environmental management. Therefore, we will have a huge focus on the test and validation of environmental models. The main PIXEL features that are going to be tested and validated are the following:

- PIXEL Models
    - Port activity scenario for the supply chain modelisation
    - Air pollution model
    - Noise dispersion model
- PIXEL Predictive Algorithms
    - Prediction of road traffic
- Port Environmental Index (PEI)
- PIXEL Platform
    - Data acquisition layer: test and validation with Piraeus data sources.
    - Information Hub: to store and manage weather station, air pollution and noise data.
    - Operational tools: to test and validate pollution model, PAS and PEI.
    - Dashboard: to test and validate the dashboard linked with traffic prediction, pollutions measurements and PEI.

Table 20: Port City integration trial Piraeus - Port Activity Scenario

Model: Port Activity Scenario (PAS) [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies

PAS deployed and published through OT	Integration of models into the PIXEL platform	Core architecture modules already deployed
Build any kind of port activity scenario based on past data and on predicted vessel calls.	Test, validation and integration of the port activity scenario builder.	Vessels' call available through DAL and IH Supply chains and machines description available through DAL and IH Results of vessels calls prediction available in IH
Use of the port activity scenario as an input for environmental model (PEI) through emissions quantifications	A defined vessel call list, supply chains and machines description, generate a port activity scenario and link it with the environmental model (PEI).	Previous features validated.

Table 21: Port City integration trial Piraeus - Air pollution model

Model: Air pollution model [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Air pollution model deployed and published through OT	Integration of models into the PIXEL platform	Core architecture modules already deployed
Execution of the model. Output storage and visualization	PPA-EM-1. Environmental manager for air pollution	Input data provided by port Previous features validated.

Model: Air pollution model [see D4.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Define different possible scenarios for air modelling to evaluate worst/best case situations and provide recommendations for port/city	PPA-EM-1. Environmental manager for air pollution	Worst/best case situations provided by port

Table 22: Port City integration trial Piraeus - Road traffic prediction

Predictive Algorithm: Road traffic prediction [see D4.3]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Prediction algorithm deployed and published through OT	Integration of predictive algorithms into the PIXEL platform	Core architecture modules already deployed
Execution of the predictive algorithm. Output storage and visualization	Prediction of traffic at the port gates in the port	Previous features validated. Input data available through IH

3.6. Port Environmental Index Development

3.6.1. Scenario and scope

PEI is a composite index which aims at integrating all the environmental aspects of ports operations and their respective indicators into a single metric. The idea behind PEI is for it to be used as a metric that the small and medium-sized ports will use to address their own environmental performance. PEI is built upon significant environmental aspects of port operations which have been identified in D5.1 Environmental aspects and mapping to pilots. For a better understanding of how the PEI works, mathematical algorithms and methodology applied, see the deliverable D5.2.

3.6.2. PEI Integration

The PEI model will be applied in a similar way to the four pilots that we are developing in the project. The differences between applying the PEI model will depend on the specific data available from each port.

Table 23: Port Environmental Index Model

PEI [see D5.2]		
Features to be evaluated	Scenario	Assumptions/Dependencies
Port Environmental Index (PEI).	Use the PEI for evaluating the overall environmental impact of port activities	Sub-indexes have been computed
Sub-indexes: emission inventories (atmosphere, wastewater, noise, light, odours)	Use the Sub-indexes for computing PEI. Use the Sub indexes to assess the environmental performance in each environmental domain separately.	Raw data has been obtained through IoT sources
Raw metrics (emissions to atmosphere, noise emissions, waste production, wastewater emissions)	Analyse raw metrics. Provide some simple statistical analysis in order to summarize and drive inferences from data.	For ships: time at berth/manoeuvring, data on ship type and engines (Lloyds register). fuel type used (Lloyds register). Engine load factors (from the literature). For terminals and port authorities: Data on quantity (tonnes) of cargo moved. PAS data. Data on noise production (from sensors), waste production data (from sensors), data on wastewater emissions (from sensors), data on odours (from sensors), data on light emissions (from sensors).

4. Testing techniques description to be applied

To develop useful and quality software, the first step is to clearly define the problem that requires solution, the way to be solved and which specific functionalities the solution will have to provide to the user. In order to specify the problem, techniques of elicitation of requirements that consist of describing as clearly as possible the problem to be solved, are commonly used. On the other hand, to ensure the quality of a software, concrete test cases need to be defined, so as to verify that the software meets the requirements, both functional and non-functional. Testing work not only helps to verify that the software meets the requirements, but the benefit of testing goes beyond verification, if applied from the early stages, assisting in clarifying what is to be developed, involving end users or customers. It also provides extra valuable information for developers who have a better vision of the work to be done.

There are several software testing standards promoted by several well-known organizations:

- The International Organization for Standardization (ISO)
- The Institute of Electrical and Electronics Engineers (IEEE)
- The Capability Maturity Model (CMM)
- The Software Engineering Institute (SEI)

Many of these standards overlap and have different approaches/objectives [5]. To evaluate the quality of the developments done in other WPs (WP4,WP5 and WP6) and use cases of the Pixel project, ISTQB guidelines will be followed, based on the best practices and testing standards. The “ISTQB Certified Tester” is the most extended standardized qualification for software testers and the certification is offered by the International Software Testing Qualifications Board[1] [6] . The qualifications are based on a syllabus, and there is a hierarchy of qualifications and guidelines for accreditation and examination. The ISTQB has over 500,000 certifications issued; the ISTQB consists of 58 member boards worldwide, representing 81 countries as of May 2017.

In order to validate the different developments done, three types of test techniques, with different testing purposes, will be used:

- Individual Test (unit tests)
- Module Test (component tests)
- Integration test (integration, functional and acceptance tests)

All these test cases will be described, organized and prioritized in a test plan defined through Testlink[2] tool, as described further in this section.

The rest of this section describes the different techniques and tools that can be used to carry out the tests.

4.1. Individual Test (Unit Tests)

Unit tests are typically automated tests written and run by software developers, to ensure that a section of an application (known as the "unit") meets its design and behaves as intended. In procedural programming a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. By writing tests first for the smallest testable units and then the compound behaviours between those, one can build up comprehensive tests for complex applications.

Unit testing is the method of testing various isolated software components separately. To isolate issues that may arise, each test case should be tested independently. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation.

[1] https://www.istqb.org/
[2] http://testlink.org/

During development, a software developer may create a unit test to verify the correctness of a method of a function developed. During test case execution, frameworks log tests that fail any criterion and report them in a summary.

The procedure of writing and maintaining unit tests can be accelerated by using Parameterized Tests (PUTs). These allow the execution of one test multiple times with different input sets, thus reducing test code duplication. Unlike traditional unit tests, which are usually closed methods and test invariant conditions, PUTs take any set of parameters. PUTs have been supported by TestNG, JUnit and its Net counterpart, XUnit. Suitable parameters for the unit tests may be supplied manually or in some cases are automatically generated by the test framework. In recent years, support was added for writing more powerful (unit) tests, leveraging the concept of Theory. A parameterized test uses same execution steps, with multiple input sets that are pre-defined. A theory is a test case that executes the same steps, but inputs can be provided by a data generating method at run time.

Test and validation of PIXEL Platform Components and Models

Unit tests will be created by developers to verify that code have been written as per expectation and it will meet with all the requirement & design of the product.

This type of test will be mainly used to test the internal functionality of the different components of the Pixel Models and algorithms (WP4), PEI (WP5) and Framework components (WP6).

Depending on the programming language and the type of component developed, there is an appropriate correspondent tool to use. For instance, for java programs, JUnit will be used and for python, unittest will be more appropriate.

4.1.1. Tools

There are a lot of unit test tools and frameworks. Some of the most common and accepted ones in the software engineering industry are:

JUnit

- It is an instance of the xUnit architecture's unit testing framework for the Java programming language. It is a simple framework to write repeatable tests.
- JUnit features include assertions for testing expected results, test fixtures for sharing common test data and test runners for running tests.
- JUnit is Open Source Software, released under the Eclipse Public License Version 1.0 and hosted on SourceForge.

TestNG

- It is a testing framework for the Java programming language inspired by JUnit and NUnit.
- It includes support for annotation, parameterized, data-driven testing and multiple instances of the same test class.
- It is designed to cover all categories of tests: unit, functional, end-to-end, integration.
- Offers flexible execution model and can be run either by Ant via build.xml or by an IDE plugin with visual results.

NUnit

- Unit-testing framework for all .NET languages.
- It is one of many programs in the xUnit family, written entirely in C# and serves the same purpose as JUnit does in the Java world.
- NUnit is Open Source software, NUnit 2.6.4 is released under the NUnit license while NUnit 3.0 uses the MIT license

Unittest

- The unittest unit testing framework was originally inspired by JUnit and has a similar flavour as major unit testing frameworks in other languages. It supports test automation, sharing of setup and shutdown code for tests, aggregation of tests into collections, and independence of the tests from the reporting framework.

4.2. Module Test (component tests)

Module testing is defined as a software testing type, which checks individual subprograms, classes, libraries or components in a program.

The basic difference between the unit testing and component testing is that, in unit testing the developers test their piece of code but in component testing the whole component is tested. For example, in an air pollution station record application, there are two modules; one which registers a new pollution station and the other module, which uploads the new data read from the station. Both the modules are developed separately and when they are tested one by one, then this is referred as a component or module testing.

This type of test will be mainly used to test the general functionality of the different components of the Pixel Framework, developed under WP4, WP5 and WP6. Most of them could be used through their API.

Test and validation of PIXEL Platform Components and Models

This type of tests will be mainly used to validate the different platform modules and models in an isolated way before integrating them into the platform

4.2.1. Tools

The same tools used for unit test (Junit, testNG) can be used. Moreover, specific tool for testing APIs can be used, such as SOAPUI, one of the mostly used tools for testing APIs.

4.3. Integration test (integration, functional and acceptance tests)

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together. Normally the former is considered a better practice, since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components, corresponding to elements of the architectural design, are integrated and tested until the software works as a system.

Through this tests category we could have integration and acceptance testing:

- Integration testing

  Integration testing tests a set of components integrated to verify that components work properly together. For example, a system test might involve testing a login interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then logoff.

- Acceptance/Functional testing

  Operational acceptance is used to conduct operational readiness (pre-release) of a product, service or system, as part of a quality management system. OAT is a common type of non-functional software testing, used mainly in software development and software maintenance projects. This type of testing focuses on the operational readiness of the system to be supported, or to become part of the production environment. Hence, it is also known as operational readiness testing (ORT) or Operations readiness and assurance (OR&A) testing. Functional testing within OAT is limited to those tests that are required to verify the non-functional aspects of the system.

In addition, software testing should ensure the portability of the system, as well as working as expected, while at the same time does not damage or partially corrupt its operating environment or cause other processes within that environment to become inoperative.

Test and validation of PIXEL Pilots, Platform Components and Models

This type of test will be mainly used to test the integration components into the Pixel Framework and to validate the different use cases.

4.3.1. Tools

Integration , Functional and Acceptance test tools

Functional and acceptance test can be executed manually following a use case description. For instance, to test a form in a web site, the common way to test it, is to fulfil and execute manually the form.

But there are some tools like Katalon Studio, Cucumber, Selenium, that can be used to automate the execution of functional test. The cost of automatization of these test cases is too high and exceeds the scope of the Pixel project but if any partner wants to automate their integration tests, it can be done.

The tools previously defined for unit and module test can be used in some cases.

4.4. Test Management tool

The management of test cases is a difficult task and a tool is required to manage them. The open source Testlink web tool will be used for this purpose. An instance of the TestLink will be available for the partners to document and manage the creation, execution or reporting of the test cases.

Testlink is the most widely used test management tool. It integrates both requirements specification and Tests specification together. The user can create test projects and document test cases using this tool. A user (with permissions) can create an account for multiple users/testers and assign different user roles. Admin user can manage test cases assignment task.

It supports both manual and automated execution of Test cases. With this tool, the testers can generate Test Report and Test Plan Documents within a minute. It supports the generation of Test reports in MS Word, Excel and HTML formats.

TestLink also supports integration with many popular Defect Tracking systems like Mantis, BugZilla, Jira, Youtrack and TRAC. A specific bug ticket can be linked with test cases. It also supports and maintains multiple Test projects. Since it is a web-based tool, multiple users can access its functionality at the same time with their credentials and assigned roles.

Some of the features of the TestLink are:

- User roles and management
- Multi-project
- Grouping of test cases in test specifications
- Test plans
- Platforms
- Requirements with versioning and revisioning
- Support for testing different builds of the software
- Reports, charts and monitors
- Customization of the user interface using Smarty templates
- Integration with LDAP

- Integration with other software using a provided API
- Bug tracking system integration (Mantis, JIRA, Bugzilla, FogBugz, Redmine, and others)

Prodevelop, as a WP leader, will provide support and training to the rest of the partners of the project, involved in test activities..

4.5. Test plans

Formally, ISTQB defines a test plan as a document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques, entry and exit criteria to be used, as well as the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

This section defines the general pilot operation plan for the PIXEL implementation, specifying the required phases to be developed at the pilot sites for the implementation of the PIXEL platform.

THIS IS A GENERIC DESCRIPTION describing the fields of the test plans. Sub-sections will change according the standard followed.

As the project and test planning progress, more information becomes available and more detail can be included in the test plan. Test planning is a continuous activity and is performed throughout the product's lifecycle. Feedback from test activities should be used to recognize changing risks so that planning can be adjusted. Test planning activities may include the following and some of these may be documented in a test plan:

- Determining the scope, objectives, and risks of testing
- Defining the overall approach of testing
- Integrating and coordinating the test activities into the software lifecycle activities
- Making decisions about what to test, the people and other resources required to perform the various test activities, and how test activities will be carried out
- Scheduling of test analysis, design, implementation, execution, and evaluation activities, either on particular dates (e.g., in sequential development) or in the context of each iteration (e.g., in iterative development)Selecting metrics for test monitoring and control
- Budgeting for the test activities
- Determining the level of detail and structure for test documentation (e.g., by providing templates or example documents)

The format and content of a software test plan vary depending on the processes, standards, and test management tools being implemented. Nevertheless, the following format, which is based on IEEE standard for software test documentation, provides a summary of what a test plan can/should contain [7]

Test plan template description

The following table provides a description of the contents expected in a test plan.

Table 24: Test Plan Explanation

Id	Name	Description
1	Test Plan identifier	• Provide a unique identifier for the document

Id	Name	Description
2	Introduction	• Provide an overview of the test plan. • Specify the goals/objectives. • Specify any constraints
3	Object to be test	• List the test items (software/products) and their versions.
4	References	List the related documents, with links to them if available, including the following: • Project Plan • Configuration Management Plan
5	Features to be tested	• List the features of the software/product to be tested. • Provide references to the Requirements and/or Design specifications of the features to be tested
6	Features not to be tested	• List the features of the software/product which will not be tested. • Specify the reasons these features will not be tested.
7	Testing Strategy	• Mention the overall approach to testing. • Specify the testing levels [if it's a Master Test Plan], the testing types, and the testing methods [Manual/Automated; White Box/Black Box/Gray Box]
8	Acceptance and Fail criteria	• Specify the criteria that will be used to determine whether each test item (software/product) has passed or failed testing.
9	Suspension Criteria and Resumption Requirements	• Specify criteria to be used to suspend the testing activity. • Specify testing activities which must be redone when testing is resumed.
10	Test Deliverables	• Test Reports
11	Test Suites	• Test cases • Test suite
12	Environment needs	• Specify the properties of test environment: hardware, software, network etc. • List any testing or related tools.
13	Responsibilities, obligations	List the responsibilities of each team/role/individual.
14	Personnel and training needs	• Specify staffing needs by role and required skills. • Identify training that is necessary to provide those skills, if not already acquired.
15	Schedule	Provide a summary of the schedule, specifying key test milestones, and/or provide a link to the detailed schedule.
16	Planning risks and contingencies	• List the risks that have been identified. • Specify the mitigation plan and the contingency plan for each risk.

Id	Name	Description
17	Approval and authorisation	● Specify the names and roles of all persons who must approve the plan. ● Provide space for signatures and dates. (If the document is to be printed.)

Pilots' test plan filled in:

The following table contains the test plans for the four pilots to be deploy at ports, which have been described in section 3. Since there are quite similarities between them and for that reason is better to create a unique test plan for all of them.

Table 25: Pilots test plan

Id	Name	Description
1	Test Plan identifier	● TP1 Energy pilot ● TP2 Intermodal pilot ● TP3 Port City pilot THPA ● TP4 Port City pilot PPA
2	Introduction	This document describes the test plan to be applied to ensure quality of the pilots development
3	Object to be test	• TP1 - Bordeaux Pilot • TP2 - Monfalcone Pilot • TP3 - Thessaloniki Pilot • TP4 - Piraeus Pilot
4	References	List of deliverables useful for the development of pilots: D3.2, D3.4, D4.3, D5.2, D6.2
5	Features to be tested	List of deliverables useful for the development of pilots: D3.2, D3.4, D4.3, D5.2, D6.2
6	Features not to be tested	The pilots will only test the integration / acceptance test of the platform. Only the validation of the integration of the platform (Tack 7.1) will take care of executing the unit and module tests
7	Testing Strategy	Section 4 of this document
8	Acceptance and Fail criteria	Acceptance: The models proposed in the requirements (D3.2 and D3.4) should work for each pilot using real data from ports. Fail: If the models cannot run properly using real data from ports.
9	Suspension Criteria and Resumption Requirements	If the Pixel platform does work properly (does not provide data), models cannot be executed or blocking exception is raised, the testing procedure will be stopped.
10	Test Deliverables	Test reports generated by Testlink tool

Id	Name	Description
11	Test Suites	A test suite has been created for each pilot using Testlink tool
12	Environment needs	This information has been defined in D6.2
13	Responsibilities, obligations	• TP1 - Bordeaux Pilot ○ Technical Responsible: Catie ○ Data Responsible: GPMB ○ Acceptance Test: GPMB • TP2 - Monfalcone Pilot ○ Technical Responsible: Insiel ○ Data Responsible: SDAG ○ Acceptance Test: SDAG • TP3 - Thessaloniki Pilot ○ Technical Responsible: iPeople ○ Data Responsible: THPA ○ Acceptance Test: THPA • TP4 - Piraeus Pilot ○ Technical Responsible: iPeople ○ Data Responsible: PPA ○ Acceptance Test: PPA
14	Personnel and training needs	• Model providers → will provide documentation and training about the models to the technical responsible of each pilot • Technical responsible of each pilot → will provide documentation and training about the platform to the specific port. A concrete set of documentation and training will be provided in D7.2
15	Schedule	• TP1 - Bordeaux Pilot plan (Section 3.2.2 of this document) • TP2 - Monfalcone Pilot plan (Section 3.3.2 of this document) • TP3 - Thessaloniki Pilot plan (Section 3.4.2 of this document) • TP4 - Piraeus Pilot plan (Section 3.5.2 of this document)
16	Planning risks and contingencies	The main important risks are: - The lack of data → use of simulated data in case it is needed - Different data for each port → definition of data models to generalize - Integration problems → modular platform with open APIs
17	Approval and authorisation	The responsible of the ports with the technical leader of the project will be in charge or validating the pilot

To define the test cases for each development a test case template has been defined. This template fits with the TestLink tool that provides forms to create test cases.

Table 26: Test Case Template

Id Scenario	
ID Requirement	
Jira Ticket	
Name	
Summary	
Prerequisites	
Steps	<table><tr><td>Nº</td><td>Step action</td><td>Expected result</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>

Where:

- **Id Scenario** : the identifier (or name of the scenario)
- **Id Requirement:** the identifier of the requirement to be tested in this test case
- **Jira Ticket:** the identifier of the related task on Jira
- **Name:** name of the test case
- **Summary:** description of what the test case does
- **Prerequisites:** if there is anything special before starting the execution of the test
- **Steps:** the different steps that should be followed by the user to verify if the functionality is working properly or not

Example of a requirement to be tested

To create a test case to validate the WEB-UI non-functional requirement defined in D3.2

Table 27:Web UI Requirement

Web UI [100]
PIXEL must provide, for each of its tools, a Web based User Interface. In particular, in order to be used by as many stakeholders as possible, PIXEL Tools must be accessible by means of a standard web browser, without requiring any additional software to be installed on operator's computer. A Web UI must be developed by considering, at the same time: • adoption of web standards (e.g.: HTML, CSS, Javascript); • portability on different devices (e.g.: responsiveness); • readability; • easy to use.
Acceptance criteria: a WEB UI is available for each PIXEL's tool.
MoSCoW Priority: MUST HAVE
Category and Type: Non-functional – Look and feel
Product: PIXEL platform
Use-case: Generic
Scenario: GENERIC
Customer Satisfaction: 3

Test cases created for the requirement

Several tests were created using Testlink tool to validate that the UI works with different resolutions. For this specific requirement, four test cases have been defined: small screen, medium screen, big screen and default tablet resolution. All these tests have been grouped by a Test suite.

Test Suite : WEB-UI
PIXEL must provide, for each of its tools, a Web based User Interface. In particular, in order to be used by as many stakeholders as possible, PIXEL Tools must be accessible by means of a standard web browser, without requiring any additional software to be installed on the operator's computer. A Web UI must be developed by considering, at the same time:

- Adoption of web standards (e.g.: HTML, CSS, Javascript);
- Portability on different devices (e.g.: responsiveness);
- Readability
- Easy to use.

The following figure shows how the template should be completed for a specific Test Case.

Test Case fw-31: Responsiveness - Resolution Screen-Medium1920-1080 **[Version : 1]**			
Summary: Check that the application displays the information in a proper and usable way. Check if there is any overlap or information not reachable			
#:	Step actions:	Expected Results:	Execution Status:
1	the user access to the dashboard	the home page is shown properly	
2	the user login into the system	the login page is shown properly	
3	The user access to all the functions available through the menu	all the pages are presented as expected.	
Execution type:	Manual		
Estimated exec. duration (min):			
Priority:	Medium		
Jira tickets:	PIX-100		
Execution Details			
Execution Result	**Not Run**		

Figure 6: Test case described using the test case template

The following figure corresponds to a screenshot of a test cases defined through the TestLink tool.

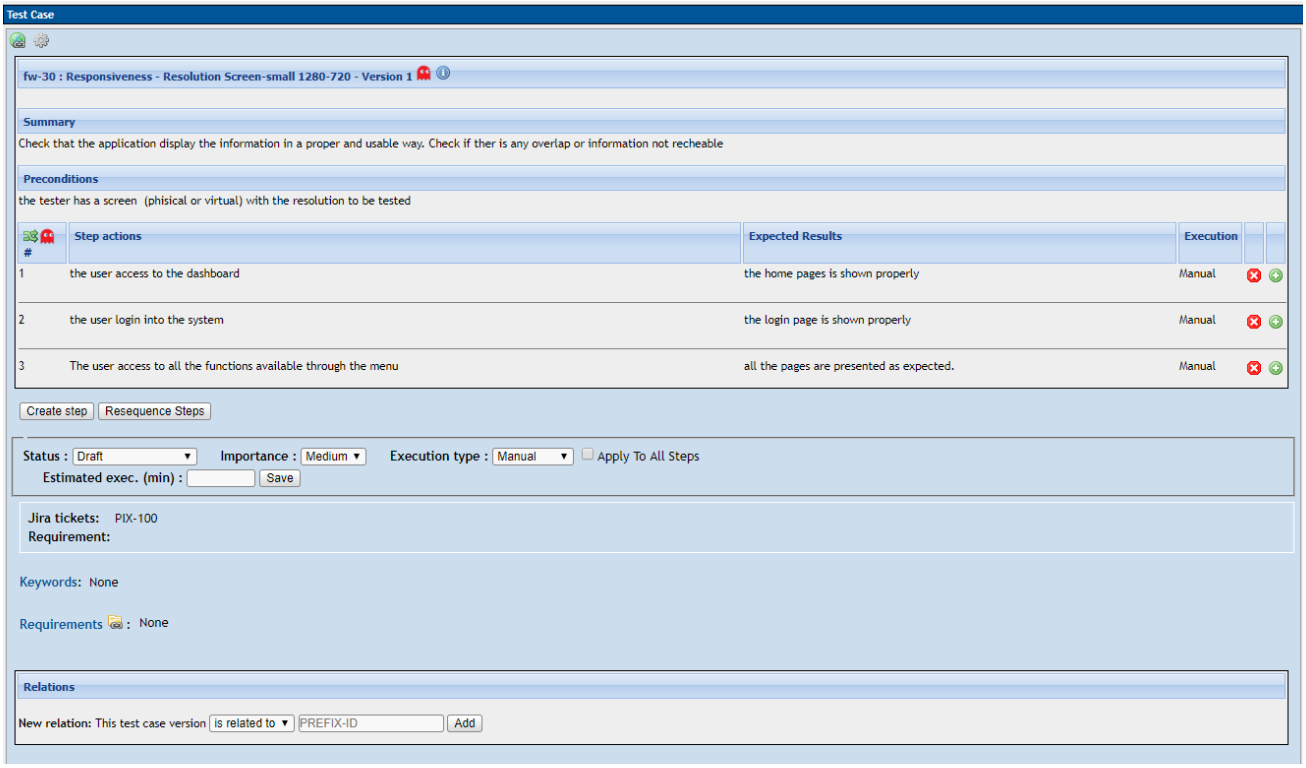


Figure 7: Test case defined using TestLink tool

5. Pilot and component test cases description

5.1. Testlink tool

Testlink tool is being used to manage the test cases and test suites mostly related with the integration/acceptance tests. The URL to access the TestLink web is "http://pixel-testing.prodevelop.es/testlink/login.php", through this page (Figure 8), which allows the user to login into the application. It is required for someone to be registered, in order to have access to the web. A user has been created for review purposes (user: "reviewer" , pass:"dfg$&3TL6,").

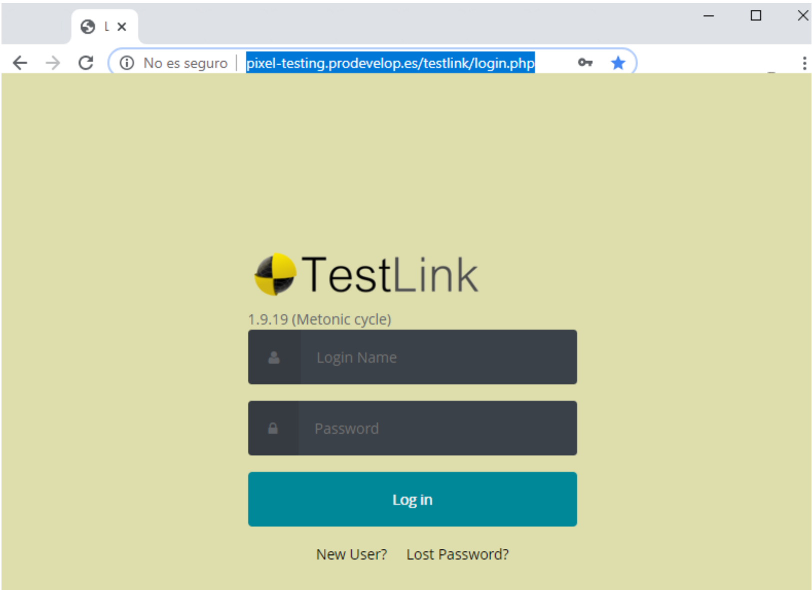


Figure 8: TestLink login page

After login into the platform, the main page is showed with a list of options/functionalities. The user should select the project in which he is going to work (Figure 9).

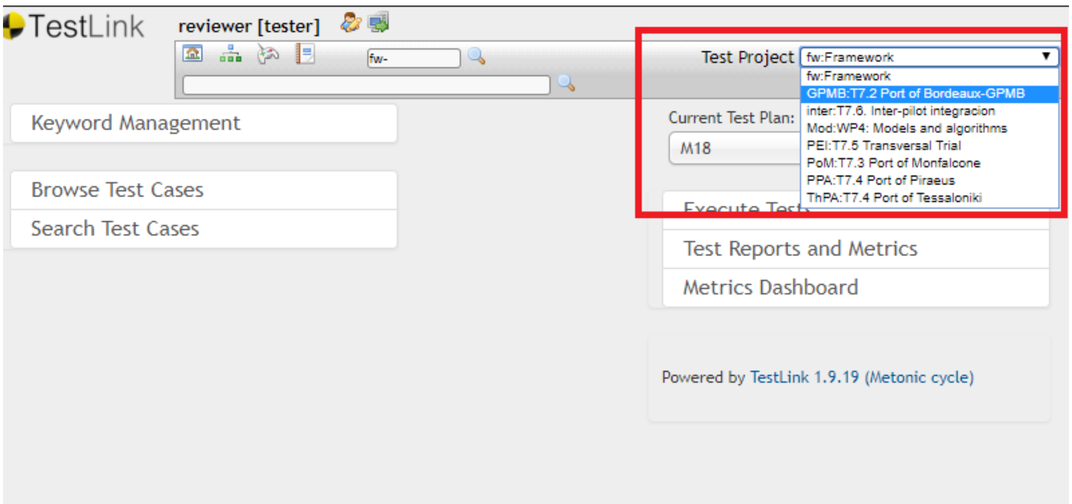


Figure 9: TestLink Project Selection

Once the project has been selected the user has access to the test suit of the project that contains a folder structure of test cases (Figure 10).

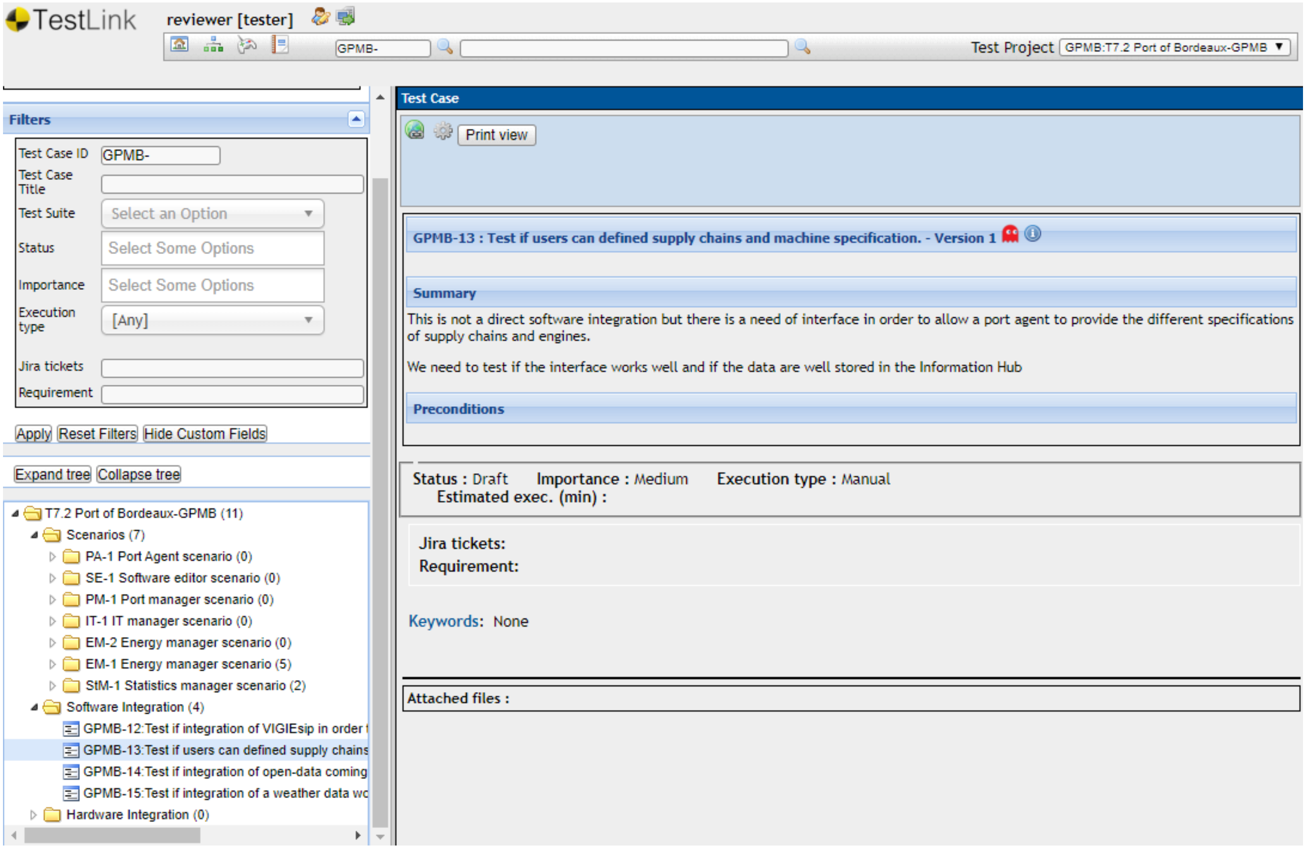


Figure 10: TestLink Test suit

5.2. PIXEL Platform

In WP4, PIXEL models have been fully described, developed and validated using some sample data. The models were developed based on the reality of port activities and as close as possible to the practices. Thus, the modeling steps use common sense and business expertise. The modeling choices that have been made, have therefore been validated by the port partners. In addition, environmental pollution models are widely validated and verified models in literature. Thus, for almost all PIXEL models, the test and validation phase will consist in the correct parameterization of the models, according to the use-case considered and the data available.

In WP6, PIXEL platform has been described and developed. To test the correct functionality of the different components and modules of the platform (Data Acquisition Layer, Information Hub, Operational Tools, Dashboards and security), these components should be tested in an isolated mode, using Module Test. The choice of specific techniques and tools to be used, depends on how these components are available through the platform (how they are exposed). Some of these components are tested through a JSON API, while others are tested as an embedded library.

5.2.1. Models and algorithms

5.2.1.1. Port activity scenario modelling

The PAS model has been developed based on the available data-sources and information about ports process for cargoes' manipulation between ships and yards (both directions). For every datatype (input, parameters, output), a data-model has been defined into the form of a JSON document (JavaScript Object Notation), described in deliverable D4.2 and available on PIXEL's git. The Figure 11 shows the data flow across the model's modules. The model's input, which refers to the list of cargoes' handlings to be processed, is provided when the model is called. Then, several modules use parameters to build the PAS. Those parameters correspond to port's settings, describing process and physical entities. From those data, the PAS builder provides an intermediate output that lists every required atomic operation, and its arrangement across time. Then successive PAS model's modules provide the corresponding outcomes on energy consumption and pollutants emissions. These additional pieces of information are added to each atomic operation to produce the PAS as a final output.

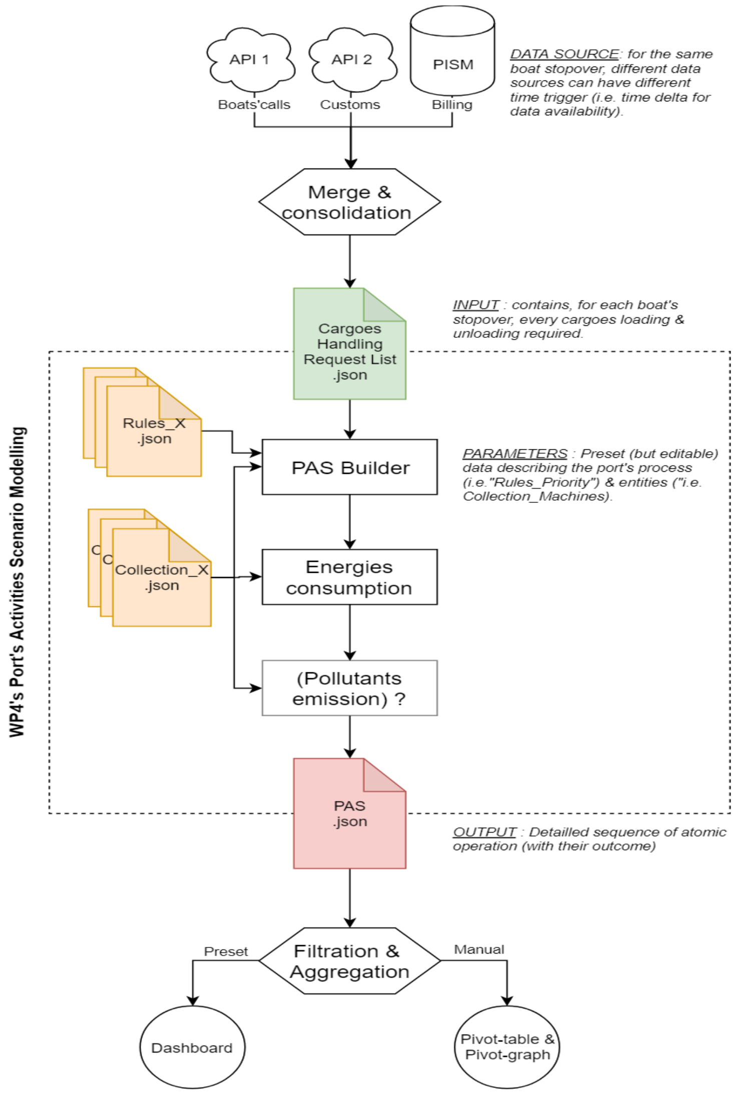


Figure 11: Schematic diagram of the model and its data flow

A series of tests have been identified for the PAS model.

1. For every data-document, the following points have to be tested:
   - creation and compliance with the data-model,
   - storage and transmission across the platform (between every WP6's component and every PAS model's module),
   - viewer and editor suitability for end users.
2. For every PAS model's modules, the following points have to be tested:
   - unit test for input data fault tolerance,
   - output significance and correctness,
   - suitable instantiation and termination, without side effect on data-documents.

3. For every tool used for PAS restitution, the following points have to be tested:
   - completeness, meaningless and usability of dashboard and data exploration tool,
   - automatic notification functioning,
   - data import and export functioning.

5.2.1.2. Energy models

For testing the energy demand model in real conditions, two different approaches will be used. First, the opinion of several experienced port operators will be requested. To do so, each expert will setup an arbitrary port activity scenario (corresponding to his/her cargo handling expertise) and give an appreciation to the module output. Second, the module output will be confronted to effective measurement. As an example, to quantify the whole activities' electricity consumption modeling, a PAS corresponding to actual activities during a billing period can be used. The ratio of modeled consumption versus actual billed amount can be used as an accuracy quantification tool. Note that GPMB will be able to provide soon, aggregated electricity consumption data in real time. This could allow a thorough inspection of the model's accuracy. In particular to study the results of the model on a fine temporal resolution such as an hour.

The PVGIS model is already fully tested and validated so there is no need to do it again. The predictive algorithm for prediction of PV system power will be validated inside WP4 (cross-validation, comparison with real measurement, …).

5.2.1.3. Transport models

There are 2 potential scenarios for testing and validating "the intermodality transport sub model" in real conditions:

- ***Scenario 1***: "Steel slab" arriving to Monfalcone's port by shipping. Slabs are handled by cranes and put into trucks. Trucks are driven by roads form Monfalcone Port to "Aussa Corno" industrial district (almost 40 km away) passing through several small and medium villages creating an increase of traffic jams and potential accidental crashes (social security decrease). The most important effect is the heavy weight of the cargo, which is heavily damaging road infrastructure during truck's trip, causing very high social costs.
- ***Scenario 2***: "Steel slab" arriving to Monfalcone's port by shipping. Slabs are handled by cranes and put into a cargo train. Trains are driven by rail from Monfalcone Port to Monfalcone urban train station (10 km), where the switch between diesel locomotives to electric locomotive takes place. From Monfalcone urban train station, trains are driven to "Aussa Corno" industrial district (almost 30 km) without passing through several small and medium villages, avoiding adding to congestion, traffic jams and potential accidental crashes (security decrease).

These two scenarios will be used to define model parameters. Moreover, in order to evaluate in the best possible manner, the pollution created by trucks in terms of CO, NOX, PM, emissions and compare it with that produced by diesel locomotives, the Copert method is going to be used, to calculate trucks emissions and Tier 2 EMEP/EEA emissions factors for "line haul locomotive", in order to calculate diesel locomotive emissions (https://www.emisia.com/utilities/copert/documentation/, interpolate data from experimental tests).

Concerning the "traffic sub-model" as defined in D4.2, the main integration test will consist in defining well the model parameters with real data. These parameters are :

- Planned arrivals and departures of vessels;
- Planned arrivals and departure of trucks;
- Real time parking occupancy;
- Real time situation on port's gates;
- External events which may impact on port truck rate.

When the model becomes correctly parameterized, test and validation of this “traffic sub-model” will be performed by comparing modelling results with the observed reality.

5.2.1.4. Environmental pollution models

Both noise and air dispersion models are fully tested and validated in literature. For this model the main integration test will consist in defining the good input data values for each use case.

5.2.1.5. Predictive algorithms

Vessel call predictive algorithm will be assembled from FAL forms and possibly AIS data, to compensate for the missing data in the FAL forms. An attempt will be made to discover patterns that could help predict future trends in the port, in terms of cargo type, volumes or even a specific ship, especially in cases of a regular line, well in advance, out of historical data. This will present a higher-level view on the evolution of the trends in the port (cargo volume, ships, types of cargo, seasonality of the cargo, etc.). For testing and evaluation of prediction algorithms, predictions will be compared to the ground truth in FAL forms. For evaluation of model, multiple pairs of training and test data sets will be generated out of historical data, that was made available by certain ports, the most comprehensive one by GPMB. Different error metrics will be calculated, according to the prediction objective:

- **Numeric:**
  - Mean Absolute Error,
  - Root Mean Squared Error,
  - Mean Absolute Percentage Error.
- **Categorical:**
  - F1 Score,
  - Confusion Matrix.
- **Time:**
  - Convert to time difference in seconds, minutes, hours or days and use numeric error metrics.

Vessel calls duration predictive algorithm will be tested on two ways. First, predictions will be compared to actual durations in historical data (on a multiple pair of training and test data set combinations). Secondly, predictions will be compared to the predictions that are estimated on-line by the port authorities, on time of ship’s entry.

When comparing algorithms' predictions to actual durations in historical data, multiple data set pairs will be made. Pairs will consist of training and test data sets and will be generated using forward-chaining and leave-one-year-out methods (a variation of cross-validation). To evaluate the algorithm's performance, mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) will be calculated. Domain experts from the port will tell, if error values are acceptable. MAE is an easily interpretable indicator, that will provide insight on how many hours on average, predictive algorithm misses with its estimation. RMSE is not directly correlated to error in hours. But its advantage is, that higher error values, have a much higher impact on error metric’s value. Advantage of MAPE metric is, that difference between prediction and real value is taken relatively to the real value. This means that the same absolute error on lower real value, will be reflected as higher error, than the same absolute error on higher real value.

Port authorities’ estimations for the time of departure will be collected from VIGIEsip API, which allows live predictions, at the time of vessel’s arrival. Both predictions will be compared to the actual time of departure, after vessels leave the port.

Short term ETA predictive algorithms will be tested on the historical AIS data that will be collected in PIXEL Information Hub around ports and regions of interest to PIXEL project, as well as on the large-scale data from other regions and data sources (i.e. Marine Cadastre). Training and test data set pairs will be generated with leave-one-arrival-out method (a variation of cross-validation). Each arrival is assembled from multiple AIS messages that are sent along the way to the port. Different error metrics will be calculated, such as MAE and RMSE. Visualizations will also be prepared, that will represent error metric in correlation with distance to the terminal.

Satellite imagery - vessel detection algorithm will be tested with AIS data on the satellite imagery from Sentinel-2 and Planet Dove. Detections, in a form of bounding boxes will be compared to the ground truth vessels reported positions in AIS data. Results will be presented as the percentage of correctly retrieved ships. Method will be evaluated on the data that was collected and automatically labelled, with AIS data from Marine Cadastre and satellite imagery around regions of Port of Long Beach and San Francisco Bay area. We will also report retrieval metrics across different ship lengths.

Short-term traffic volume prediction algorithm will be evaluated on historical data from SILI system, ThPA internal data and the data that will be collected from the external provider for PPA. Data will be aggregated to different time intervals, ranging from a few minutes to multiple hours. Using forward-chaining method (a variation of cross-validation), multiple training and test data set pairs will be made. Training data set will get larger at each iteration, while test data set will be of the same length, as the forecasting horizon. Reasonable horizon will be determined in the testing process, while different domain related standard error metrics, will be calculated (MAE, MSE, RMSE, MAPE). Results will be also compared to simple baseline predictions, such as moving averages and average value on the same day and at the same time in the past etc.

Prediction of renewable energy production will be evaluated based on historical irradiance data and associated weather conditions, obtained either by measurement or by satellite-based tools (PVGIS). This prediction will in fact reflect a typical day, week, month, year based on past data. The main work here will be to interact with web-services like PVGIS to obtain historical data and extract a typical irradiance. Predictive algorithms will be also developed based on historical production data, mainly obtained as open data. Moreover, predictive algorithms will be implemented for photovoltaic production, based on past data of real production and associated weather conditions. Even if the models are not directly designed for an in-site PV system, the full methodology will be reproducible. Since there is access to real data, the predictive results will be compared with real data. The objective here is to provide to the port a tool, to estimate their production based on their real installation, whenever the installation is available.

5.2.2. Data acquisition layer

The Data Acquisition Layer is composed of three different parts:

- The core composed of FIWARE modules.
- NGSI Agents develop by the FIWARE Community.
- NGSI Agents develop for the PIXEL project.

For the unit tests and module tests of external development, FIWARE quality strategy will be followed, to provide the tests quality required for the PIXEL project.

For example, for the Context Broker Orion, FIWARE provides a complete testing process for END-TO-END and UNIT TEST: https://github.com/telefonicaid/fiware-orion#testing

5.2.2.1. Unit tests for NGSI Agents developed for PIXEL

Different languages are used to write the different NGSI Agents, depending on the partner developing it. But each NGSI agent has to be delivered with the appropriate unit tests suite developed, using a framework adapted.

For example, for our NGSI Agents written in Python, unit tests are operated thanks to the Pytest framework, which is a mature unit-testing framework for Python (Figure 12)

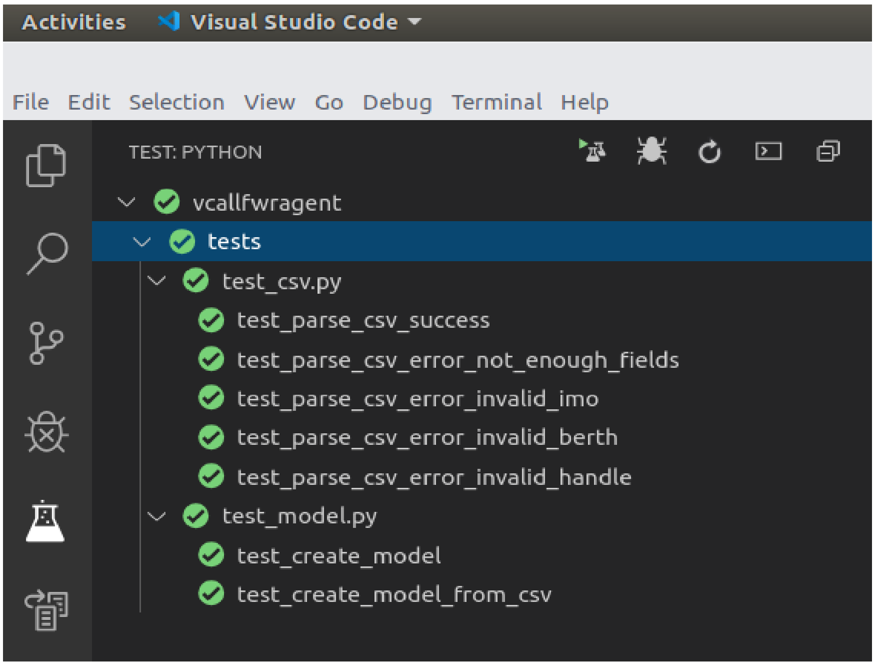


Figure 12: Pytest Unit Test

Tests can be run by hand inside the developer IDE or automatically by CLI.

```
(venv) ~/d/p/p/vcallfwragent master> pytest -v
================================================================= test session starts ==================================================================
platform linux -- Python 3.6.8, pytest-5.0.1, py-1.8.0, pluggy-0.12.0 -- /home/fieb7512/dev/pixel/pixel-acquisition/vcallfwragent/venv/bin/python3
cachedir: .pytest_cache
rootdir: /home/fieb7512/dev/pixel/pixel-acquisition/vcallfwragent
collected 7 items

vcallfwragent/tests/test_csv.py::test_parse_csv_success PASSED                                                                                   [ 14%]
vcallfwragent/tests/test_csv.py::test_parse_csv_error_not_enough_fields PASSED                                                                   [ 28%]
vcallfwragent/tests/test_csv.py::test_parse_csv_error_invalid_imo PASSED                                                                         [ 42%]
vcallfwragent/tests/test_csv.py::test_parse_csv_error_invalid_berth PASSED                                                                       [ 57%]
vcallfwragent/tests/test_csv.py::test_parse_csv_error_invalid_handle PASSED                                                                      [ 71%]
vcallfwragent/tests/test_model.py::test_create_model PASSED                                                                                      [ 85%]
vcallfwragent/tests/test_model.py::test_create_model_from_csv PASSED                                                                             [100%]

=============================================================== 7 passed in 0.04 seconds ===============================================================
```

Figure 13: Unit tests execution through CLI

In addition, a docker-compose file is available to the developer.

With a single command, it is possible to setup and run a Fiware platform as Docker containers on the development machine and then providing the ability to test the module in a more advanced manner.

5.2.2.2. Module tests

For the Data Acquisition Core module, we rely on functional tests provided by FIWARE.

To test that the NGSI agents are able to correctly handle data insertion on the Context Broker Orion, a docker-compose environment is provided with a set of data, to run the agents and check the data is well inserted on Orion.

5.2.2.3. Integration tests

As the Data Acquisition Layer is the entry point of data into the PIXEL platform, it is really important to provide integration tests, in order to ensure that modules could "speak" to each other.

When it is deployed, the context broker exposes a simple get version API request, that allows to check if it is working well. Also request the entities list can occur, to ensure it can communicate with its database.

Each NGSI agent developed for the PIXEL project has to expose two API requests:

- Get Version: to ensure the agent is working
- Get Status: to ensure the agent can communicate with Orion and if applicable to its data source.

5.2.3. Information Hub

The PIXEL Information Hub consists of the following components:

- Data Collector
- Data Writer
- Data Reductor
- Data Extractor
- System management components
- Storage components (Elasticsearch)
- Data broker (Apache Kafka)
- Config Storage (Zookeeper)

For the unit tests and module tests of external components the QA of those 3rd party component developers, will be used. Elastic, Apache Kafka, Docker and Zookeeper are all well-established projects with strict QA procedures for software releases. For the PIXEL Information Hub, only stable releases of those components are used.

5.2.3.1. Unit Tests

PIXEL Information Hub components are developed in Java and JUnit test framework, which is being used to develop tests for specific PIXEL IH components. Tests are incorporated in Apache Maven build scripts, which is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Maven Surefire[3] plugin is used to automatically run JUnit tests during each project build. This allows fully automated unit testing during each project build, thus ensuring correct functioning of single classes at each development stage.

5.2.3.2. Module Tests

To test the IH as a module Data Collector and Data Extractor REST API will be tested. A set of invocations per each API endpoint will be executed to check the results (success and failures). Data

[3] https://maven.apache.org/surefire/maven-surefire-plugin/

Collector tests will simulate insertion of data in the IH, while Data Extractor tests will verify that those datasets have been correctly written in the storage and subsequently retrieved.

5.2.3.3. Integration Tests

Integration tests as perceived by the PIXEL Information Hub implies links with the following core modules (see D6.1, Figure 15: Global architecture):

PIXEL Data Acquisition Layer (DAL): DAL is the primary data source for PIXEL. It aims at providing software mechanisms to enable appropriate data acquisition in the different port areas and through logistic agents. In principle, all data collected through DAL should be stored in the IH either to be used for immediate consumption or as a historical data storage. Technically, the link between DAL and IH is implemented through a subscription to the Orion Context Broker.

The integration test should verify that data collected through DAL is correctly received by the IH.

PIXEL Integrated Dashboard & Notifications: Data received through DAL and output data from model executions will be stored in the IH. The data will then be retrieved by the dashboard for visualization purposes.

The integration test should verify that data stored in the IH is correctly retrieved and displayed in the Dashboard component.

Operational Tools (OT): Models and predictive algorithms are to be fed with data from the IH, while a similar approach applies for the results of executions when they are scheduled and represented in a time series mode. Basically, what needs to be tested is whether the Operational Tools are able to read data from the IH before executing models, and write to the IH the result (or part of it) form the executions.

More detailed information will be provided via the Testlink tool.

5.2.4. Operational tools

The Operational Tools component is composed of various parts (see deliverable D6.2):

- The OT UI: serves as UI to use and test the API gateway.
- The API gateway: REST API to perform all functionalities assigned to the Operational Tools.
- The publication component, able to publish, deploy, edit and delete models and/or predictive algorithms.
- The Engine, able to run real time or scheduled models and/or predictive algorithms, which have been previously published and configured.
- Event processing, able to configure and handle some events related to Operational Tool management, with a bridge for using other external backend services (ElastAlert) used in PIXEL.
- Data processing and database: internal components.

Figure 14 depicts how the different test types (unit, module, integration) map to the architecture of the Operational Tools (as for D6.2)

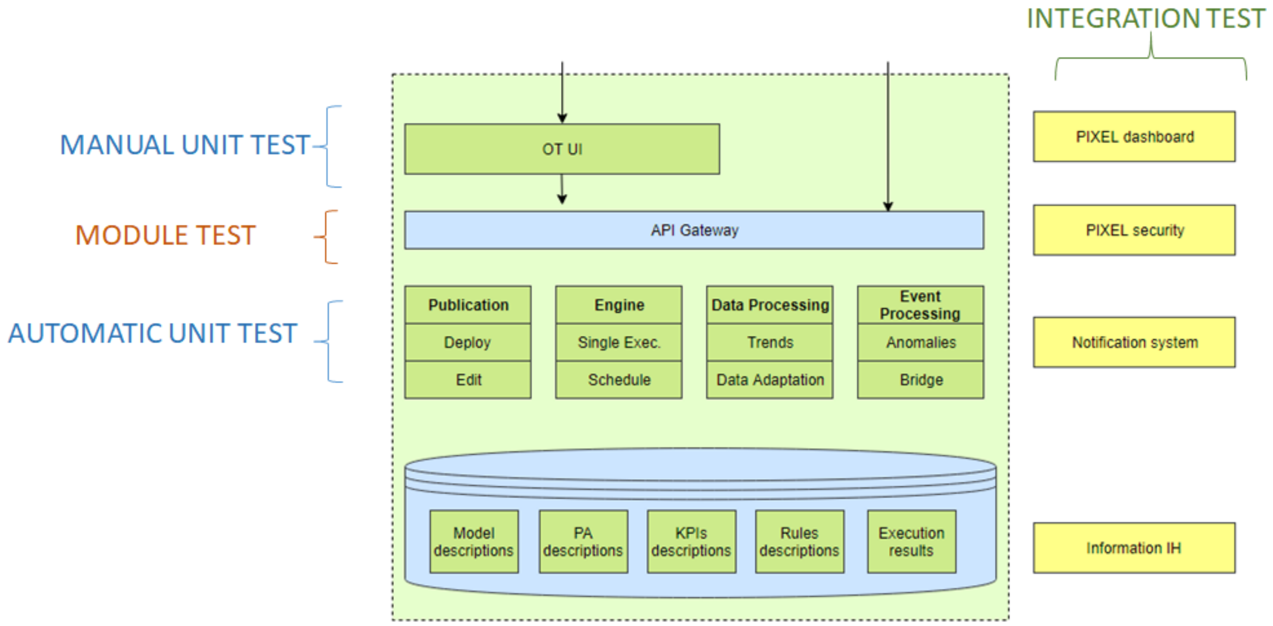


Figure 14: Operational Tools Test strategy

5.2.4.1. Unit tests

Internal components in the Operational Tools are developed in Java and JavaScript, and for each of them, unit test will be defined and implemented. However, for automatic unit tests, it makes more sense to focus on backend components, whereas manual unit test will apply to the UI, where the visualization is to be checked and requires human intervention.

Currently the developments for all internal modules are not finished and this impacts the unit tests, which will evolve consequently. However, some examples will be exposed here in order to illustrate the process.

The backend publication service for models and predictive algorithms is implemented in Java and some unit tests have already been defined. Such tests are part of the code and can be easily found with a programming IDE such as Eclipse, as depicted in Figure 15. It shows examples for checking reading and writing operations in different databases (internal memory, Mongo).

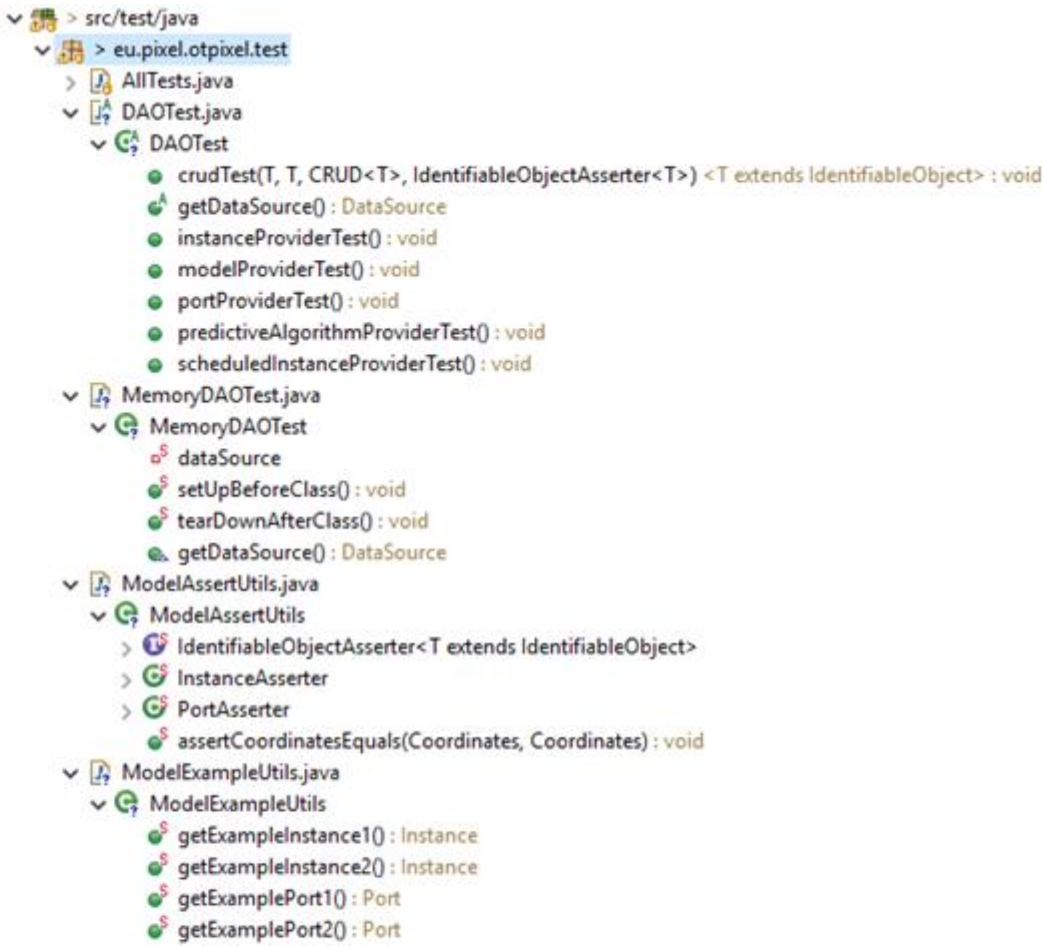


Figure 15: Unit test for the publication of models

The tests can be run by hand inside the programming IDE. Eclipse has a Junit module to run the test, as depicted in Figure 16.

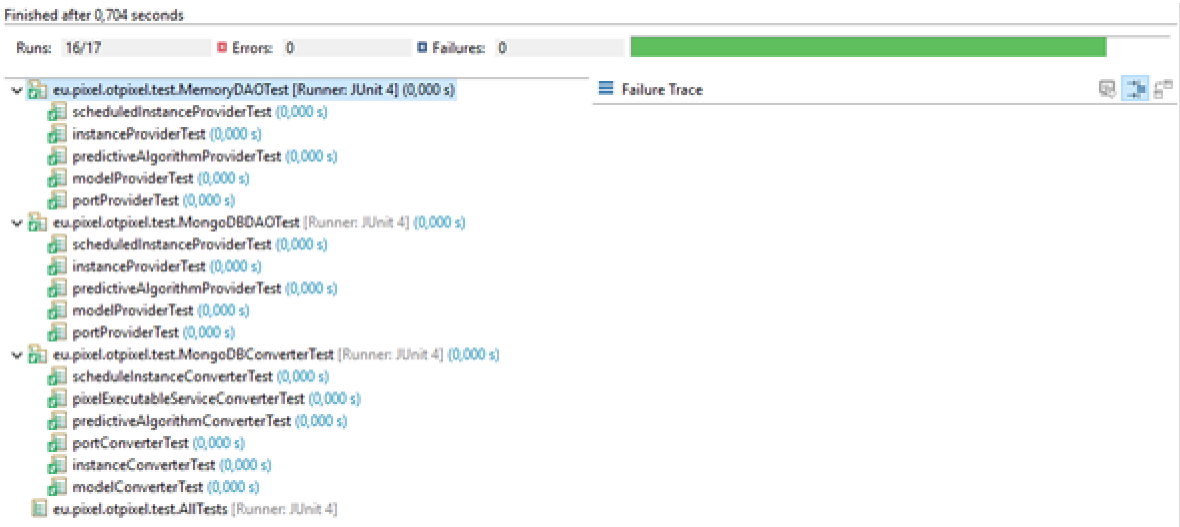


Figure 16: Junit execution within Eclipse IDE

Another option, more integrated within CI environments is the usage of Maven, which incorporates plugins such as surefire, able to use different testing providers (TestNG, Junit and POJO). Maven can be launched through the IDE or the console. It can be included into the deployment cycle, and take actions if some unit tests are not satisfied (e.g. do not deploy a module if some test is not passed).

5.2.4.2. Module tests

To test the Operational Tools as a module the API gateway will be tested. It has been developed as a REST API with a Swagger (Open API) interface, so a set of invocations per each defined request, will be executed to check the results (success and failures).

In order to monitor general availability, two methods will be provided:

- Get Version: to ensure the module is working
- Get Status: to ensure the module can communicate internally with its internal database

The module test will involve checking all methods exposed through this API, for the Operational Tools, while a Swagger interface will be provided. The documentation of the interface will be thoroughly described in D6.5 and the tests will be performed via a specific tool (e.g. SOAP UI, Postman).

5.2.4.3. Integration tests

Integration tests, as perceived by the Operational Tools, imply links with the following core modules:

- Information Hub (IH): Models and predictive algorithms are supposed to be fed with data that should come from the IH. And a similar approach applies for the results of executions, at least if they are supposed to be scheduled and represented in a time series mode. Basically, what needs to be tested is whether the Operational Tools are able to read from the IH (Read API) before executing models, and write to the IH the result (or part of it) from the executions.
- Dashboard: Part of the OT UI will be directly integrated within the Dashboard, so checking if the behaviour is the same as an independent or integrated component will be a first test. A second test relates to data harmonization through the IH. Output data from model executions will be stored in the IH, which should be later retrieved by the dashboard, at least for visualization purposes.
- Notification: event processing within the Operational Tools will be integrated (lined) with ElastAlert, which exposes a REST API. From the OT, it will be ensured that the rules are generated and properly propagated to ElastAlert.

Additionally, even if not treated as a module, models and predictive algorithms are supposed to be somehow available as services that will be invoked by the Operational Tools, therefore it seems also important to establish some checking of the services/predictive algorithms, before the integration in PIXEL (publication through the Operational Tools).

More detailed information will be provided via the Testlink tool.

5.2.5. Dashboard and notifications

The dashboard and notifications modules are being programmed using Node.js as backend and vue.js as front-end framework.

To evaluated them, unit, module and integration tests have been used. To create the unit and module test, the tools that have been selected are jasmine testing module and mock.js (Figure 17); the first one will be used to validate the backend and the second one will be used to evaluate the frontend. For the integration test a test suit (T6.5 Dashboard) has been defined in the Testlink project “fw:Framework”

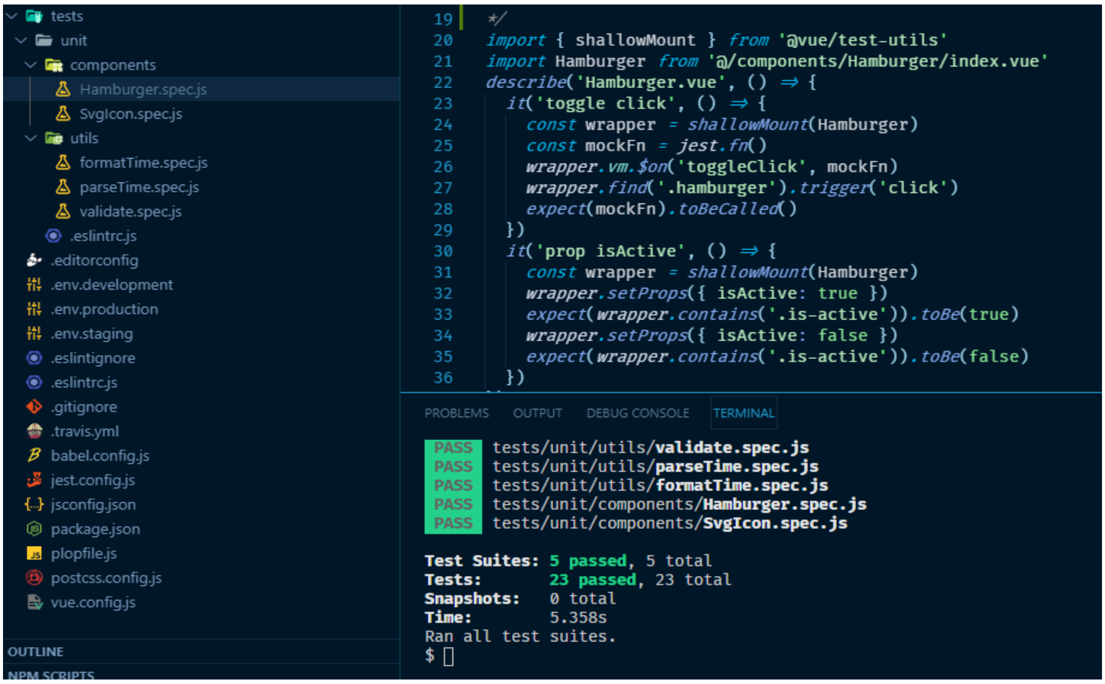


Figure 17: mock.js screenshot

5.2.6. Security

The security layer relies on FIWARE Generic Enablers that come with a full set of quality tests. For example on Github we have a full repository for functional tests but also non-functional tests.

For KeyRock, the main Identity Management server, FIWARE provides Sanity Check Procedures.

5.3. Pilots

The definition of the functional requirements of the pilots to be developed within the project, has been carried out in WP3, deliverables D3.2 and D3.4. A test suite has been created to validate the requirements of the pilots. Given the large number of test cases to be created, and the changes they will experiment over time, it was decided to use "Testlink" tool to carry out the management of the tests.

Table 28: TestLink trial projects

Trial Name	Testlink project
Bordeaux pilot	GPMB:T7.2 Port of Bordeaux-GPMB
Monfalcone pilot	PoM:T7.3 Port of Monfalcone
Thessaloniki pilot	ThPA:T7.4 Port of Thessaloniki
Piraeus Pilot	PPA:T7.4 Port of Piraeus

6. Conclusions and Future work

6.1. Conclusion

This document represents the first version of the Integration report of all the PIXEL IoT platform and all the trials. It contains a schedule and the roles assigned to each for the pilots and an evaluation plan with test cases to verify that the pilots fit their requirements. This information will be used to monitor the progress of developments and also as additional documentation to obtain better developments during the execution of the WP7.

At management level, this is the most complicated work package as it includes the integration of developments made in other work packages and the realization of 4 pilots. The delay of other WP may affect this one, for this reason a close and fluent communication is needed.

D7.1 proposes a planning as realistic as possible, but there are still issues to clarify that may affect the development of pilots. Currently, the format of some available data sources and when they will be available is not defined. For this reason, the data integration tasks of the pilots have been planned with flexibility, in order to manage this risk and in the event of a problem, some data can be temporarily simulated so that the pilot will not be delayed.

To avoid that this problem affects the definition and the schedule of other components of the platform, the information stored in the IH, should be stored following data models. In this way the platform for the data sources is independent. The unique module that is aware of the data sources, is DAL that transforms the original data in a predefined data model.

In order to facilitate the integration of the models and specially the addition of new models in the future (even beyond the life of the project), it has been decided to make use of microservices for the models. The models will be encapsulated in dockers and a common API is being defined, that will allow invoking them in the same way, regardless of the model of choice.

Thanks to the modular and scalable architecture defined in D6.2 the risk of having different data sources can be mitigated and the effort of adding new models minimized, which greatly facilitates the development of different pilots.

6.2. Future work

This document has been written based on 7 months of work (11 – 18) and it is the first version of the integration report. The second version (D7.2 month 27) will contain a refined version of the schedule and test plans and a more detailed list of test cases with preliminary test execution reports. Thanks to the fact that, at the time of writing the D7.2, the pilots will be in an advanced state and a working version of the architecture will be available, a strategy for monitoring and tracking the status of the platform can be defined. In addition, a realistic training plan can be created based on the experience acquired with the pilots.

The deliverable D7.3 “Pilots and Cross Pilot Collaboration Report” planned for M33 will summarize the work done in the four port pilots, including the deployment of PEI in all of them and also in inter-pilot integration and collaboration.

7. Bibliography

[1] Pipenv: Python Development Workflow for Humans. https://pypi.org/project/pipenv/

[2] Atmo Nouvelle-Aquitaine Open Data https://data-atmo-na.opendata.arcgis.com/

[3] FIWARE data model for air quality https://fiware-datamodels.readthedocs.io/en/latest/Environment/AirQualityObserved/doc/spec/index.html

[4] FIWARE data model for weather observed https://fiware-datamodels.readthedocs.io/en/latest/Weather/WeatherObserved/doc/spec/index.html

[5] Erik van Veenendaal . Standards – What about it?. Testing experience. December 2009 ISSN 1866-5705

https://www.istqb.org/images/Articles/veenendaal_Standards%20to%20be%20used%20with%20common%20sense!.pdf

[6] What is the Best Software Testing Certification? https://astqb.org/what-is-the-best-software-testing-certification/

[7] Test Plan. http://softwaretestingfundamentals.com/test-plan/

8. Annex

Example of a report with a list of some test cases created for the Energy management trial, were created with the TestLink tool.

Access to the TestLink tool can be obtained by using the credentials used in section 5 of this document to have the latest version of the test cases described.

TestLink

Test Plan Design Report

Test Project: T7.2 Port of Bordeaux-GPMB
Test Plan: Initial Test plan

Printed by TestLink

Table of Contents

Test Project: T7.2 Port of Bordeaux-GPMB

This project will be in charge of evaluating the work done in the Task7.2: energy management trial

1.1.Test Suite : Scenarios

1.1.1.Test Suite : PA-1 Port Agent scenario

Test Case GPMB-16: External Data Sources [Version : 1]	
Author:	cgarnier
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-17: GPMB IT system [Version : 1]	
Author:	cgarnier
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-18: Dashboard / notifications [Version : 1]	
Author:	cgarnier
Execution type:	Manual

Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

1.1.2.Test Suite : EM-1 Energy manager scenario

1.1.2.1.Test Suite : Support electricity consumption sensors [9]

PIXEL must be able to read/receive data from a set of interconnected electricity consumption sensors or by means of an existing database collecting such data from different sources.

Energy consumption data is available through an API call

To be able to use electricity consumption data in prediction algorithms, calculation of PEI, dashboard status updates, etc.

Test Case GPMB-2: DAL: Verify that electricity consumption is loaded [Version : 1]	
Author:	pixel
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-3: OT: Verify the prediction algorithms with electricity consumption [Version : 1]	
Author:	pixel
Summary: the operational tool should be able to execute prediction algorithms using electricity data	
Preconditions: the data are properly loaded. the prediction algorithms are available.	
Execution type:	Manual

Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-4: DH:Verify the graphs related with data [Version : 1]		
Author:	pixel	
Summary: Verify that the dashboard is able to show graphs related with energy consumption. Verify that the graph info is consistent with the data evaluated - verify alerts /notification? - verify kpi? -		
Preconditions: the data are loaded, and the algorithms executed / model created		
#:	Step actions:	Expected Results:
1	Open the dashboard administrator	the dashboard administrator is opened
2	configure the different visualizations "graphs" for the data	the visualizations are created
3	access to the home dashboards	the visualizations are shown
4	verify that the visualizations show correct information (for all visualizations related with this algorithms)	check (with the model and data owners) that the visualization data is correct
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Jira tickets:		
Requirement:		
Keywords:	None	

Test Case GPMB-5: Verify notifications and alerts [Version : 1]	
Author:	pixel
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-6: verify the PEI [Version : 1]	
Author:	pixel
Summary: verify that the PEI takes these algorithms in consideration	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

1.1.3.Test Suite : StM-1 Statistics manager scenario

which aims to provide a data analysis of vessels calls by connecting VIGIEsip to Pixel Platform to obtain historical data and then use small analytics tools. This means the integration of the data acquisition layer of PIXEL developed in WP6 with the PMS of GPMB to acquire and then analyse vessells' calls

Test Case GPMB-10: Test if the data coming for the statistical model are stored in the Information Hub [Version : 1]	
Author:	cgarnier
Preconditions: Successful tests for GPMB-7, GPMB-8, GPMB-9	
Execution type:	Manual
Estimated exec. duration (min):	

Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-11: Test that the Dashboard accessed to the data of the statistical model [Version : 1]	
Author:	cgarnier
Preconditions: Successful tests for GPM7, GPMP-8, GPMB-9 and GPMB-10	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

1.2.Test Suite : Software Integration

Test Case GPMB-12: Test if integration of VIGIEsip in order to collect information about ship call works [Version : 1]	
Author:	cgarnier
Summary: GPMB, trough VIGIEsip push data every hour and a NGSI agent acquire and transform them into the data model defined for vessel calls (This data model has been proposed by the port activity scenario model in WP4).	
Preconditions: Data model for vessels calls have been detailed in WP4	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	

Requirement:	
Keywords:	None

Test Case GPMB-13: Test if users can define supply chains and machine specification. **[Version : 1]**	
Author:	cgarnier
Summary: This is not a direct software integration but there is a need of interface in order to allow a port agent to provide the different specifications of supply chains and engines. We need to test if the interface works well and if the data are well stored in the Information Hub	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-14: Test if integration of open-data coming from ATMO Nouvelle-Aquitaine for air quality works **[Version : 1]**	
Author:	cgarnier
Summary: ATMO Nouvelle Aquitaine provide web-services https://data-atmo-na.opendata.arcgis.com. PIXEL data acquisition layer could use existing FIWARE data model for air quality https://fiware-datamodels.readthedocs.io/en/latest/Environment/AirQualityObserved/doc/spec/index.html We need to test if the data are obtained through the Data Acquisition Layer and stored in the Information Hub	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None

Test Case GPMB-15: Test if integration of a weather data works [Version : 1]	
Author:	cgarnier
Summary: A weather station is installed in GPMB and send data to the DAL. We needed to test it the data are well obtained by the Data Acquisition Layer and stored in the Information Hub.	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Jira tickets:	
Requirement:	
Keywords:	None