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

D8.3 – Technical Evaluation v2

Deliverable No.	D8.3	**Due Date**	30-SEP-2021
Type	Report	**Dissemination Level**	Public
Version	2.0	**Status**	Final
Description	This document will present the results of technical performance and user acceptance evaluations.		
Work Package	WP8		

Authors

Name	Partner	e-mail
Ignacio Lacalle	P01 – UPV	iglaub@upv.es
Benjamín Molina	P01 – UPV	benmomo@upvnet.upv.es
Rafa Vaño Garcia	P01 - UPV	ravagar2@upv.es
Miguel Ángel Llorente Carmona	P02 - PRODEVELOP	mllorente@prodevelop.es
Álvaro Garcia	P03 - XLAB	alvaro.garcia.faura@xlab.si
Gilda de Marco	P04 - INSIEL	gilda.demarco@insiel.it
Charles Garnier	P05 CATIE	c.garnier@catie.fr
Romain Quéraud	P05 CATIE	r.queraud@catie.fr
Marc Despland	P06 - ORANGE	marc.despland@orange.com
Olivier le Brun	P07 - CREOCEAN	lebrun@creocean.fr
Teodora Milosevic	P08 - MEDRI	teodoramilosevic1@gmail.com
Tamara Corsano	P09 - SDAG	t.corsano@sdag.it
Eirini Tserga	P10 - THPA	etserga@thpa.gr
Athanasios CHALDEAKIS	P11 - PPA	ahaldek@gmail.com
Stefano Bevilacqua	P12 - ASPM / P16 - APT	stefano.bevilacqua@porto.trieste.it
Fabrice Klein	P13 - GPMB	f-klein@bordeaux-port.fr
Leonidas Pitsikas	P14 - PEOPLE	lpitsikas@peoplethinkbeyond.com
Annie Kortsari	P15 - CERTH	akorts@certh.gr

History

Date	Version	Change
2019-10-07	0.1	Document creation with first version of ToC
2020-11-30	0.2	Draft of section 3
2021-03-31	0.3	Added answers to the old platform
2021-06-25	0.4	Draft of section 4 without answers from all ports
2021-09-01	0.5	Draft of section 4 analysis with all answers
2021-09-16	1.0	Ready for internal review
2021-09-27	2.0	Ready for submission

Key Data

Keywords	Data collection, data analysis, technical assessment, impact assessment, platform, product quality, quality in use, data quality
Lead Editor	Romain Quéraud, P05 CATIE
Internal Reviewer(s)	GPMB, APT

Abstract

The goal of this deliverable "Technical Evaluation v2.0" is to assess the technical work done in PIXEL up to the end of the project. Most of the work done in the document is the application of what we defined in the previous deliverable D8.1 "Evaluation Plan" and follows the same structure as its predecessor D8.2 "Technical Evaluation v1.0". The technical impact assessment is, as such, split into two distinct evaluations:

- The technical impact assessment of the PIXEL platform
- The technical impact assessment of the PIXEL use-cases

For the PIXEL Platform, we evaluated technical characteristics per module at a laboratory level, such as memory consumption and CPU usage and obtained different KPIs. Those characteristics are derived from the ISO/IEC norm "Product Quality Model". Evaluated modules are those defined in the table 5 of deliverable D8.1 that were not finished for D8.2. They all showed very good results.

The technical impact assessment of the PIXEL Use-cases evaluates the project following the ISO/IEC norm "Quality In Use Model" for the user acceptance and "Data Quality Model". We waited that the end-users had enough knowledge of the platform to disseminate the questionnaires. Having a large enough pool of answerers from ports allows us to calculate the characteristics planned by the ISO/IEC models.

Most of the characteristics show very good results and prove that the platform has been well received by end users, both in terms of quality in use, and with the quality of data. Due to the extension of WP7 and the postponement of the final delivery of the platform, the only characteristics that could be improved are those referring to the requirements and the user stories completion. However, comparing the technical impact assessment of the PIXEL platform and the technical impact assessment of the PIXEL use cases allowed us to assert that those requirements/user stories are indeed achievable with the platform thanks to the variety of modules/models.

The overall evaluation showed that the platform is working as intended, and that the only thing left to do is using it on a regular basis to reveal its full potential.

Statement of originality

Table of contents

List of Tables

List of Figures

List of acronyms

Acronym	Explanation
AIMQ	Methodology for Information Quality Assessment
CPU	Central Processing Unit
DAL	Data Acquisition Layer
EMP	Environmental Management Plan
GPMB	Grand Port Maritime de Bordeaux - Port of Bordeaux
IH	Information Hub
IT	Information Technology
ISO/IEC	International Organisation for Standardization / International Electrotechnical Commission
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
OS	Operating System
OT	Operational Tools
PA	Port Authority
PAS	Port Activity Scenario; It is the model provided by T4.1.
PMIS	Port Management Information System
PPA	Piraeus Port Authority SA
TAM-3	Technology Acceptance Model 3
RAM	Random-Access Memory
REST API	Representational State Transfer Application Programming Interface
SDAG	Stazioni Doganali Autoportuali Gorizia
SILI	Sistema Informativo Logistico Integrato (Integrated Logistic Information System), a system provided by Regione Friuli Venezia Giulia and managed by Insiel to monitor and authorize entries to the Ports of Monfalcone and Trieste; it also monitors dangerous goods flows along the regional motorway network
THPA	Thessaloniki Port Authority

1. Introduction

1.1. Objectives & scope of the document

This document is the third document of WP8 and deals with the technical impact assessment of the PIXEL project. Its goal is to present the collected data and evaluate them according to the methodology defined in the previous documents, D8.1 and D8.2. This document is structured in two main parts:

- The first one addresses the technical evaluation of the PIXEL platform. This evaluation has been performed by the technical partner of the project and focuses on technical achievement and performance of PIXEL.
- The second one addresses the technical evaluation of the PIXEL Use Cases. This evaluation has been performed by the ports and focuses on the feedback of end-users. This explains why for the same indicator different results are obtained.

For both parts, we took inputs from what had been defined in the previous deliverable (D8.1) in which we defined the characteristics and sub-characteristics to evaluate. We also defined involved partners for the evaluation, and, as a result, we present the collaborative work in this document.

This deliverable provides deep inputs and analysis of the PIXEL platform both from a technical side and the end-users point of view. This deliverable is valuable for the PIXEL consortium as the evaluation of the work performed during the whole project but also to the external stakeholders and EU commission since it sums up what has been achieved with the associated performance.

1.2. Deliverable context and structure

Table 1. Deliverable context and structure

Keywords	Subjects
Objectives	The overall goal of WP8 is to evaluate the project in terms of (i) technical functioning and interoperability of all PIXEL Components, (ii) usability and (iii) results. The scope of D8.3 is to apply the methodology defined in D8.1 in order to gather data and derive different characteristics.
Exploitable results	Although not directly generating any exploitable results, D8.3 evaluates the technical side of all KERs, providing KPI and reports in order to assess the technical impact of the PIXEL project.
Work plan	The D8.3 is directly related to: ● WP3 for the users requirements; ● WP4 and WP5 for the technical evaluation of the models; ● WP6 which gather all elements to a laboratory working platform; ● WP7 which integrates the platform in the ports.
Milestones	This deliverable contributes to MS10 – Final Evaluation (Means of

	verification: D8.3, D8.4 and D8.5 released and approved).
Deliverables	Detected inputs from: ● D3.2: PIXEL Requirements Analysis ● D3.4: Use cases and scenarios manual v2 ● D4.2: PIXEL Models v2 ● D4.4: Predictive Algorithms v2 ● D5.3: PEI definition and Algorithms v2 ● D6.4: PIXEL data acquisition, information hub and data representation v2 ● D7.2: Integration Report v2 ● D8.1: Evaluation Plan ● D8.2: Technical Impact Assessment v1
Risks	This deliverable deals with a risk identified in D8.1, relative to the delay of the platform trials beginning. As such, evaluation of the PIXEL use-cases have been delayed to this deliverable, instead of D8.2. However, for this deliverable, PIXEL end-users have had enough time to use the platform and gain enough experience using it. Another identified risk is that the evaluation could have shown bad results. However everything has been controlled and monitored by the previous deliverables, so this risk has been minimised.

1.3. Intended audience

This deliverable aims at providing feedback and guidelines to PIXEL developers. As such, we directly target the responsible partners from WP4-5-6-7 that would know how their technical work is perceived by the end-users, and how they could improve it for future developments. As we also consider the final users' feedback, it can also be read by those final users which would like to check how their evaluation has been considered. Finally, this deliverable is targeted to any reader interested in knowing how the platform is performing.

This deliverable provides an overall evaluation of the PIXEL platform, what has been technically achieved, what are the lessons learned and how the PIXEL platform is perceived by the end-users. The deliverable can also be used as a basis to build on all the work performed during the project.

2. Applying Evaluation and Validation Framework

The Technical Impact Assessment has been conducted for both the PIXEL Platform (for the evaluation of the IT part of the PIXEL project) and the PIXEL use cases (for the evaluation of the user acceptance and data quality). It focused on:

- Technical performance;
- User acceptance;
- Information security and robustness.

To develop the technical impact assessment framework, we based on three evaluation models. These models are based on the International Standards on System and Software Quality Requirements and Evaluation (Square):

- The first model (ISO/IEC 25010 Product Quality Method) is related to the evaluation of the PIXEL platform in regard to the properties of the software and the dynamic properties of the system.
- The second model (ISO/IEC Quality in Use Model) is directly linked with the assessment of the usage evaluation of the platform by end-users (ports for PIXEL).
- The last model (ISO/IEC 25012 Data Quality Model) is somewhat complementary with the two others since it refers to the evaluation of the data provided by the PIXEL platform.

For the technical impact assessment of PIXEL, these models have been used, adapted or modified to our specific context. The ISO standard defines a list of characteristics and sub-characteristics for each of the three models. In order to clearly identify which ones of these characteristics are applicable to PIXEL, a survey has been shared with the whole consortium. Results of this survey have been described and analysed in D8.1. We used them as a basis for the technical impact assessment. For each characteristic or sub-characteristic listed in the ISO standards, the PIXEL consortium has agreed on which ones must be assessed and has established how to measure them. The evaluation criteria were also defined in the previous deliverable, D8.1. We have already used those criteria to do the evaluation in D8.2, and we are now reconducting it here in D8.3. WP8 is heavily dependent on other work packages that focus on technical development. It has been noticed as a risk, in D8.1, that integration (WP7 mainly) would encounter some delay. This risk is also related to the fact that, due to this delay, many of the PIXEL partners waited as much time as possible to answer the questionnaires in order to have the maximum experience using the platform. Thus, we have had to be efficient at analysing the questionnaires. For the technical impact assessment of the PIXEL Use Cases, we defined in D8.1 the characteristics/sub-characteristics to be evaluated and the calculation method in order to obtain the different KPIs. We also defined in D8.1 and D8.2 the different questionnaires that have been disseminated. While this deliverable, D8.3, extends the work done in D8.2 for the technical impact assessment of the PIXEL platform, it also presents the collected and analysed results for the technical impact assessment of the PIXEL use cases. As this is the technical impact assessment deliverable of the project, we took input from all technical work packages, which means, for the technical impact assessment of the PIXEL platform:

- WP4: T4.1, T4.2, T4.3, T4.4, T4.5
- WP5: Development of PEI algorithm and model
- WP6: T6.2, T6.3, T6.4, T6.5, T6.6

For the technical impact assessment of the PIXEL Use-Cases, inputs are mainly coming from WP7 integration: WP7: T7.2, T7.3, T7.4, T7.5

3. Technical Impact Assessment of the PIXEL platform

For each module that is evaluated, we list the characteristics/sub-characteristics and the calculated results for the according KPI.

KPIs are estimated either by expert judgement or by the development of tools for automated measurements.

Expert judgement has been used for those KPIs that are either too complicated to automate and an expert approach is more efficient, or where a more qualitative evaluation approach is needed. Expert judgement is performed using desk research, where an expert evaluates the KPI using the approach defined in D8.1/D6.3. The defined **experts are the technical partners of the project** who developed the different modules. As an exemple, functional suitability and Maintainability will be estimated using this approach.

Automated measurements are performed either by usage of existing evaluation software or by development of custom tools for this purpose. Part of the KPIs have been collected using JMeter measurements. The Apache JMeter™ application is an open-source software designed to load test functional behaviour and measure performance. Performance efficiency and Reliability have been measured using this approach.

3.1. Vessel ETD prediction from FAL forms

This predictive algorithm was presented in Section 2.2.2 of Deliverable 4.4 (D4.4). It is implemented as a Docker image whose core component is a Flask API to which requests with vessel calls data are made for it to return a predicted ETD.

3.1.1. Assessment scenario

A custom script was implemented to measure all the KPIs that were not evaluated through expert judgement. The strategy we followed consisted in running the API and, in a separate process, measuring the resource utilization of the API process when carrying several requests. Specifically, results are provided when making 5000 requests sequentially (so for 5000 different vessel calls) all of them equivalent to those used in the real environment. Tests were also carried out using parallel requests. The sub-module handling the communication with the Information Hub was not included in the evaluation, since the results may vary depending on network conditions. Both the API and the performance evaluation script were running in the same machine, a Lenovo ThinkPad T14 Gen 1 with the following technical specifications:

- OS: Ubuntu 20.04.2 LTS 64 bits
- CPU: AMD Ryzen 7 Pro 4750U Processor (1.70 GHz, 8 Cores, 16 Threads, 8 MB Cache)
- RAM: 32 GB = 2x 16GB DDR4 (3200 MT/s)
- Disk: 512 GB SSD
- CPU and memory measurements were carried out using the Python library psutil[1].

[1] https://psutil.readthedocs.io/en/latest/

3.1.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes.

This KPI verifies that (1) a process to build a model for a new port (training phase) is straightforward and documented; (2) A process to make predictions does not include unnecessary steps, i.e. is executed by one API call.

1 Training phase: The process is as straightforward as possible, but as there is not a harmonised dataset for FAL forms across different ports, there are still some implementation steps needed for each new port. After this initial transformation of input data, the model can be built and deployed following a few simple, documented, steps[2].

2 Usage phase: the prediction model has been implemented according to specifications for the integration in the Operational Tools and the Information Hub, thus supporting the deployment and usage of the model with the standard PIXEL API specification.

2. The portion of completed requirements: 100% (relevant)

Common functional requirements (Must Have, Should Have) from deliverable D3.2 will be taken as input in order to extract all requirements targeting this PA. We evaluated this KPI on all relevant Common functional requirements (Product: PIXEL platform or PA) directly targeting this PA. For all requirements, we stated if they were relevant or not for our specific PA, then we stated if we completed it (if relevant). The Table below summarizes this assessment.

Table 2. PIXEL PA ETD Predictions Common Functional Requirements status

Common functional Requirements	Priority	Relevant	Completed	Comments
Import historical Data (36)	M	1	1	Model is trained on historical data.
Interaction with models (41)	M	1	1	The model provides an OT interface and is integrated in the platform.
Anomaly and event list (44)	M	0		The requirement refers to CEP implementation, not relevant.
Homogenize Data (61)	M	1	1	Agreed data model for port calls is used in the PA.
Catalogue of models (62)	M	1	1	The model provides an OT interface and is integrated in the platform.
Detection of anomalies (63)	M	0		Not relevant for this PA.
Feedback (64)	S	0		This hasn't to be dealt at the model level.

[2] https://gitlab.xlab.si/dejan_stepec1/vessel_calls/-/blob/master/05_ensemble/Initial%20results%20-%20CatBoost%20proper%20-%20clanek.ipynb

Centralized user administration system (65)	M	0		This hasn't to be dealt at the model level.
UI notification System (67)	M	0		This hasn't to be dealt at the model level.
Operational Interface (71)	M	1	1	The model provides an OT interface and is integrated in the platform.
Discovery service for data (104)	M	0		This hasn't to be dealt at the model level.

A total of 5 requirements are related to functionality provided by the ETD PA. Out of those all have been implemented and tested.

- Total requirements: 5.
- Fulfilled requirements: 5.
- Portion of completed requirements: 100%.

3. Maximum number of connected data sources: 1

Only one data source is connected at a time to every deployed instance of the model.

4. Maximum database size: No maximum.

Records (i.e. vessel calls) are processed sequentially, so there is no limit regarding the size of the database to be processed by the algorithm.

5. Average latency: 25.2992 ms.

The elapsed field in the API response is taken as the value for latency. The maximum and minimum values measured in our tests were 109.7720 ms and 15.6940 ms, respectively.

6. Throughput: 16.36 KB/s (average).

The throughput is computed as the size of the content in an API response divided by its elapsed time. The maximum and minimum values measured in our tests were 25.69 KB/s and 3.58 KB/s respectively.

7. Mean CPU Utilisation: 105.1557 %

The CPU utilization goes over 100 % because the process can run in several threads. The plot below depicts the CPU utilization for all the 5000 API requests made sequentially.

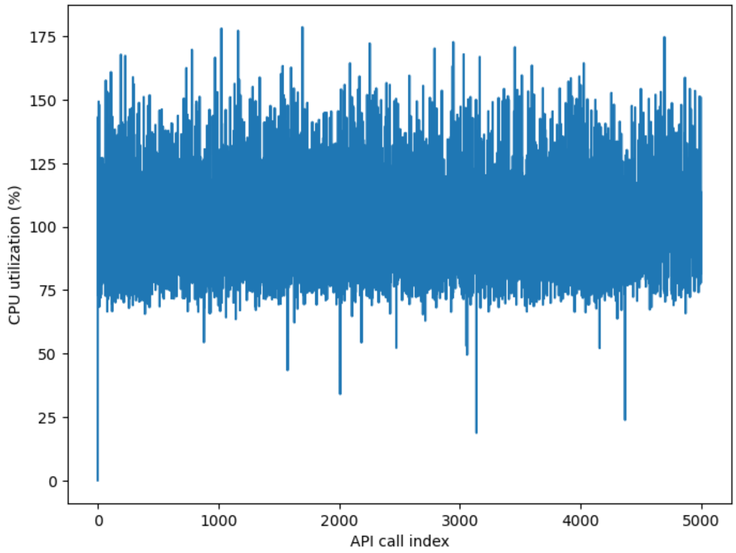

Figure 1. CPU utilization for all the different 5000 ETD prediction API requests performed sequentially.

8. Mean memory usage: 100.47 MB

The memory used by the API process increases with the number of API calls up to a certain point, after which it is kept practically constant. The maximum value measured is 100.47 MB. We speculate that the garbage collection mechanism enters in force and prevents the memory from growing larger. Subsequent API calls do not increase the memory usage.

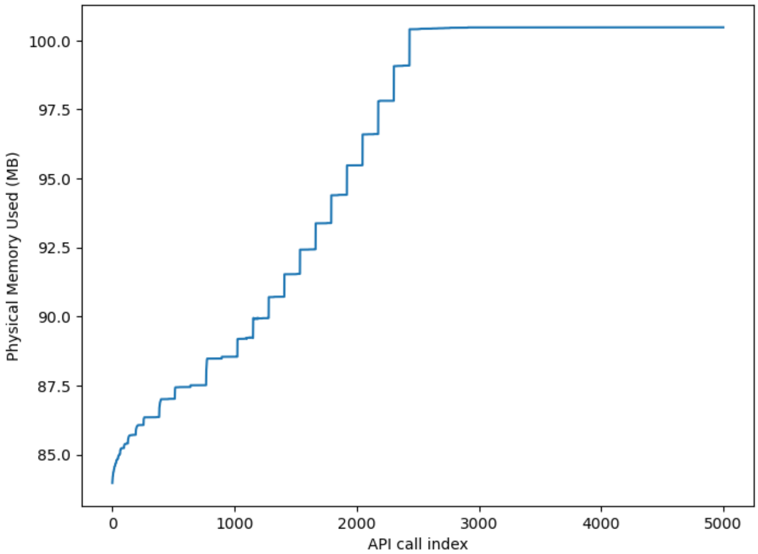

Figure 2. Memory utilization for all the different 5000 ETD prediction API requests performed sequentially.

9. Maximum memory usage: 100.47 MB.

10. Maximum processing power used: 178.6 % of CPU.This is using more than one thread.

11. Simultaneous requests: Linear.

We ran a test submitting from 5 up to 100 simultaneous API requests and measured the mean elapsed time per request. In the plot below, we can observe that the curve generally follows a linear increasing trend. No performance degradation is observed.

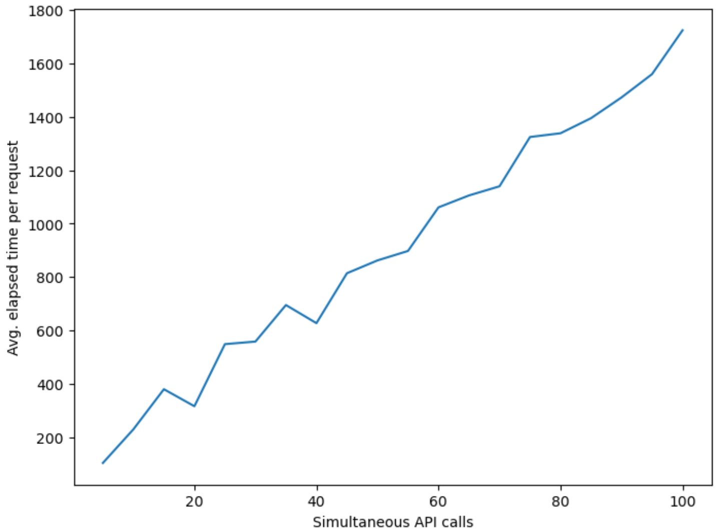

Figure 3. Elapsed time when performing a varying number of simultaneous ETD prediction API requests.

12. % of modularity: 33%The PA consists of several core modules, APIs to integrate it in the PIXEL architecture and prediction models for ports. In principle, either core modules (the learning module and prediction algorithms) or the APIs (REST, OTs, IH) could be replaced independently. However, the models and core modules are highly interdependent and they cannot be replaced without affecting the other. If we assume that we have three main modules parts as explained above (core ML, APIs, Models) the % of modularity is 1/3 = 33%.

13. % of reusable assets: 66%

Of the three modules identified above (core ML, APIs, Models), models cannot be re-used among different use cases because they must be built on historical data for the concerned port. All other parts of the PA can be reused with no restrictions: 2/3=66%.

Results for all KPIs are summarized in the table below :

Table 3. KPI summary for the Prediction of PA ETD Predictions model

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	100% (relevant)
Performance efficiency		

Capacity	3. Maximum number of connected data sources	1
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	25.2992 ms
	6. Throughput	16.36 KB/s (average)
Resource utilization	7. Mean CPU Utilisation	105.1557%
	8. Mean memory usage	100.47 MB
	9. Maximum memory usage	100.47 MB
	10. Maximum processing power used	178.6 % of CPU
Reliability		
Maturity	11. Simultaneous requests	Linear
Maintainability		
Modularity	12. Percentage of modularity	33%
Reusability	13. Percentage of reusable assets	66%

3.2. Vessel short-term ETA prediction from AIS data

Deliverable D8.1 contains a clerical error in Table 4, row 6. The text should state "Vessel short-term ETA prediction from AIS data" instead of "Vessel short-term ETD prediction from AIS data". This predictive algorithm was presented in Section 3.1.4 of Deliverable 4.4 (D4.4).

3.2.1. Assessment scenario

The algorithm has been evaluated for the same use case that was presented in D4.4: the Bassens terminal in Bordeaux. The prediction model has been trained with historical data. Re-training is not expected unless a significant amount of new historical data is available. The normal execution of the algorithm would thus only imply the inference of the ETA given a current AIS data point. Therefore, the performance evaluation we have carried out in this document is restricted to the inference phase. The best performing model is a simple linear regression. Given the low complexity of the model, it is unfeasible to evaluate its performance on a low number of data points. The results provided here are obtained when running the model for 100 million data points. The performance evaluation script was executed in a Lenovo ThinkPad T14 Gen 1 with the following technical specifications:

- OS: Ubuntu 20.04.2 LTS 64 bits
- CPU: AMD Ryzen 7 Pro 4750U Processor (1.70 GHz, 8 Cores, 16 Threads, 8 MB Cache)
- RAM: 32 GB = 2x 16 GB DDR4 (3200 MT/s)
- Disk: 512 GB SSD

CPU and memory measurements were carried out using the Python library psutil[3].

3.2.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes.

This KPI verifies that (1) a process to build a model for a new port (training phase) is straightforward and documented[4]; (2) A process to make predictions does not include unnecessary steps, i.e. is executed by one API call.

1. Training phase: The training phase requires several input parameters and afterwards, the process is automated. The process consists of: (1) selecting AIS input data for training, (2) running port calls calculations, (3) transforming output to a structure suitable for training, (4) training the prediction model and (5) deployment.
2. Usage phase: the prediction model has been implemented according to specifications for the integration with the Information Hub, thus supporting the deployment and usage of the model with the standard PIXEL API specification.

[3] https://psutil.readthedocs.io/en/latest/

[4] https://gitlab.xlab.si/pixel/ais-trajectory-processing#estimated-time-of-arrival-eta-prediction-from-ais-data

2. The portion of completed requirements: 60% (relevant) (however, these “missing” requirements have been dealt with via the OTs or Dashboard correspondingly) – see below.

Common functional requirements (Must Have, Should Have) from deliverable D3.2 have been taken as input in order to extract all requirements targeting this PA. We evaluated this KPI on all relevant Common functional requirements (Product: PIXEL platform or PA) and requirements from Use cases directly targeting this PA. For all requirements, we stated if they were relevant or not for our specific PA, then we stated if we completed it (if relevant). Tables below summarizes this assessment.

Table 4. PIXEL PA ETA Predictions Common Functional Requirements status

Common functional Requirements	Priority	Relevant	Completed	Comments
Import historical Data (36)	M	1	1	Model is trained on historical data.
Interaction with models (41)	M	1	1	The model is integrated in the PIXEL Information Hub.
Anomaly and event list (44)	M	0		The requirement refers to CEP implementation, not relevant.
Homogenize Data (61)	M	1	1	Usage of the standard AIS data structure.
Catalogue of models (62)	M	1/0	0	The model does not provide an OT interface. However, this requirement is dealt with via the OTs.
Detection of anomalies (63)	M	0		Not relevant for this PA.
Feedback (64)	S	0		This hasn't to be dealt at the model level.
Centralized user administration system (65)	M	0		This hasn't to be dealt at the model level.
UI notification System (67)	M	0		This hasn't to be dealt at the model level.
Operational Interface (71)	M	1	0/1	The model does not provide an OT interface. However, this model has an associated visualization via Views in the Dashboard.
Discovery service for data (104)	M	0		This hasn't to be dealt at the model level.

A total of 5 requirements are related to functionality provided by the ETA PA. Out of those three have been implemented and tested. As there was no need to integrate the PA with Operational Tools, two requirements were not fulfilled.

- Total requirements: 5.
- Fulfilled requirements: 3.
- Portion of completed requirements: 60%. (however, these “missing” requirements have been dealt with via the OTs or Dashboard correspondingly).

3. Maximum number of connected data sources: 1

Only one data source is connected at a time to every deployed instance of the model.

4. Maximum database size: No maximum.

Records (i.e. AIS messages) are processed sequentially, so there is no limit regarding the size of the database to be processed by the algorithm.

5. Average latency: 457.9722 ms for all the 100 million records.

6. Throughput: 218353858.95 records/s.

7. Mean CPU Utilisation: 87.12 %.

8. Mean memory usage: 397.07 MB.

9. Maximum memory usage: 775.57 MB.

10. Maximum processing power used: 100 %.

11. Simultaneous requests: 1

This prediction algorithm only handles one request at a time. Simultaneous requests are not contemplated.

12. % of modularity: 33% The PA consists of several core modules, APIs to integrate it in the PIXEL architecture and prediction models for ports. In principle, either core modules (the learning module and prediction algorithms) or the APIs (REST, OTs, IH) could be replaced independently. However, the models and core modules are highly interdependent and they cannot be replaced without affecting the other. If we assume that we have three main modules parts as explained above (core ML, APIs, Models) the % of modularity is 1/3 = 33%.

13. % of reusable assets: 66%

Of the three modules identified above (core ML, APIs, Models), models cannot be re-used among different use cases because they must be built on historical data for the concerned port. All other parts of the PA can be reused with no restrictions: 2/3=66%.

Results for all KPIs are summarized in the table below :

Table 5. KPI summary for the Prediction of PIXEL PA ETA Predictions model

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes

Functional completeness	2. Portion of completed requirements	60% (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	1
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	457.9722 ms for all 100 million records
	6. Throughput	218353858.95 records/s.
Resource utilization	7. Mean CPU Utilisation	87.12%
	8. Mean memory usage	397.07 MB
	9. Maximum memory usage	775.57 MB
	10. Maximum processing power used	100 %
Reliability		
Maturity	11. Simultaneous requests	1
Maintainability		
Modularity	12. Percentage of modularity	33 %
Reusability	13. Percentage of reusable assets	66 %

3.3. Vessel detection from remote sensing

This predictive algorithm was presented in Section 4.2.2 of Deliverable 4.4 (D4.4).

3.3.1. Assessment scenario

For this prediction algorithm, only the inference phase when using a pre-trained model was evaluated, since there is no need to re-train the model once deployed. The technical evaluation was carried out running the algorithm for 100 images of size 1333 x 800 pixels. The performance evaluation script was executed in a machine with the following technical specifications:

- OS: Ubuntu 20.04.2 LTS 64 bits
- CPU: AMD Ryzen Threadripper 3960X 24-Core Processor (48 threads)
- GPU: 4x NVIDIA GeForce RTX 2080 Ti 11GB
- RAM: 128 GB DDR4
- Disk: Samsung 970 EVO+ NVMe M.2. (2x 1TB + 1x 2TB

For completeness, we have included an additional KPI, “maximum GPU memory usage”, in this scenario. CPU and memory measurements were carried out using the Python library psutil[5].

3.3.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes.

This KPI verifies that (1) a process to build a model for a new port (training phase) is straightforward and documented; (2) A process to make predictions does not include unnecessary steps, i.e. is executed by one API call.

1 Training phase: The process is as straightforward as possible, but there is a specific step to be customised to get satellite images from different providers (ESA, Planet, Airbus, etc.). After this initial gathering and transformation of input data, the model can be built and deployed following a few simple, documented, steps[6].

2 Usage phase: As the model is general-purpose and not tightly coupled with any of the use-cases (i.e. it is global), it has not been integrated into the standard PIXEL workflow.

2. The portion of completed requirements: 40% (relevant) (however, these “missing” requirements have been dealt with via the OTs or Dashboard correspondingly) – see below

Common functional requirements (Must Have, Should Have) from deliverable D3.2 will be taken as input in order to extract all requirements targeting this PA. We evaluated this KPI on all relevant Common functional requirements (Product: PIXEL platform or PA) and requirements from Use cases directly targeting this PA. For

[5] https://psutil.readthedocs.io/en/latest/

[6] https://gitlab.xlab.si/pixel_eo/data_builder

all requirements, we stated if they were relevant or not for our specific PA, then we stated if we completed it (if relevant). The Table below summarizes this assessment.

Table 6. PIXEL Vessel Detection Predictions Common Functional Requirements status

Common functional Requirements	Priority	Relevant	Completed	Comments
Import historical Data (36)	M	1	1	Model is trained on historical data.
Interaction with models (41)	M	1/0	0	The model has not been integrated with OTs, thus it does not support interaction with other models. However, this is not considered a shortcoming as this model has not been required from any pilot to meet their objectives. This model would be easily integrated if needed.
Anomaly and event list (44)	M	0		The requirement refers to CEP implementation, not relevant.
Homogenize Data (61)	M	1	1	Usage of the standard AIS data structure and standard satellite images format.
Catalogue of models (62)	M	1/0	0	The model has not been integrated with OTs and the Dashboard. However, this requirement is dealt with via the OTs.
Detection of anomalies (63)	M	0		Not relevant for this PA.
Feedback (64)	S	0		This hasn't to be dealt at the model level.
Centralized user administration system (65)	M	0		This hasn't to be dealt at the model level.
UI notification System (67)	M	0		This hasn't to be dealt at the model level.
Operational Interface (71)	M	1	0/1	The model has not been integrated with OTs. However, the results of this model could be associated to a visualization in “Views” in the Dashboard if those would be integrated in the Information Hub.
Discovery service for data (104)	M	0		This hasn't to be dealt at the model level.

A total of 5 requirements are related to functionality provided by Vessel detection from remote sensing. Out of those 2 have been implemented and tested.

- Total requirements: 5.
- Fulfilled requirements: 2.

- Portion of completed requirements: 40%. (however, these "missing" requirements have been dealt with via the OTs or Dashboard correspondingly)

3. Maximum number of connected data sources: 1

Only one data source is connected at a time to every deployed instance of the model.

4. Maximum database size: No maximum.

Satellite images are processed sequentially, one at a time, so there is no limit regarding the size of the database to be processed by the algorithm.

5. Average latency: 20.4432 s for all the 100 images

6. Throughput: 4.89 images/s.

7. Mean CPU Utilisation: 95.39 %.

8. Mean memory usage: 1.3650 GB.

9. Maximum memory usage: 3.7966 GB.

10. Maximum processing power used: 106.6 % (several threads can be used).

11. Simultaneous requests: 1

This prediction algorithm only handles one request at a time. Simultaneous requests are not contemplated.

12. % of modularity: 60%

The PA consists of several modules: (1) AIS data processing, (2) satellite images processing, (3) data fusion algorithm, (4) detection model learning module and (5) vessel detection module. In principle, learning-detection modules are highly dependent among each other, while others can be easily replaced by keeping the same API. If we assume that we have 5 main parts as explained above the % of modularity is 3/5 = 60%.

13. % of reusable assets: 100%

All the modules defined above can be reused without restrictions for different detection scenarios with no need ro rebuild the model or any other part.

14. Maximum GPU memory usage (additional KPI): 4.2979 GB.

Results for all KPIs are summarized in the table below :

Table 7. KPI summary for the Prediction of Vessel Detection Predictions model

Sub-characteristics	KPI	Result
Functional suitability		

Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	40 % (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	1
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	20.4432 s for all the 100 images
	6. Throughput	4.89 images/s
Resource utilization	7. Mean CPU Utilisation	95.39%
	8. Mean memory usage	1.3650 GB
	9. Maximum memory usage	3.7966 GB
	10. Maximum processing power used	106.6%
Reliability		
Maturity	11. Simultaneous requests	1
Maintainability		
Modularity	12. Percentage of modularity	60%
Reusability	13. Percentage of reusable assets	100%

3.4. Port events detection from AIS Data

The AIS maritime communication standard allows static and dynamic operational information to be obtained from the ships. This information provides a better insight into port operations, as it can be seen in D4.4. Moreover, this information can be used to predict in advance events like the start of pilotage, towing, refueling and speed exceed, allowing logistics operations to be more efficient and more secure. In D4.4, section 3.1.3 described the algorithm developed to forecast in advance this event using AIS data. Random Forest and KNN were the algorithms used to solve the event classification problem. Even though, in the end, only Random Forest has been used during this evaluation process, since, as seen in table 6 of section 3.1.3 in D4.4, its precision results surpassed those obtained by KNN. Random Forest is an ensemble learning method based on the construction of multiple decision trees during the training stage. For classification problems, the result is the class selected by most decision trees. Its performance is superior to that of decision trees and is comparable to the use of gradient boosted trees. Some of the techniques used to create different independent decision trees are bootstrap aggregating or feature bagging, their use is widespread and their structure escapes the content of the deliverable. The version of Random Forest used has been the one implemented in the scikit-learn library. This version allows the modification of several hyperparameters that allow the best possible adjustment, some of these are the number of trees, their depth, the information gain criterion, as well as the possibility of enabling multi-thread training, since training can be a totally parallelizable procedure and, therefore, it makes possible to take advantage of the available resources.

3.4.1. Assessment scenario

Although this algorithm has not been included in any of the pilots, the evaluation of its performance will allow an estimation of the requirements for its incorporation and therefore facilitate its integration within the platform. This fact of not inclusion in any of the pilots has meant that some of the requirements mention in Table 8 are actually not applicable from a pure technical/operational point of view. The data source used in this predictive tool corresponds to the information obtained from the AIS maritime communication standard. Although for the training phase it has been necessary to pass said dataset through the existing rules within the Posidonia Operations tool, which allows the suite to detect such events. Posidonia Operations, as described in D4.4, is a real-time monitoring system for ship activity capable of detecting various events during the life of the ship at port, allowing the automation of actions and assistance to the port operator in the control of the visit of the ship to the port. Some of the existing fields in AIS messages and used by the predictive tool are latitude, longitude, speed over the ground, course over the ground and IMO.

The re-training of the model is not expected until a considerable amount of new historical data is available, so the normal execution of the algorithm will correspond to the inference made from the corresponding AIS message. The results presented here correspond to the training and test process, since the complexity of the model allows immediate inference, so it will be evaluated on the 30,250 records used during the training phase and the 3,400 in the test and validation phase.

The evolution of the performance has been carried out on a Lenovo ThinkPad E470 with the following configuration of technical specifications:

- OS: Windows 10 Pro 21H1
- CPU: Intel Core i5-7200 CPU (2 Cores at 2,50GHz and 4 Threads)

- RAM: 16 GB = 2x8GB DDR4
- Disk: 500GB SSD

3.4.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes.

1 Training phase: The training phase is fully automatic and is executed during the deployment of the predictive algorithm.

2 Usage phase: the prediction model has been implemented according to specifications for the integration in the Operational Tools and the Information Hub, thus supporting the deployment and usage of the model with the standard PIXEL API specification.

2. Portion of completed requirements: 50% (relevant). As indicated before, fact of not inclusion in any of the pilots has meant that 50% of the requirements mention in Table 8 are actually not applicable from a pure technical/operational point of view

We evaluated this KPI on all relevant Common functional requirements (Product: PIXEL platform or PA) directly targeting this PA. For all requirements, we stated if they were relevant or not for our specific PA, then we stated if we completed it (if relevant). Table below summarizes this assessment.

Table 8. Port events detection from AIS Data Common Functional Requirements status

Requirement	Completed	Comments
Import historical data (36)	Yes.	The algorithm is trained on historical data.
Homogenize data (61)	Yes.	The model is based on the standard AIS data structure.
Interaction with models (41)	No.	The model is not integrated in the PIXEL Information Hub. This is due to the fact there has not been the need to integrate the model deployed in a PIXEL instance (not part of the PIXEL pilots). Therefore this is not considered a shortcoming.
Operational Interface (71)	No.	The model does not provide an OT interface. Same rationale than the row above (not considered a shortcoming, but an operational decision).

A total of 4 requirements are related to the functionality provided by the AIS Event PA. Out of those, two have been implemented and tested. As there was no need to integrate the PA with Operational Tools, two requirements were not fulfilled. The reason for this is that in the final definition of the pilots, no use case considered the use of AIS Event detection, so finally no UI was developed in favor of other new cases initially not identified.

- Total requirements: 4.

- Fulfilled requirements: 2.
- Portion of completed requirements: 50%. As indicated before, fact of not inclusion in any of the pilots has meant that 50% of the requirements mention in Table 8 are actually not applicable from a pure technical/operational point of view

3. Maximum number of connected data sources: 1
Only one data source is connected at a time to every deployed instance of the model.

4. Maximum database size: No maximum.

The model does not present an upper limit for the size of the data set used for training. The limit resides on the memory resources of the machine where the training is performed and AIS messages are processed sequentially.

5. Average latency: 578.94s
Several training runs have been launched consecutively, obtaining an average time to complete the process of 578.94s. The duration of the training process depends on the size of the data set. For the inference process, it is possible to obtain estimations for the test dataset in 1,24s, again, multiple requests have been made to quantify a mean inference time.

6. Throughput: 52.25 records/s
These values were computed as the number of records in the training dataset (30,250 records) divided by the training time (578.94s), obtaining 52,25 records/s (training). For the inference process, the number of records during inference (3,400 records) divided by the inference time (1,24s), obtaining 2,741.93 records/s (inference).

8. Mean CPU Utilisation: 117.34% (training), 68.45% (inference).
These values have been obtained as the average of CPU resources required during all the simulations carried out, both during the training and inference processes. During the training and inference process, the CPU presents a value higher than 100% since the process can be carried out in multiple threads thanks to the use of the scikit-learn library.

7. Mean memory usage: 843,27 MB (training), 134,92 MB (inference).
These values have been obtained as the average of memory resources required during all the simulations carried out, both during the training and inference processes.

8. Maximum memory usage: 1191,72 MB (training), 241,87 MB (inference).
These values have been obtained as the average of maximum memory resources required during all the simulations carried out, both during the training and inference processes.

9. Maximum processing power used: 250% (training), 200% (inference).
These values have been obtained as the average of maximum CPU resources required during all the simulations carried out, both during the training and inference processes.

10. Simultaneous requests: 1
This prediction algorithm only handles one request at a time. Simultaneous requests are not considered.

11. % of modularity: 33%

The Predictive Algorithm consists of core modules, APIs and prediction models. The core modules and the prediction models are highly dependent so they can not be replaced without affecting each other. So, modularity is close to 33%, as only one module can be replaced independently.

12. % of reusable assets: 66%
Related to reusable assets, only prediction models can not be re-used between different use cases because they are built on historical data from a specific port. All other parts can be reused, so the percentage of re-usable parts is 66%.

Results for all KPIs are summarized in the table below :

Table 9. KPI summary for the Prediction of Port events detection from AIS Data model

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	50% (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	1
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	578.94s
	6. Throughput	52.25 records/s
Resource utilization	7. Mean CPU Utilisation	117.34% (training), 68.45% (inference).
	8. Mean memory usage	843,27 MB (training), 134,92 MB (inference).
	9. Maximum memory usage	1191,72 MB (training), 241,87 MB (inference).
	10. Maximum processing power used	250% (training), 200% (inference).
Reliability		

Maturity	11. Simultaneous requests	1
Maintainability		
Modularity	12. Percentage of modularity	33%
Reusability	13. Percentage of reusable assets	66%

3.5. Traffic predictions module – ASPM/SDAG

This predictive algorithm was presented in Sections 5.1.1 and 5.2.1 of Deliverable 4.4 (D4.4). Its implementation consists in a Docker image which retrieves historical data from Information Hub, automatically trains a model for each location, and infers the traffic conditions for a determined time span in the future.

3.5.1. Assessment scenario

For the technical evaluation, historical data equivalent to that in the real use cases was used. Since an independent prediction model is trained for each of the locations, it is expected that most of the resource consumption will happen during this process. The historical data has been filtered so that there is approximately the same number of entries for all locations. This resulted in 25 locations each with 34717.6 entries on average, being 28968 and 37272 the minimum and maximum, respectively. The performance of the inference process was also measured for all the 25 different locations. The submodule handling the communication with the Information Hub was not included in the evaluation, since the results may vary depending on network conditions. The performance evaluation script was executed in a Lenovo ThinkPad T14 Gen 1 with the following technical specifications:

- OS: Ubuntu 20.04.2 LTS 64 bits
- CPU: AMD Ryzen 7 Pro 4750U Processor (1.70 GHz, 8 Cores, 16 Threads, 8 MB Cache)
- RAM: 32 GB = 2x 16GB DDR4 (3200 MT/s)
- Disk: 512 GB SSD

CPU and memory measurements were carried out using the Python library psutil[7].

3.5.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes
This KPI verifies that (1) a process to build a model for a new port (training phase) is straightforward and documented; (2) A process to make predictions does not include unnecessary steps, i.e. is executed by one API call.

1. Training phase: The training phase is fully automatic and is executed during the deployment of the predictive algorithm.

[7] https://psutil.readthedocs.io/en/latest/

2 Usage phase: the prediction model has been implemented according to specifications for the integration in the Operational Tools and the Information Hub, thus supporting the deployment and usage of the model with the standard PIXEL API specification.

2. The portion of completed requirements: 100% (relevant)

Common functional requirements (Must Have, Should Have) from deliverable D3.2 will be taken as input in order to extract all requirements targeting this PA. We evaluated this KPI on all relevant *Common functional requirements* (Product: PIXEL platform or PA) directly targeting this PA. For all requirements, we stated if they were relevant or not for our specific PA, then we stated if we completed it (if relevant). The Table below summarizes this assessment.

Table 10. PIXEL PA Traffic Predictions Common Functional Requirements status

Common functional Requirements	Priority	Relevant	Completed	Comments
Import historical Data (36)	M	1	1	Model is trained on historical data.
Interaction with models (41)	M	1	1	The model provides an OT interface and is integrated in the platform.
Anomaly and event list (44)	M	0		The requirement refers to CEP implementation, not relevant.
Homogenize Data (61)	M	1	1	Agreed data model for traffic flow is used in the PA.
Catalogue of models (62)	M	1	1	The model provides an OT interface and is integrated in the platform.
Detection of anomalies (63)	M	0		Not relevant for this PA.
Feedback (64)	S	0		This hasn't to be dealt at the model level.
Centralized user administration system (65)	M	0		This hasn't to be dealt at the model level.
UI notification System (67)	M	0		This hasn't to be dealt at the model level.
Operational Interface (71)	M	1	1	The model provides an OT interface and is integrated in the platform.
Discovery service for data (104)	M	0		This hasn't to be dealt at the model level.

A total of 5 requirements are related to functionality provided by the Road Traffic PA. Out of those all have been implemented and tested.

- Total requirements: 5.
- Fulfilled requirements: 5.
- Portion of completed requirements: 100%.

3. Maximum number of connected data sources: 1

Only one data source is connected at a time to every deployed instance of the model.

4. Maximum database size: No maximum.

The model does not present a maximum as for the size of the dataset that can be used for training it. The limit would depend on the memory resources of the host machine in which the algorithm is executed.

5. Average latency: 152.8555 s (training) and 3.4793 s (inference).
These values are the average times for all the 25 locations. The value stands for the complete training process and the complete inference process. In the figure below we include a comparison between the number of records in the training dataset for a location and the time needed to train the model.

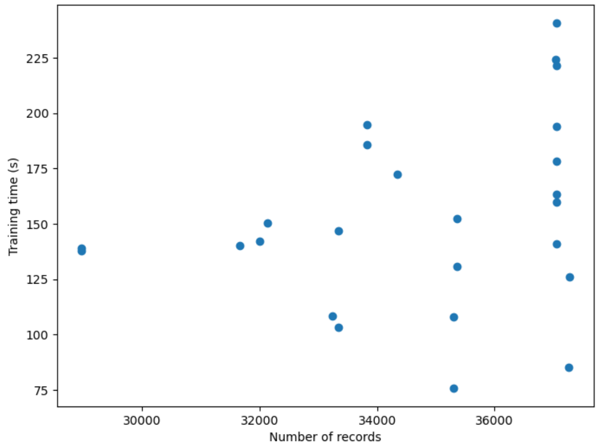

Figure 4. Total training time with respect to the number of records for different locations in the training dataset for the Traffic PA.

Given the similar number of records for all the 25 locations, no clear relation can be observed between the number of records in the training set and the time needed to train the model. It is reasonable to assume that with a higher variability in the number of records we could observe a relation between these variables, but in this case the training time varies depending solely on the particularities of the training data and not its size.

6. Throughput: 245.2112 records/s (training). 53.1443 records/s (inference).
These values were computed as the average for all locations of the number of records in the dataset divided by the training time, and the number of generated records during inference (168 in all cases = 7 days x 24 h) divided by the inference time, respectively.

7. Mean CPU Utilisation: 99.0006 % (training), 1361.7565 % (inference).
The values reported here are the average for all locations of the mean CPU utilization. During inference, the CPU utilization goes over 100 % because the process runs in several threads. Specifically, the Python library *numpy*[8] is used during the inference process, which uses multithreading to speed up the execution of some functions. The machine where the tests were performed has 16 threads, so the CPU utilization could be as high as 1600 %. The plots below show the CPU utilization during training (left) and inference (right) for one of the locations.

[8] https://numpy.org/

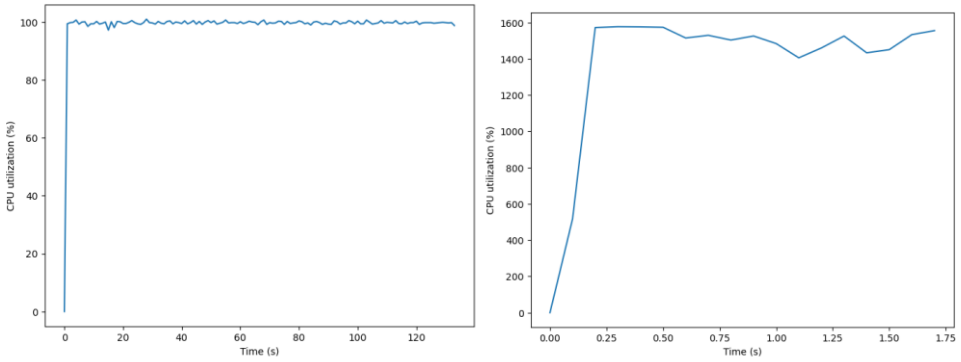

Figure 6. CPU utilization during training (left) and inference (right) when using the Traffic PA.

8. Mean memory usage: 416.53 MB (training), 31.64 MB (inference).

The values reported here are the average for all locations of the mean memory utilization. The plot below shows a comparison between the number of records in the training dataset for a location and mean memory used during the training process for its corresponding model. In this case, we do observe an increasing trend in the memory consumption with respect to the size of the training set.

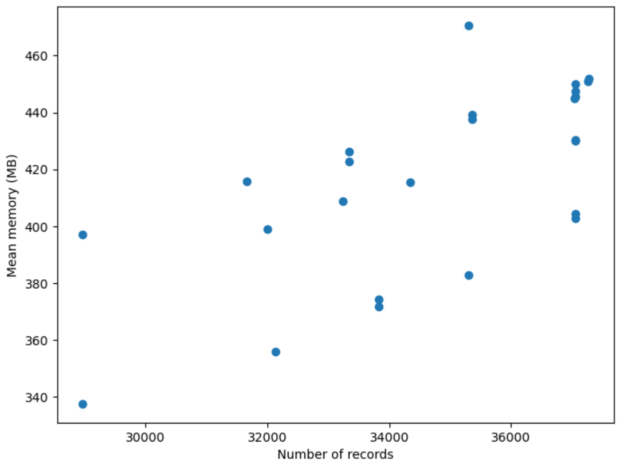

Figure 7. Mean memory usage during training with respect to the number of records for different locations in the training dataset for the Traffic PA.

9. Maximum memory usage: 480.02 MB (training), 88.75 MB (inference).

10. Maximum processing power used: 100 % (training), 1600 % (inference).

11. Simultaneous requests: 1

This prediction algorithm only handles one request at a time. The training and inference phases for the different locations within the same request -as it was the case in the evaluation carried out here- are made sequentially by design, so the values reported above still apply.

12. % of modularity : 33%

The PA consists of several core modules, APIs to integrate it in the PIXEL architecture and prediction models for ports. In principle, either core modules (the learning module and prediction algorithms) or the APIs (REST, OTs, IH) could be replaced independently. However, the models and core modules are highly interdependent and they cannot be replaced without affecting the other. If we assume that we have three main modules parts as explained above (core ML, APIs, Models) the % of modularity is 1/3 = 33%.

13. % of reusable assets: 66%
Of the three modules identified above (core ML, APIs, Models), models cannot be re-used among different use cases because they must be built on historical data for the concerned port. All other parts of the PA can be reused with no restrictions: 2/3=66%.

Results for all KPIs are summarized in the table below :

Table 11. KPI summary for the PIXEL PA Traffic Predictions model

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	100 % (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	1
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	152.8555 s (training) and 3.4793 s (inference).
	6. Throughput	245.2112 records/s (training). 53.1443 records/s (inference).
Resource utilization	7. Mean CPU Utilisation	99.0006 % (training), 1361.7565 % (inference).
	8. Mean memory usage	416.53 MB (training), 31.64 MB (inference).
	9. Maximum memory usage	480.02 MB (training), 88.75 MB (inference).
	10. Maximum processing power used	100 % (training), 1600 % (inference).
Reliability		

Maturity	11. Simultaneous requests	1
Maintainability		
Modularity	12. Percentage of modularity	33%
Reusability	13. Percentage of reusable assets	66%

3.6. Traffic predictions module – PPA

The Port of Piraeus is the main receiving port for cruise ships in the Mediterranean, so the impact that this type of tourism has on the surrounding traffic is quite noticeable. Therefore, the need arises to have a traffic behavior prediction algorithm that allows a better forecast of the evolution of this in order to avoid congestion and ensure its fluidity. This algorithm was initially described in D4.4 section 5.1.1 and 5.2.2. This algorithm has changed according to some feedback received by the PPA and other partners, to make it more suitable to the actual needs of the port. The algorithm currently predicts the traffic in some areas of the port according to historic data and some inputs such as weather and expected cruises. The description of the integration of the model, data sources used and visualization generated is provided in D7.2 (7.4.1). The predictive algorithm is based on Facebook Prophet (https://facebook.github.io/prophet/), which provides a general framework for time series data analysis and predictive modelling and can be easily integrated to the ports and the data that is commonly available there. From the historical data set, it is able to learn the components that make up the time series: its trend, the seasonal components and the random component whose variance cannot be modeled. Once these components are obtained, it is possible to estimate new values. In addition to the base information, Prophet allows the incorporation of additional attributes that explain part of the variability of the data and therefore obtain more precise estimates. Furthermore, although obtaining a standard solution is very simple, it has multiple configuration parameters that allow a better modeling of the available data.

3.6.1. Assessment scenario

As the information available from data sources is gathered daily (due to the configuration of the agents), this is the maximum meaningful resolution for the predictions. Thus, the model is scheduled to be executed daily.

Finally, the real-time traffic data provider service is HERE Traffic API. This service provides real-time traffic information via API from a rectangular area of interest, defined by the coordinates of its upper left and lower right corner. In the response, information has been obtained from different points within this area of interest, the presence of these POI being in the response of the API is variable, since information from the same points is not always included, so for the pilot it has been decided to select a point with the best balance between interest for the port and available data. In order not to exceed the maximum number of free requests to the API, calls are made every 15 minutes, this being the maximum frequency in the estimates made by the model. In addition to traffic information, another API service such as OpenWeather has also been used, from which to obtain meteorological information such as probability and intensity of precipitation, wind and temperature. This meteorological information will allow more precise estimates to be made as these variables have a direct impact on the intensity of traffic, being those days with poor weather forecasts in which congestion increases and fluidity decreases. This information is also collected every 15 minutes, coinciding with the temporal accuracy of the traffic information. Besides, in terms of port activity, the types of vessels that according to the port authority have the most impact on traffic (i.e. cruise ships), have been considered as well. So, the number of cruises and their capacity, the number of buses they require and the number of total passengers arriving at the port have been used in order to increase accuracy of estimations. Once all the data is obtained, data treatment processes developed in Python are carried out that allow cleaning and transforming the data to the optimal format. Once a day, the models are retrained with the most up-to-date data history available and estimates are made that will be displayed by the platform. The different versions of the model, as well as the different sources of data considered, have been subject to multiple configurations according to the feedback received from the specialist people of the Port Authority, a continuous improvement procedure that will continue until the end of the project. The location selected from among those available for the development of the prediction model is described below. The specific area selected corresponds to the suburb of Δραπετσώνα, located in the southwestern part of the Athens agglomeration, and immediately next to the port, its arteries being the main transit routes from the port to the outside. The evolution of the performance has been carried out on a Lenovo ThinkPad E470 with the following configuration of technical specifications:

- OS: Windows 10 Pro 21H1
- CPU: Intel Core i5-7200 CPU (2 Cores at 2,50GHz and 4 Threads)
- RAM: 16 GB = 2x 8GB DDR4
- Disk: 500GB SSD

3.6.2. KPI Data Collection and Results

Considering that the traffic prediction module has been developed for the three pilots, and that some of the key performance indicators are difficult to express quantitatively, an expert judgement approach has been used where the qualitative expression is more efficient.

According to the assessment KPIs identified in D8.1, these are the results collected so far.

1. Straightforward task accomplishment: Yes.

1 Training phase: The training phase is fully automatic and is executed during the deployment of the predictive algorithm.

2 Usage phase: the prediction model has been implemented according to specifications for the integration in the Operational Tools and the Information Hub, thus supporting the deployment and usage of the model with the standard PIXEL API specification.

2. Portion of completed requirements: 100% (relevant)

We evaluated this KPI on all relevant *Common functional requirements* (Product: PIXEL platform or PA) directly targeting this PA. For all requirements, we stated if they were relevant or not for our specific PA, then we stated if we completed it (if relevant).

Table 12. List of completed requirements for the PPA traffic prediction module

Requirement	Completed
Monitor expected port calls (11)	Yes. The algorithm gets the number of public buses from the data of the PPA vessel calls.

Port congestion forecasting (34)	Yes.
Port - City road congestion forecasting (91)	Yes.

A total of 3 requirements are related to functionality provided by the Road Traffic PA. Out of those all have been implemented and tested.

3. Maximum number of connected data sources: 3
We can load data from files belonging to Port Authority data, as well there are two more data services which provide external information: Here Traffic API and Weather API.

4. Maximum database size: No maximum.
The model does not present an upper limit for the size of the data set used for training. The limit resides on the memory resources of the machine where the training is performed.

5. Average latency: 234.57s.
For the selected location, ten training runs have been launched consecutively, obtaining an average time to complete the process of 234.57s. The duration of the training process depends to a greater extent on the variability of the data and therefore on the complexity to learn the components of the time series and not so much on the size of the data set. For the inference process, it is possible to obtain estimations in 2,14s, again, multiple requests have been made to quantify a mean inference time.

6. Throughput: 96,19 records/s (training) and 224.30 records/s (inference)
These values were computed as the number of records in the dataset (22,56 records) divided by the training time (234.57s), obtaining 96,19 records/s (training). For the inference process, the number of generated records during inference (480 records = 5 days x 24 h x 4 (15min accuracy) divided by the inference time (2.14s), obtaining 224.30 records/s (inference).

7. Mean CPU Utilisation: 97.08% (training), 127.45% (inference).
These values have been obtained as the average of CPU resources required during all the simulations carried out, both during the training and inference processes. During the inference process, the CPU presents a value higher than 100% since the process can be carried out in multiple threads thanks to the use of the NumPy library.

8. Mean memory usage: 470,54 MB (training), 58,82 MB (inference).
These values have been obtained as the average of memory resources required during all the simulations carried out, both during the training and inference processes.

9. Maximum memory usage: 595,78 MB (training), 94,37 MB (inference).
These values have been obtained as the average of maximum memory resources required during all the simulations carried out, both during the training and inference processes.

10. Maximum processing power used: 100% (training), 400% (inference).
These values have been obtained as the average of maximum CPU resources required during all the simulations carried out, both during the training and inference processes.

11. Simultaneous requests: 1

This prediction algorithm only handles one request at a time. The training and inference phases for the different locations within the same request are made sequentially by design, so the values reported above still apply.

12. % of modularity: 33%

The Predictive Algorithm consists of core modules, APIs and prediction models. The core modules and the prediction models are highly dependent so they can not be replaced without affecting each other. So, modularity is close to 33%, as only one module can be replaced independently.

13. %reusable assets: 66%

Related to reusable assets, only prediction models can not be re-used between different use cases because they are built on historical data from a specific port. All other parts can be reused, so the percentage of re-usable parts is 66%.

Results for all KPIs are summarized in the table below :

Table 13. KPI summary for the PPA Traffic predictions module

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	100% (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	3
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	234.57s.
	6. Throughput	96,19 records/s (training) and 224.30 records/s (inference)
Resource utilization	7. Mean CPU Utilisation	97.08% (training), 127.45% (inference).
	8. Mean memory usage	470,54 MB (training), 58,82 MB (inference).
	9. Maximum memory usage	595,78 MB (training), 94,37 MB (inference).

	10. Maximum processing power used	100% (training), 400% (inference).
Reliability		
Maturity	11. Simultaneous requests	1
Maintainability		
Modularity	12. Percentage of modularity	33%
Reusability	13. Percentage of reusable assets	66%

3.7. Traffic predictions module – THPA

This predictive algorithm was described in D4.4 section 5, and for THPA subchapters 5.1.2 and 5.2.3 showed the methodology and (initial) results. It is able to predict the traffic in specific points based on past historical data and follows a FIWARE data model approach called TrafficFlowObserved with some adaptations. The algorithm estimates future traffic in two different units depending on how historical data is provided:

- Intensity or Average speed

For THPA, data is given in terms of intensity and the special points selected are the main traffic gates at the port: gate 10A and gate 16.

Figure 8. Port of Thessaloniki. Gates 10A and 16

The predictive algorithm is based on Facebook prophet (https://facebook.github.io/prophet/) which is trained before every execution with historical data. This is probably the part that consumes more computational resources.

3.7.1. Assessment scenario

We use the code available in our internal (PIXEL) github repository: https://gitpixel.satrdlab.upv.es/xlab/traffic_pa/src/master. The code includes a Dockerfile able to generate a Docker which will be executed for the tests. The Docker container is also in our internal (PIXEL) Docker repository (docker.pixel-ports.eu/traffic_pa:1.2). Using the Operational Tools of the PIXEL platform in THPA we were able to publish the predictive algorithm by importing the Docker image. Once published, the algorithm can be executed once or it can be scheduled; the latter one is the typical case in a production environment. It can be executed hourly (if there is not much load on the corresponding servers) or daily (in case of medium or high load on the corresponding servers). For the tests, we will configure it hourly and get result data from these executions. All this information is provided in terms of an input JSON file including all needed information.

- For the schedule, we just set it to be run every hour
- For the Prophet information, we just include (i) the horizon (number of days to estimate traffic information starting from the current date) as well as (ii) the country – GR for THPA- to consider holidays, and (iii) the type of target (intensity for THPA)
- For the historical data, these are available through an index in the Information Hub (Elasticsearch database). The predictive algorithm supports a connector (es-api) to retrieve such data.

Traffic estimation, in general, is done based on following input parameters:

- RFID (Radio-frequency identification) traffic data at the gates
- Weather
- Traffic at the city
- Vessel calls

For THPA, an analysis was made to investigate the correlation of the different inputs with the accuracy of the estimation based on past (historical) results, but no significant result was observed for the last inputs, even if available. Therefore, only past data had been taken into account and no special regressors for considering all possible inputs were included. Note that, for other ports, this could not be the case. An additional aspect to consider is that real traffic data obtained via RFID at the port gates are given every 15 minutes, whereas Facebook Prophet is only able to provide estimations every hour. For this, we will need to work with average values every hour whenever the accuracy is being tested. The predictive algorithm will be executed in the PIXEL platform deployed in THPA. More specifically, it is in the CORE VM, with following properties:

- 4 cores,
- 16 GB RAM,
- 300 GB HDD

The CORE VM includes several Dockers from the PIXEL platform, and runs the models or predictive algorithms published there. For that reason, we will first check the average background load to compare it with the times the algorithm is executed. Furthermore, we will stop any other executions from other possible models and predictive algorithms that may be running in the background so as to not disturb the results.

3.7.2. KPI Data Collection and Results

Considering that the traffic prediction module has been used in all the three pilot ports (in Monfalcone and in THPA the same, in PPA some add-ons were introduced), we will follow a similar report structure and will be using an expert judgement with the same sections.

1. Straightforward task accomplishment: Yes (same result as for Monfalcone).
This KPI verifies that (1) a process to build a model for a new port (training phase) is straightforward and documented; (2) A process to make predictions does not include unnecessary steps, i.e. is executed by one API call.

1 Training phase: The training phase is fully automatic and is executed during the deployment of the predictive algorithm.
2 Usage phase: the prediction model has been implemented according to specifications for the integration in the Operational Tools and the Information Hub, thus supporting the deployment and usage of the model with the standard PIXEL API specification.

2. The portion of completed requirements: 100% (relevant)
Common functional requirements (Must Have, Should Have) from deliverable D3.2 will be taken as input in order to extract all requirements targeting this PA. We evaluated this KPI on all relevant *Common functional requirements* (Product: PIXEL platform or PA) directly targeting this PA. For all requirements, we stated if they were relevant or not for our specific PA, then we stated if we completed it (if relevant). Here the same table as for Monfalcone summarizes the assessment (see Section 3.5.2).

A total of 5 requirements are related to functionality provided by the Road Traffic PA. Out of those all have been implemented and tested (same result as for Monfalcone).

5. Maximum number of connected data sources: 1
Only one data source is connected at a time to every deployed instance of the model (same result as for Monfalcone).

6. Maximum database size: No maximum.
The model does not present a maximum as for the size of the dataset that can be used for training it. The limit would depend on the memory resources of the host machine in which the algorithm is executed. The result is the same as for Monfalcone; the difference here is that we feed the algorithm with historical data from THPA. We were able to retrieve/import data from several years, and started from 2020 for the two gates in THPA (gate 10A and 16). As the data is inserted hourly, there should be no database issue here, even including 10 years, which is, by far, much more than the PA needs.

7. Average latency: 120 s (training) and 2 s (inference)
These values are the average times for the 2 locations (gates). The value stands for the complete training process and the complete inference process. We rounded the values to seconds and obtained those by checking the logs and timestamps during the execution of scheduledInstance in a day-interval (every hour). Values differ from Monfalcone and the environment and data are different.

8. Throughput: 83.3 records/s (training). 60 records/s (inference).

These values were computed as the average for both locations/gates of the number of records in the dataset divided by the training time, and the number of generated records during inference (120 in both cases/gates = 5 days x 24 h) divided by the inference time, respectively. The approach is the same as for Monfalcone, but the numbers differ, because the (i) average latency differs, (ii) we have different amount of historical data in the IH, and (iii) we are producing values for the next 5 days, instead of the next week.

Note: The following values for Mean and Maximum CPU and memory usage have been difficult to estimate, as the predictive algorithm was tested directly on the THPA platform and therefore, other services and processes were also running. Therefore, the given values are more qualitative than quantitative, as we couldn't isolate the single execution.

9. Mean CPU Utilisation: 90 % (training), 150 % (inference).
The values reported here are the average for both locations of the mean CPU utilization. During inference, the CPU utilization goes over 100 % because the process runs in several threads. Specifically, the Python library *numpy* is used during the inference process, which uses multithreading to speed up the execution of some functions. The machine where the tests were performed has 4 CPUs, with one core per socket (and one thread per core), so it could potentially go to 400%.

10. Mean memory usage: 380 MB (training), 20 MB (inference).
The values reported here are the average for both locations of the mean memory utilization.

11. Maximum memory usage: 450 MB (training), 31 MB (inference).

12. Maximum processing power used: 100 % (training), 200 % (inference).

13. Simultaneous requests: 1
This prediction algorithm only handles one request at a time. The training and inference phases for the different locations within the same request -as it was the case in the evaluation carried out here- are made sequentially by design, so the values reported above still apply (same result as for Monfalcone).

3. % of modularity: 33% (same result as for Monfalcone).
The PA consists of several core modules, APIs to integrate it in the PIXEL architecture and prediction models for ports. In principle, either core modules (the learning module and prediction algorithms) or the APIs (REST, OTs, IH) could be replaced independently. However, the models and core modules are highly interdependent and they cannot be replaced without affecting the other. If we assume that we have three main modules parts as explained above (core ML, APIs, Models) the % of modularity is 1/3 = 33%.

4. % of reusable assets: 66%
Of the three modules identified above (core ML, APIs, Models), models cannot be re-used among different use cases because they must be built on historical data for the concerned port. All other parts of the PA can be reused with no restrictions: 2/3=66%.

66% of assets can be reused in different port scenarios. The result is the same as for Monfalcone; the difference here is that we feed the algorithm with historical data from THPA. We were able to retrieve/import data from several years, and started from 2020.

Results for all KPIs are summarized in the table below :

Table 14. KPI summary for the THPA Traffic predictions module

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	100% (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	1
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	120 s (training) and 2 s (inference)
	6. Throughput	83.3 records/s (training). 60 records/s (inference).
Resource utilization	7. Mean CPU Utilisation	90 % (training), 150 % (inference).
	8. Mean memory usage	380 MB (training), 20 MB (inference).
	9. Maximum memory usage	450 MB (training), 31 MB (inference).
	10. Maximum processing power used	100 % (training), 200 % (inference).
Reliability		
Maturity	11. Simultaneous requests	1
Maintainability		
Modularity	12. Percentage of modularity	33 %
Reusability	13. Percentage of reusable assets	66 %

3.8. Prediction of renewable energy production

This model, introduced in D4.4 section 6, predicts renewable energy production of photovoltaic panels using historical data. As no port in the project has access to photovoltaic panels yet, we gathered data from PVoutput and evaluated multiple solutions for the prediction. Training an LSTM Neural Network was the chosen solution for the prediction. As the model was trained for PVoutput data, we had to keep in mind that re-training is part of the module as it will be done in each port. Indeed, model training is the most responsible for resource consumption in this module, as model inference is quite fast and only necessitates to load the model in RAM. In this evaluation, we will mainly focus on training performances but also give insights on model inference.

3.8.1. Assessment scenario

We use the code available in the github repository :

https://github.com/pixel-ports/PV_prod_predic/blob/master/MODEL_PREDIC_BENCHMARK/PV_Prediction.ipynb.

As this repository shows the exhaustive list of models trials, we will only run the 2 following scenarios :

- Train the LSTM model, implying the following steps:
    - Loading the full dataset.
    - Pre-processing the data.
    - Writing the preprocessing scaler to disk.
    - Training the model.
    - Writing the model to disk.
- Infer the LSTM model on new data, implying the following steps:
    - Loading only 1 sequence of data.
    - Loading the preprocessing scaler from disk.
    - Pre-processing the data.
    - Loading the previously trained model.
    - Inferring the model.

The model does not aim to run continuously but instead to be run per call. So, trying to be as close as the production model, we take into our calculations the time to start the python interpreter. Two data files are used as inputs for the runs :

- The file one_sequence.csv which contains 49 lines of data.
- The file multiple_sequences.csv which contains 2587 lines of data. This file contains approximately 7 years of data, with one entry per day. We won't generate nor simulate more data because it is unexpected that we will have an order difference compared to our 7 years of data.

As the neural network implemented is not heavy, we will run it on a DELL Latitude E5570 machine with the following specs given for reproducibility:

- OS: Ubuntu 20.04.1 LTS 64 bits.
- CPU: Intel® Core™ i5-6300U CPU @ 2.40GHz × 4.

- GPU: Mesa Intel® HD Graphics 520 (SKL GT2).
- RAM: 2 * 8192MB DDR4 with a 2133 MT/s clock speed.
- Drive: SSD model CT1000MX500SSD4.

3.8.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes.
The model itself only runs the needed data which are detailed in the section above. However, the way the PIXEL platform starts this model, with Dockerization, forces us to reload the trained LSTM model at every call. An improvement is suggested in the simultaneous requests KPI section.

2. Portion of completed requirements: 70% (relevant) (however, these “missing” requirements have been dealt with via the OTs or Dashboard correspondingly) – see below.
We evaluated this KPI on all common functional requirements defined in D3.2. For all requirements, we stated if they were relevant or not for our specific model, then we stated if we completed it. Table below summarizes this assessment.

Table 15. List of completed requirements for the Prediction of renewable energy production model

Common functional Requirements	Priority	Relevant	Completed	Comments
Import historical Data (36)	M	1	1	Model is trained on historical data.
Interaction with models (41)	M	1	1	Data coming from PVGIS have been compared with the PAS output and are available through the PIXEL Dashboard.
Anomaly and event list (44)	M	1	1	We can have a list of errors in historical data during training.
Homogenize Data (61)	M	1	1	Data is pre-processed during training.
Catalogue of models (62)	M	1/0	0	We haven't done the Dockerization as no ports have access to PV data yet. However, this requirement is dealt with via the OTs.
Feedback (64)	S	0		This hasn't to be dealt at the model level.
Centralized user administration system (65)	M	0		This hasn't to be dealt at the model level.
Configurable Dashboard (66)	M	0		This hasn't to be dealt at the model level.
UI notification System (67)	M	0		This hasn't to be dealt at the model level.
Port Operational KPI list (70)	M	1	1	This model gives a prediction that can

				be used as a KPI.
Operation Interface (71)	M	0		This hasn't to be dealt at the model level.
Analyze historical data (81)	M	1	1	Model is trained on historical data.
Support for manually provided data (86)	M	1	1	Data can be loaded through a csv file.
Discovery service for data (104)	M	0		This hasn't to be dealt at the model level.
Visualization of data (105)	M	1	1	Visualizations are provided.

3. Maximum number of connected data sources: No limit.
As we load data from files, we can have any number of data sources.

4. Maximum database size: No maximum.
Data doesn't need to be loaded all at once, so there is no maximum in the database size.

5. Average latency: 78.7s (training), 1.73s (inference).
We conducted multiple runs of the inference scripts and found out that the mean total time is 1.73s for inference and 78.7s for training..

6. Throughput: 2.27KB/s (training)
We use the time tool of ubuntu to time the execution of the model and get the real execution time. We took 3 different runs:

- Training: With the multiple_sequences.csv file which is 181KB in size. The run took 79.6s to execute, so it is a 2.27KB/s throughput.
- Inference: One with the one_sequence.csv file which is 3.7KB in size. The run took 1.72s to execute, so it is a 2.15KB/s throughput.
- Inference: One with the multiple_sequences.csv file which is 181KB in size. The run took 1.83s to execute, so it is a 99.12KB/s throughput.

However, as shown by the two inference runs, the inference throughput increases with the input size. This is due to the start of the python process to be independent of data in input, so it doesn't make sense to give a throughput for inference.

7. Mean CPU Utilisation: 70.75% (training), 32.11% (inference).
We use python subprocess to start a new process for the model run only and use the psutil python package to measure CPU utilization over time. Psutil gives a per-core usage in percentages, which sums to 1200% with our configuration, which we bring back to a 0-100% scale. Results are shown below.

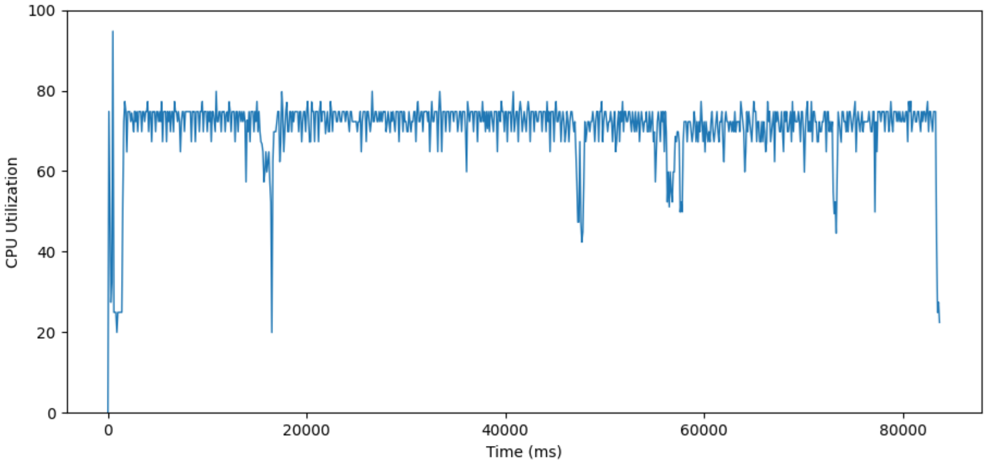

Figure 9. CPU Utilization over time for the Prediction of renewable energy production model during training

We see from the plot and from the recorded data that over the 80 seconds run, CPU during training has a mean utilization of 70.75%.

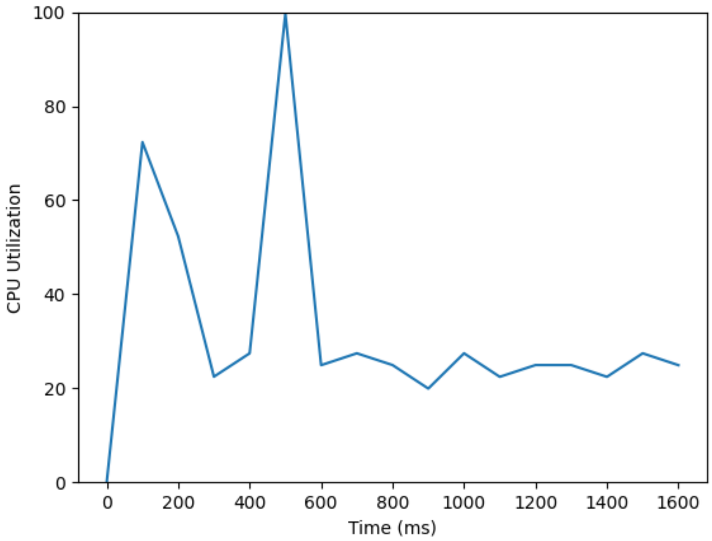

Figure 10. CPU Utilization over time for the Prediction of renewable energy production model during inference

We see from the plot and from the recorded data that over the 1.6 seconds run, CPU during training has a mean utilization of 32.11%.

8. Mean memory usage: 1.96GB (training), 28.72MB (inference).
We use the massif tool of valgrind in order to monitor the execution of our python script, which gives memory evolution along a single run of the module. Results of the training run is shown below :

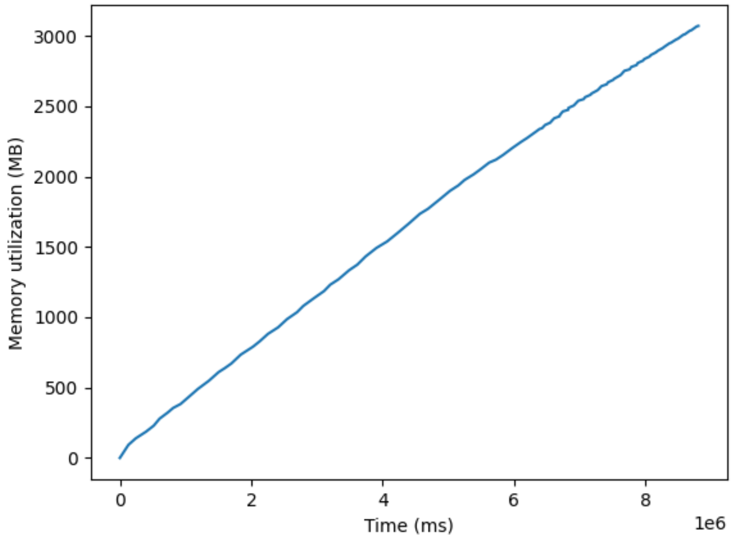

Figure 11. Memory Utilization over time for the Prediction of renewable energy production model during training

We also have a csv output with those values, one value per 0.1 seconds. It allows us to calculate the mean memory usage : 1.96GB. We can see that memory is constantly growing over time. Even if we don't expect to have more than 7 years of data to train our model, and then a bigger memory consumption, this indicates a memory leak that has to be corrected. We conduct the same analysis during inference:

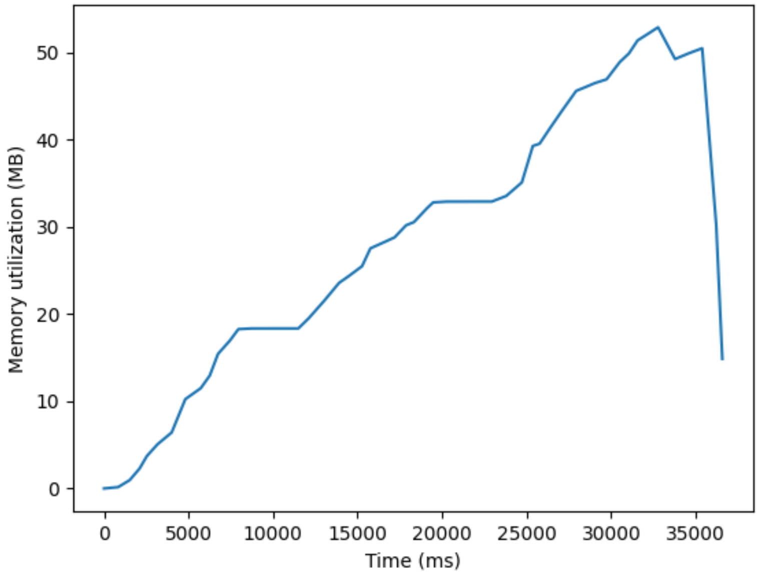

Figure 12. Memory Utilization over time for the Prediction of renewable energy production model during inference

It allows us to calculate the mean memory usage during inference: 28.72MB.

9. Maximum memory usage: 3.07GB (training), 52.92MB (inference).
The same method as above is applied. We find the maximum memory usage to be :

- For training: 3.07GB.

- For inference: 52.92MB.

10. Maximum processing power used: 319.20 (training), 309.30 (inference).
Using the same method as for the mean CPU Utilisation, we keep the summed data and don't divide it with the number of probes. Results are shown below :

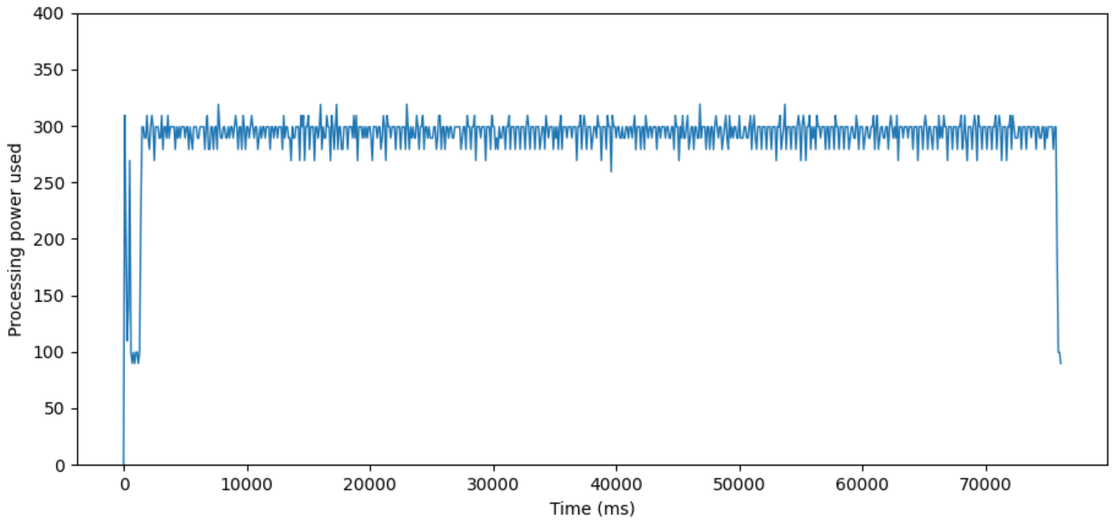

Figure 13. Processing power used over time for the Prediction of renewable energy production model during training

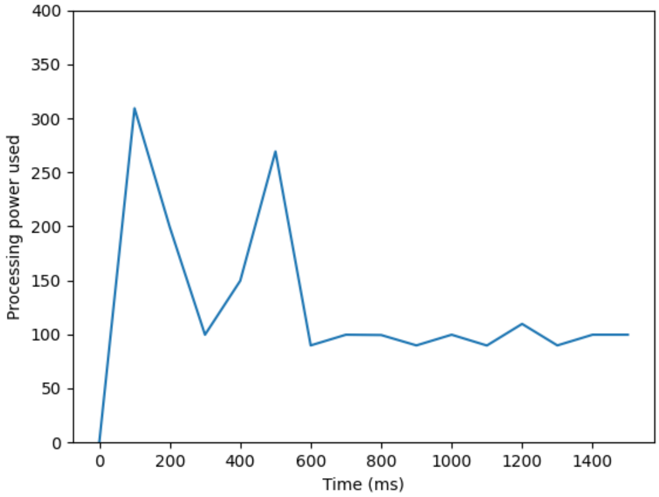

Figure 14. Processing power used over time for the Prediction of renewable energy production model during inference

We see from the plots and from the data collected, that the maximum processing power used is :

- 319.20 for training.
- 309.30 for inference.

11. Simultaneous requests: Linear.
We used Flask to start a python REST HTTP API, and used a custom script to send an increasing number of requests at the same time: We send 1 request, wait for it to finish then take the total time, we then send 5 requests,

wait for them to finish and take the total time, … up to 50 requests. We measured only inference time, because training doesn't aim to be called by the end user. First graph below shows the results of the inference script.

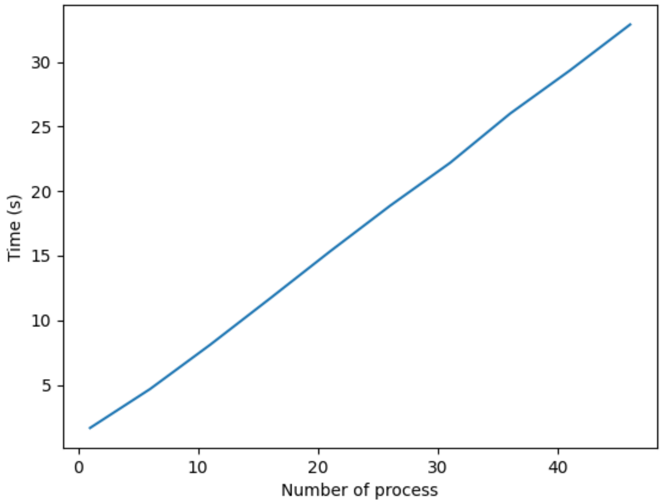

Figure 15. Total execution time of multiple requests for the Prediction of renewable energy production model during inference

We see that the total time is increasing linearly. Indeed flask starts a thread per incoming request, so this is the intended behaviour as our 4 cpu are quickly busy. Whereas it is not how a model is called in the PIXEL platform, we also monitored simultaneous requests in a way that inference is done with the preloaded model. While it is still showing a linearly increasing time, performance is far better.

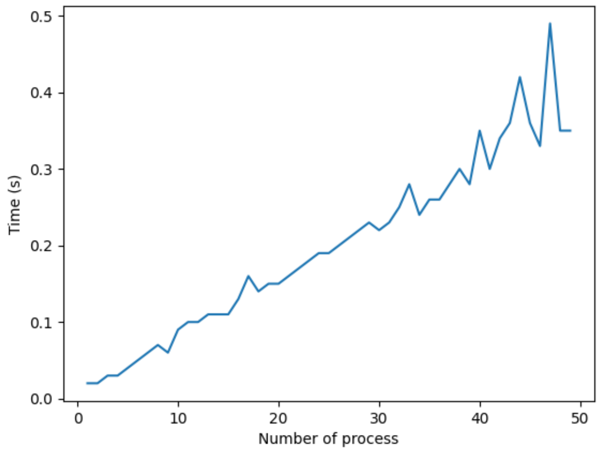

Figure 16. Total execution time of multiple requests for the Prediction of renewable energy production model during inference with a preloaded LSTM model

12. Percentage of modularity: 30%.
This model contains 3 components:

- The training data, which can be reused to train another model.
- The trained scaler, which is dependent on the training data.
- The trained model, which needs the scalar to operate properly.

13. Percentage of reusable assets: 100%.
The exploration step to select the better model has highlighted multiple ways of achieving prediction results. All this work is generic and can be reused on time series data.

Results for all KPIs are summarized in the table below :

Table 16. KPI summary for the Prediction of renewable energy production model

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	70% (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	No limit
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	78.7s (training) 1.73s (inference)
	6. Throughput	2.27KB/s (training)
Resource utilization	7. Mean CPU Utilisation	70.75% (training) 32.11% (inference)
	8. Mean memory usage	1.96GB (training) 28.72MB (inference).
	9. Maximum memory usage	3.07GB (training) 52.92MB (inference)
	10. Maximum processing power used	319.20 (training) 309.30 (inference)
Reliability		
Maturity	11. Simultaneous requests	Linear
Maintainability		

Modularity	12. Percentage of modularity	30%
Reusability	13. Percentage of reusable assets	100%

3.9. PEI software module

The evaluation carried out to select the better model has highlighted multiple ways of achieving prediction results. The PEI aims at calculating a single environmental impact metric of a port via obtaining a composite indicator gathering information from various environmental aspects of a port. The idea is to retrieve heterogeneous information (air pollution, wastes, noise) and compile them together, making the index comparable in time and between ports to assess their ranked environmental impact. In addition, the PEI also includes the calculation of a second composite index called "Reliability Rating" (RR) or "IoT-Readiness level" (IoTRL), that aims at representing how close to "total automation and digitalisation" the data collection process is in each PEI execution.

As presented in D5.3, the structure of the calculation of the PEI is divided in two blocks: (i) obtaining the "minimum expression of environmental impact", that consists of populating a series of environmental-Key Performance Indicators (eKPIs) that will be later processed to calculate the PEI. This execution directly modifies/interprets/filters/pre-processes data gathered from sensors or from other accepted data sources. (ii) Calculating the PEI indicator. This execution consists of (primarily) a two-sided composite indicator calculation procedure, going from the "leaf nodes" of a hierarchical tree to the "root node", applying at each step various mathematical operations such as normalization, aggregation and/or weighting. Whereas the first calculation block takes place at the NGSI agents (and is highly dependent to the data provided by each port) the PEI itself (as a PIXEL tool/software module) runs similarly to the rest of models in this section (managed/scheduled by the Operational Tools), so it can be properly assessed in this document. Therefore, the second block is what the WP5 team considered "the PEI as a model", which is the software that is evaluated in the context of this deliverable. This evaluation (next subsections) focuses on the execution of such a "PIXEL model" under two specific scenarios.

3.9.1. Assessment scenario

The code used in the evaluation is the final version (M37 of the project) of the PEI, which consists of a containerised Java program. The software project has not been released yet as open source as the exploitation strategy is still being discussed in the context of WP9. As with the rest of PIXEL models, PEI must be executed within PIXEL via the Operational Tools after deploying the containerised software.

The PEI is, per definition, a retrospective metric. Meaning that, for a specific moment of time, the result of the PEI is calculated for a previous time period. The exact past range depends on the amount/granularity of data provided by the port for the calculation. If all eKPIs needed to execute the PEI bring "fresh" data each week, the PEI could be calculated once per week (e.g., from a Sunday till the previous Saturday). Drawing from the execution of WP5 and T7.5, usual cases in maritime ports are monthly, quarterly or yearly, being the latter the safest approach to ensure proper, useful, solid results. Considering that, it was decided that for the technical evaluation to take place, two scenarios could be built: a monthly execution of the PEI and a yearly execution of the PEI. The "quarterly" scenario would be matched to a three-time running of the monthly case. The main difference among those cases is the amount of data covered by each of them. The inner PEI (composite indices) calculations are mostly the same, only increasing the number of data retrieval (initial) operations.

The scenarios assessed in this deliverable are, therefore, the following:

1. Execute the PEI model covering the previous month (monthly PEI). The execution entails:
   - 1.a Retrieving one month of data from the Information Hub of PIXEL.
   - 1.b Normalization of 32 eKPIs
   - 1.c Calculation of PEI (leaf-to-root tree)
   - 1.d Calculation of RR (leaf-to-root tree)
   - 1.e Generating a PDF report.
   - 1.f Preparing the output for the visualization, including associated recommendations.
2. Execute the PEI model covering one year of data (yearly PEI). The execution entails:
   - 2.a Retrieving one year of data from the Information Hub of PIXEL.
   - 2.b Normalization of 32 eKPIs
   - 2.c Calculation of PEI (leaf-to-root tree)
   - 2.d Calculation of RR (leaf-to-root tree)
   - 2.e Generating a PDF report.
   - 2.f Preparing the output for the visualization.

As indicated previously for other models, the PEI has been executed (in the two scenarios) as a standalone container (outside the PIXEL framework) using Docker. The computer used for conducting this evaluation has the following characteristics:

- OS: Windows 10 Education
- CPU: Intel Core i5 7400 (CPU 3GHz)
- GPU: Integrated Intel HD Graphics 630
- RAM: 16GB
- Drive: SSD 256GB + HDD 1TB
- JVM used in the container: OpenJDK 8

3.9.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes.
The model of PEI runs according to the scenario definition, gathering one month and one year of data from the Information Hub and producing a JSON output including the resulting value of all eKPIs, the sub-indices expected and the PEI. The JSON result also includes the RR final and partial values and the normalization details that have been applied. As mentioned, this assessment does not include the computing resources needed to obtain the eKPIs in the first place, that is overtaken by the NGSI agents. A parallel study could be done for those, but the results would be hugely different from one port to another.

2. Portion of completed requirements: 100% (relevant)
Here below is the evaluation of the PEI executions based on all common functional requirements defined in D3.2. For all requirements, it is argued whether they were relevant or not for the PEI specific model, then it is indicated whether they were completed, accompanied by a quick comment. Table below summarises such an assessment.

Table 17. List of completed requirements for the PEI software module

Common functional Requirements	Priority	Relevant	Completed	Comments
Import historical Data (36)	M	1	1	PEI executes over past data, depending on the scenario. It is capable to retrieve the data range indicated over the Information Hub.
Interaction with models (41)	M	1	1	One agent (not evaluated) retrieves the output of the PAS as input data for terminal air pollution eKPIs.
Anomaly and event list (44)	M	1	1	The recommendations engine within the PEI model triggers according to a pre-established list of conditions.
Homogenize Data (61)	M	1	1	NGSI agents convert data into the eKPI format before the execution of the PEI as a PIXEL model.
Catalogue of models (62)	M	0		The model is Dockerised.
Detection of anomalies (63)	M	0		This hasn't to be dealt at the model level.
Feedback (64)	S	0		This hasn't to be dealt at the model level.
Centralized user administration system (65)	M	1	1	The user sees the results and configures the tree/math.methods using the UI developed for the PEI:
Configurable Dashboard (66)	M	0		This hasn't to be dealt at the model level.
UI notification System (67)	M	0		This hasn't to be dealt at the model level.
Port Operational KPI list (70)	M	1	1	All elements in the output JSON can be considered measurable KPIs.
Operation Interface (71)	M	0		This hasn't to be dealt at the model level.
Analyze historical data (81)	M	1	1	PEI executes over past data, depending on the scenario. It is capable to retrieve the data range indicated over the Information Hub.
Support for manually provided data (86)	M	1	1	Yes, some data is allowed to be introduced manually through the UI.
Discovery service for data (104)	M	0		This hasn't to be dealt at the model level.
Visualization of data (105)	M	1	1	Users can see the PEI results in a graphic way, they can obtain the results in a PDF report and they can check recommendations for improving the index values..

3. Maximum number of connected data sources: No limit.

As the data come from NGSI agents, the PEI can be "connected" to any number of agents providing eKPI values.

4. Maximum database size: No maximum.
Depending on the server where PIXEL would be running (or, if standalone, where the PEI would be running).

5. Average latency: 4s (monthly), 5s (yearly).
Multiple runs of the model were conducted, finding that the mean total time is 4s for a monthly PEI calculation and 5s yearly PEI calculation.

6. Throughput: 13.925 kB/s (monthly), 11.14 kB/s (yearly)
HTTP requests data size.

- Scenario 1:
  - PEI Tree: 1.46 kB
  - RR Tree: 1.02 kB
  - 213 eKPI entries as input : 9.81 kB (monthly)*
  - eKPI normalization: 311 B per eKPI
  - RR dataPiece element: 479 B per dataPiece
  - PEI output: 44.5 kB
  - RR output: 11.2 kB

  * There are 45 eKPIs (as indicated before), but the periodicity of updates of each of those can be different. Coming from the agents, some eKPIs bring only one value in 1 month (e.g., fishing waste) and others are updated daily (e.g., average noise level at night). Therefore, in the period of 1 month, in the scenario used (Thessaloniki Port 2019), there were 213 pieces of information associated with 45 eKPIs.

- Scenario 2:
  - PEI Tree: 1.46 kB
  - RR Tree: 1.02 kB
  - 2460 eKPIs input: 19.04 kB (yearly). * Same explanation as above, but for 1 year.
  - eKPI normalization: 311 B per eKPI
  - RR dataPiece element: 479 B per dataPiece
  - PEI output: 44.5 kB
  - RR output: 11.2 kB

Summing all the outputs:

- Throughput (monthly) = 13.925 kB/s
- Throughput (yearly) = 11.14 kB/s

7. Mean CPU Utilisation: 16.35% (monthly), 17.2% (yearly)
Eclipse Java EE IDE tool was used for developing and running the model and the VisualVM tool for measuring CPU utilization over time. VisualVM gives a percentage of total CPU utilization (0 to 100%). Results are shown below:

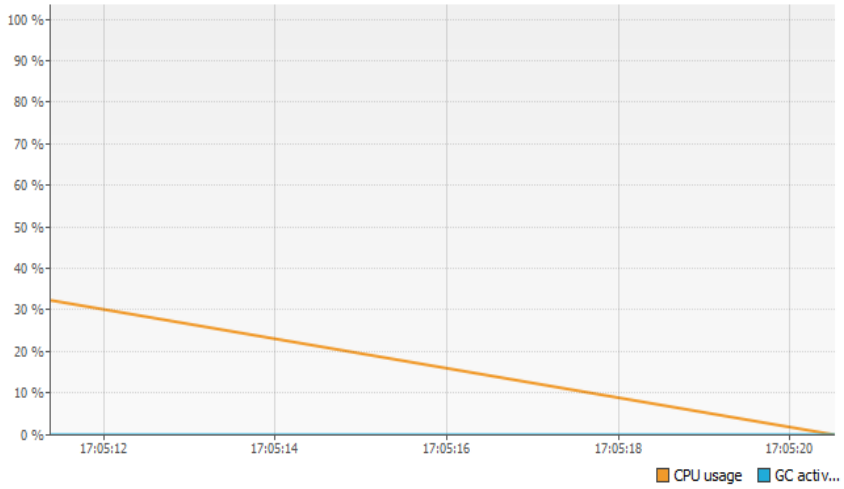

Figure 17. CPU Utilization over time for the PEI model during monthly execution

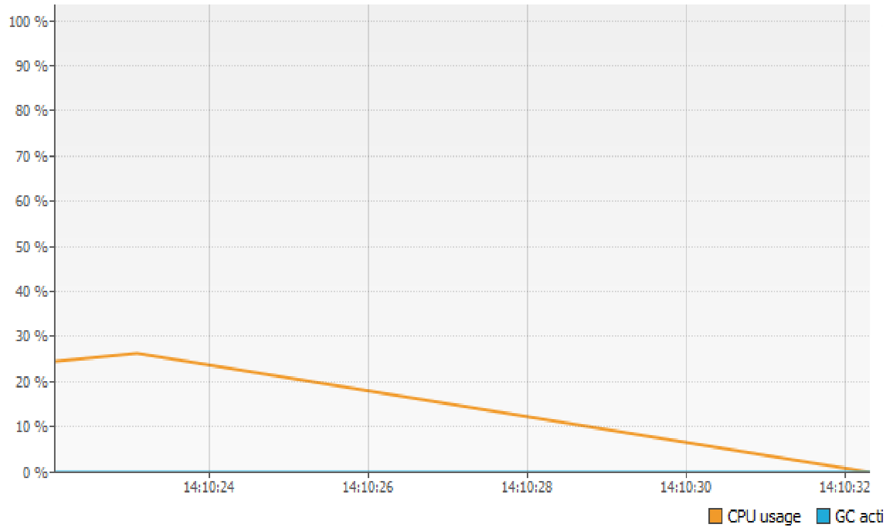

Figure 18. CPU Utilization over time for the PEI model during yearly execution

8. Mean memory usage: 23 MB (monthly), 35.4 MB (yearly)
Eclipse Java EE IDE tool was used for developing and running the model and the VisualVM tool for monitoring the memory Heap usage. This monitoring gives as a result two graphs: the orange one indicates the heap memory size of the Java Virtual Machine (JVM), while the blue one indicates the heap memory usage of the Java application. Results are shown below:

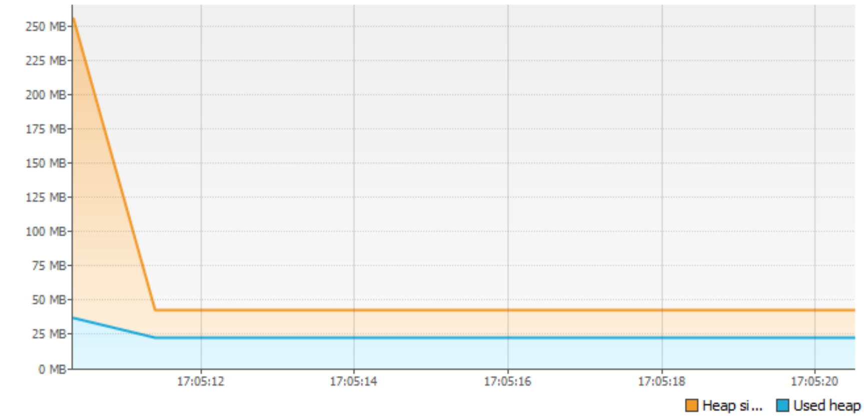

Figure 19. Memory Utilization over time for the PEI model during monthly execution

There is a memory usage value peak at the application startup, but immediately it decreases and stabilizes at a constant value.

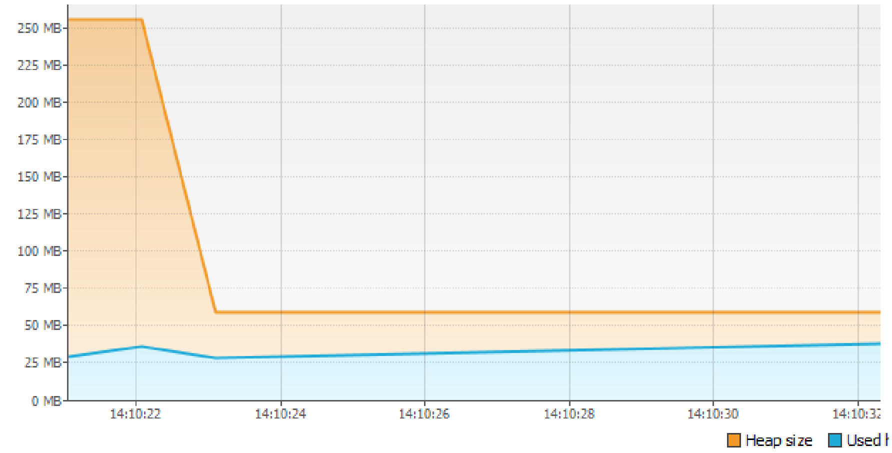

Figure 20. Memory Utilization over time for the PEI model during yearly execution

9. Maximum memory usage: 37.5 MB (monthly), 40.8 MB (yearly)
The same method as above is applied.

10. Maximum CPU usage: 32.7% (monthly), 26.5% (yearly)

11. Simultaneous requests: This model doesn't support simultaneous requests because it runs once on demand or scheduled (e.g., each month, each week, each semester) by the Operational Tools.

12. Percentage of modularity: 20%
The model is composed of 5 components:

- The set of mathematical tools to be applied over the eKPIs (normalization, aggregation, weighting)
- The PEI calculation (a composite indicator, based on a "leaf-to-root" node tree)
- The RR calculation (a composite indicator, based on a "leaf-to-root" node tree)
- The recommendation engine, that dynamically reads the outputs produced and provides some recommendations for improving the index.
- The tool for generating a PDF report.

13. Percentage of reusable assets: 100%.

14. GUI module availability: No

The UI is fully integrated in the Dashboard, where user types and roles are managed in cooperation with Keyrock.

15. WCAG 2.0 Conformance Level: None

As there is no GUI module available, the PIXEL Dashboard is considered the only dashboard to be used by all user types and as such, it is the one that will receive the WCAG evaluation.

Results for all KPIs are summarized in the table below :

Table 18. KPI summary for the PEI software module

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	100% (relevant)
Performance efficiency		
Capacity	3. Maximum number of connected data sources	No limit
	4. Maximum database size	No maximum
Time behaviour	5. Average latency	4s (monthly) 5s (yearly)
	6. Throughput	13.925 kB/s (monthly) 11.14 kB/s (yearly)
Resource utilization	7. Mean CPU Utilisation	16.35% (monthly) 17.2% (yearly)
	8. Mean memory usage	23 MB (monthly) 35.4 MB (yearly)
	9. Maximum memory usage	37.5 MB (monthly) 40.8 MB (yearly)
	10. Maximum processing power used	32.7 % (monthly) 26.5 % (yearly)

Operability		
Ease of Use	GUI module availability	No
Technical Accessibility	WCAG 2.0 Conformance Level	None
Reliability		
Maturity	11. Simultaneous requests	N/A
Maintainability		
Modularity	12. Percentage of modularity	20%
Reusability	13. Percentage of reusable assets	100%

3.10. PIXEL Data Acquisition

3.10.1. Assessment scenario

KPIs assigned to the assessment (PIXEL Data Acquisition assessment) have been described in D8.1, while the tools and methods for their collection have been defined in D6.3. In this section this methodology is further elaborated, and the assessment scenario is described in detail.

In order to assess the performance in the port area, measurements will be performed with a predefined set of realistic input data relevant to port operations. In the beginning, all measurements will be performed under laboratory conditions, and on the infrastructure, which will be defined in WP7 (cloud environment versus on-premises installation and other parameters).

Custom modules: reliability and **portability** are going to be measured using custom modules.

Reliability, portability and few other KPIs depend on the deployment of the modules in an operational scenario in order to measure them, as they are mostly statics related to an operational environment.

In the previous deliverable (D8.2) a set of KPIs were identified and the Data Acquisition Layer was evaluated in an early stage. Some aspects were possible to evaluate whereas others were delayed or postponed for this second phase. The summary table is provided below and updated from D8.2.

Table 19. KPI for data acquisition layer KPI Data Collection and Results

KPI	Measurement method	Reporting
Functional suitability		
Straightforward task accomplishment	Expert judgement	D8.2, D8.3

The portion of completed requirements	Expert judgement	D8.2, D8.3
Performance efficiency		
Maximum number of connected data sources	JMeter	D8.3
Maximum database size	(JMeter)	D8.3
Average latency	JMeter	D8.3
Throughput	JMeter	D8.3
Mean CPU Utilisation	JMeter	D8.3
Mean memory usage	JMeter	D8.3
Maximum memory usage	JMeter	D8.3
Maximum processing power used	JMeter	D8.3
Compatibility		
% of APIs coverage	Expert judgement	D8.3
Ability to acquire data from different data formats	Expert judgement	D8.3
Ability to support different IoT platforms	Expert judgement	D8.3
Ability to export different data formats	Expert judgement	D8.3
Reliability		
Simultaneous requests	JMeter	D8.3
% Monthly availability	Custom module, Phase 2 based on Orion API	D8.3
Success rate	Custom module, Phase 2 based on Orion API	D8.3
Maintainability		
% of modularity	Expert judgement	D8.2
% of reusable assets	Expert judgement	D8.2
% of update	Expert judgement, Phase 2	D8.3
Level of analysability	Expert judgement	D8.2
Portability		

Mean number of errors per hardware or OS change/ upgrade	Custom module, Phase 2	D8.3
Mean number of errors per software change/ update	Custom module, Phase 2	D8.3
Mean number of errors per software install	Custom module, Phase 2	D8.3
Mean number of errors per software uninstall	Custom module, Phase 2	D8.3

3.10.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes.
A process to add new data sources will be analysed to verify that the process does not include unnecessary steps.

Connecting a new Data Source to the Data Acquisition Layer is a manual operation. Its complexity depends on the data source itself and its exposition mechanism. An NGSI agent has to be developed in order to connect the Data Source to the Data Acquisition Layer in order to access the data and convert them to the right format through the chosen security protocol. A generic framework is provided in order to develop those agents quickly. Once developed, the agent is packaged using Docker and deployed in the PIXEL infrastructure. When the agent is deployed, it starts to collect the data and they are immediately available for PIXEL Information Hub. D8.3: The DAL Orchestrator has been developed to facilitate the deployment of new NGSI Agents using an API. It also manages the creation of the different objects used by Information Hub to collect the data from Orion.

2. The portion of completed requirements: 93%
"Should have" and "Must have" requirements from deliverable D3.2 will be taken as input in order to extract all requirements specifically targeting T6.2.

Table 20. lists all PIXEL requirements related to the Data Acquisition Layer that have the priority set to "Should have" or "Must have". It also lists other PIXEL software modules related to the requirements and the status of development in the DAL. The status does not assess the fulfilment of the requirement in other modules.

Requirement	Addressed in additional modules	Implemented in DAL
Common functional requirements		
Homogenize Data [61] *Status: This is the purpose of the NGSI agents. They import the data and transform them using a common Data Model before pushing them to IH.*	IH	yes
Support for manually provided data [86]	IH	yes

Status: NGSI Agent provides several ways to import data through the DAL, one of them is pushing CSV or Json files. But it is also possible to handle HTML forms requests.		
Port of Bordeaux – Energy Management Use Case		
Support electricity consumption sensors [9] *Status: The NGSI agents is developed and deployed*	IH	yes
Monitor expected port calls [11] *Status: The NGSI agents is developed and deployed*	IH	yes
Collect sensor data through Port Community System (VIGIEsip) [12] *Status: The NGSI agents is developed and deployed*	IH	yes
Support Air Quality Sensors [14] *Status: The NGSI agents is developed and deployed*	IH	yes
Support wind speed sensors [16] *Status: The NGSI agents is developed and deployed*	IH	yes
Support weather sensor/service [17] *Status: The NGSI agents is developed and deployed*	IH	yes
Support old sensors (gauge stations network) [18] *Status: The NGSI agents is developed and deployed*	Not defined	yes
Monitoring l'Ostrea dredge environmental impact [20] *Status: No data API identified yet*	IH, PEI	no
Monitor energy consumption of the port authority [22] *Status: DAL provide Python Framework to develop NGSI Agent*	PA	yes
Port of Monfalcone – SDAG – Intermodal Transport Use Case		
Integration with the SILI Information System [23] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Integration with the PMIS Information System [24] *Status: DAL provides Python Framework to develop NGSI Agent.*	IH	yes
Integration with ASPM video monitoring system [25] *Status: DAL provides Python Framework to develop NGSI Agent.*	IH	yes
Integration with the SDAG Access Control System [27] *Status: DAL provides Python Framework to develop NGSI Agent.*	IH	yes
Integration with data provided by sensors, cameras and feeds by third parties [28] *Status: DAL provides Python Framework to develop NGSI Agent.*	IH	yes

Port of Thessaloniki – Port City Integration Use Case		
Support wind and weather sensors [47] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support air quality sensors [48] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support water quality sensors and data [49] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support noise sensors and data [50] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support real-time fuel consumption sensors [51] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support real-time gate surveillance sensors [52] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support wind and weather data provided by third party [53] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support air quality data provided by third party [54] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support traffic data provided by third party [55] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Port of Pireaus – Port City Integration Use Case		
Support air quality sensors [73] *Status: DAL provides Python Framework to develop NGSI Agent*	Not defined yet	Yes*
Support water quality data [75] *Status: DAL provides Python Framework to develop NGSI Agent*	Not defined yet	yes*
Integration with the PMIS SPARC N4 [76] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes
Support noise sensors and data [87] *Status: DAL provides Python Framework to develop NGSI Agent*	Not defined yet	yes*
Support pollution and traffic data provided by third party [88] *Status: DAL provides Python Framework to develop NGSI Agent*	IH	yes

Legend:

- yes: common functional requirements that are implemented in the Data Acquisition Layer
- yes*: Data Acquisition Layer provides the mechanism needed to implement the requirement. But the development of the required NGSI Agent has to be done together with WP7.
- no: The functionality is not yet available.

A total of 30 requirements are related to functionality provided by the Data Acquisition Layer. Out of those, 3 are fully available, 5 in progress and for 20 of them DAL provide the mechanism to implements them

- Total requirements: 30
- Fulfilled requirements (functionality available): 28
- Portion of completed requirements: 93%.

3. % of APIs coverage: 100%

The use of the NGSI Framework allows the acquisition of data from different kinds of systems exposing or consuming data through API. An NGSI Agent has been developed to acquire data from all sources selected for the PIXEL project. The PyNGSI Python framework allows to quickly develop new NGSI Agent and the platform is designed to easily deploy those agents.

4. Ability to acquire data from different data formats: 100%

The use of the NGSI Framework allows the acquisition of data from different kinds of formats. An NGSI Agent has been developed to acquire data from all data formats selected for the PIXEL project. The PyNGSI Python framework allows to quickly develop new NGSI Agent and the platform is designed to easily deploy those agents.

5. Ability to support different IoT platforms: 100%

An IoT platform is a data source like others. The use of the NGSI Framework allows the acquisition of data from different kinds of systems exposing or consuming data through API or MQTT mechanism and the PyNGSI framework could be enhanced to support other protocols. All the IoT platforms selected for the PIXEL project have been successfully connected to the DAL.

6. Ability to export different data formats: Not applicable

Even if the NGSI Framework allows exporting data to other systems, only IH has been connected to Orion to receive data. The Data Exportation to external PIXEL system is managed by IH and its Data Extractor.

7. % Monthly availability : 100%

NAGIOS log provides information about all the components availability. A tool has been developed to parse that information. No incident has been detected during the active pilot phase. For example Orion in GPMB report an uptime of : 238 days (31/07/2021)

8. % Success Rate : 99.8%

The analysis of the Success rate is not easy to evaluate. The usage of FIWARE Orion as the main DAL components and the way NGSI Agents push data using its API doesn't allow error requests on the DAL components itself. No NGSI error detected on log Orion log analysis. Only NGSI Agents can receive error requests from the data source. For example VesselCAll NGSI Agent in GPMB reports 676 errors for 346020 records received.

9. % of update: 100%

Will be measured by reporting the level of success in software updates on the data acquisition layer module. It compares successfully completed updates versus all executed updates.

The process to manage updates for the Data Acquisition Layer as for the full platform relies on docker and docker-compose solutions. The advantage of those features guarantee the capability to reproduce an installation or update on all platforms, with the exact same process. All the installation and update files are shared on a GIT repository and all the platform specific parameters are stored in external files: .env and secrets. That way we can ensure that if an update works on the integration platform it will be successfully deployed on all platforms. In the following table a list of updates has been provided for the period of seven months (January - July 2021). A total of 20 updates have been performed in the three months period, all of them have been successful.

Table 21. PIXEL DAL update log

Date	Version	Component	GPMB	ThPA	PPA	ASPM	Integration Platform
10.03.2021	1.1.1	DAL Proxy	Done	Done	Done	Done	Done
28.04.2021	1.1.2	DAL Proxy	Done	Done	Done	Done	Done
27.05.2021	1.3.3	DAL Orchestrator	Done	Done	Done	Done	Done
23.07.2021	1.3.4	DAL Orchestrator	Done	Done	Done	Done	Done
Total updates			**4**	**4**	**4**	**4**	**4**

10. Mean number of errors per hardware or OS change/upgrade: Not relevant
Will be measured by analysing the system & application error logs. This parameter is not relevant for the PIXEL Data Acquisition Layer, as in principle it is not affected by hardware or software updates. (The use of docker prevent of this kind of issues)

11. Mean number of errors per software change/update: 0
Will be measured by analysing the system & application error logs. For the Data Acquisition Layer, this equals the success rate provided in the *% of update* table.

12. Mean number of errors per software install: 0
Will be measured by analysing the system & application error logs. An analysis will be performed by expert judgement instead of using custom-made modules. The operator will note problems during installation of the four pilots and the test platform and provide an estimate of the mean number of errors per software install.

13. Mean number of errors per software uninstall: Not relevant
Will be measured by analysing the system & application error logs. Data Acquisition Layer as the all platform is distributed as a set of Docker images and installed using Docker Compose. Consequently, the uninstallation procedure is very simple. Using the single 'docker-compose down' command Data Acquisition Layer can be removed. Using the '-v' switch the persisted data (state) is removed as well. For the purpose of data collection, a setup with two workstations has been used:

- PIXEL DAL deployment in a workstation with:

    - Docker
    - DAL (FIWARE Orion)
    - JMeter PerfMon
- Testing workstation: a workstation with installed JMeter probing and reporting tools.

The deployment workstation has the following specifications:

- CPU: Intel(R) Core (TM) i7-8559U CPU @ 3.40GHz
- RAM: 16 GB RAM
- SDD: 500 GB
- OS: Ubuntu 20.04 (64-bit)
- The FIWARE foundation provides Performance testing results and scripts for ORION https://github.com/telefonicaid/fiware-orion/tree/master/test/loadTest. We have run several tests to evaluate its usage in the PIXEL context.

We have used 3 scenarios, with clients that simulate data source that push new entities in Orion at the around 5 req/second, with 5, 60 and 100 clients simultaneously and with clients that push entities as quickly as possible with 5 simultaneous clients.

Table 22. PIXEL DAL Load Tests

Test ID	Test setup				Test execution - achieved performance
	clients	req/s/client	Total req/s	Requests (3 min)	Requests (3 min)
1	4	~6,25	25	4500	4462
2	60	~6.25	375	67500	62444
3	100	~6.25	625	112500	72378
4	5	~80	400	75000	75125

Memory CPU and memory utilization is provided in the table below. All values are provided in %.

Table 23. PIXEL DAL Load Test result

Test ID	CPU mean	CPU max	Memory mean	Memory max
1	3.94	5.78	47.71	47.78
2	15.59	18.94	48.59	48.77
3	16.68	28.49	47.62	47.94
4	17.78	18.63	49.13	49.34

14. Maximum number of connected data sources: 60

The maximum number of connected data sources depends on the profile of each data source connected. The current test shows that with this configuration Orion can manage around 400 requests per second. So if we have a data source that homogeneously uses the available capabilities to push 10 requests per day, we can manage up to 3 million of data sources per day. But obviously the repartition is never homogenous, with scenario 2, we can manage 60 simultaneous data sources.

15. Maximum database size: Depends on MongoDB deployment
It depends on the MongoDB deployment.

16. Average latency: 5ms
In normal situations the response time of the Orion Context Broker is around 5ms, but when the load start to become too high the response time increases a lot, but the requests are still managed. With the current configuration the maximum throughput to keep the response time at a normal level is around 400 requests per seconds. Here we increase slowly the number of client up to 100 simultaneous client at 6.25r/s

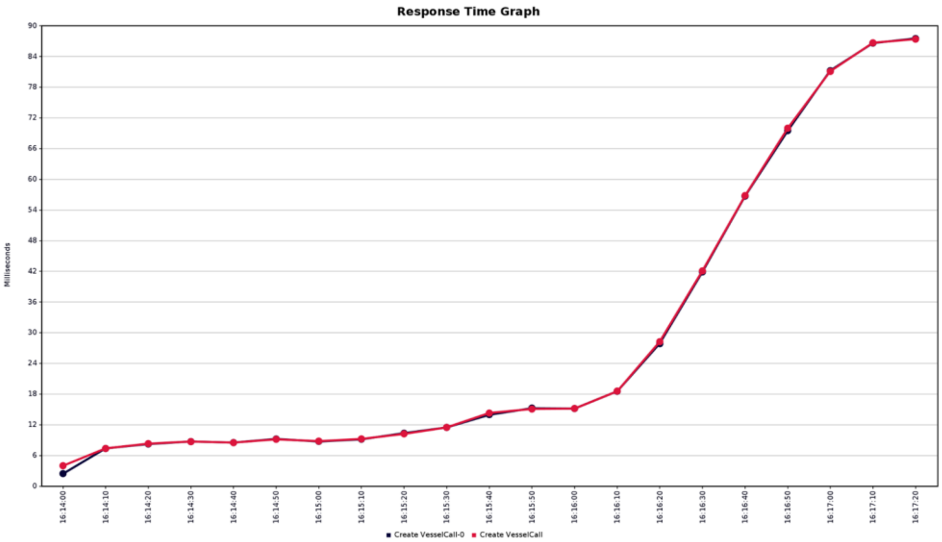

Figure 21. Response time analysis for the DAL

17. Throughput: 400 requests/s
In this configuration we reach the maximum throughput at around 400 requests per second. The throughput is quite the same if we use 5 simultaneous clients pushing at 80 requests per second or 60 clients pushing at 6 requests per second.

18. Mean CPU Utilisation: 12.5%
The CPU usage limitation comes from MongoDB. Our maximum use of MongoDB is for writing, in this situation the implementation of the MongoDB engine can only use one CPU core. As our test server is an 8 core computer, MongoDB can't use more than 12.5% of the total CPU. That is the main limitation for the capabilities of the DAL solution. But even with this limitation the capabilities are enough for PIXEL usage, and it presents the advantage to keep the other core for other PIXEL components. Here we increase slowly the number of clients up to 100 simultaneous client at 6.25r/s.

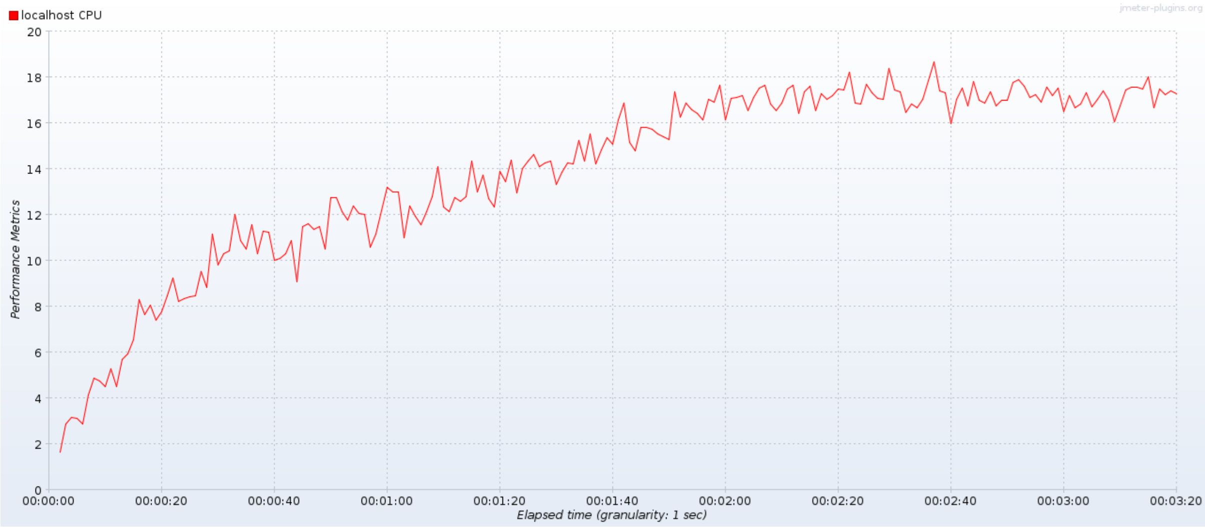

Figure 22. CPU Utilisation analysis for the DAL

19. GUI module availability: No

DAL Acquisition Layer provide API (and swagger) to manage all the feature of the different components, but no GUI has been developed for that purpose.

20. WCAG 2.0 Conformance Level: None

As there is no GUI module available, the PIXEL Dashboard is considered the only dashboard to be used by all user types and as such, it is the one that will receive the WCAG evaluation.

Results for all KPIs are summarized in the table below :

Table 24. KPI summary for the DAL

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1.Straightforward task accomplishment	Yes
Functional completeness	2. The portion of completed requirements	97%
Performance efficiency		
Capacity	3. Maximum number of connected data sources	60
	4. Maximum database size	Not measured - depends on MongoDB deployment.
	5. Average latency	6 ms

Time behaviour	6. Throughput	400 msg/s
Resource utilisation	7. Mean CPU Utilisation	18%
	8. Mean memory usage	48%
	9. Maximum memory usage	50%
	10. Maximum processing power used	50%
Compatibility		
Interoperability	11. % of APIs coverage	100%
	12. Ability to acquire data from different data formats	100%
	13. Ability to support different IoT platforms	100%
	14. Ability to export different data formats	Not applicable
Operability		
Ease of Use	GUI module availability	No
Technical Accessibility	WCAG 2.0 Conformance Level	None
Reliability		
Maturity	15.Simultaneous requests	60
Availability	16. % Monthly availability	100%
	17. % Success Rate	99.8%
Maintainability		
Modifiability	18. % of update	100%
Portability		
Adaptability	19. Mean number of errors per hardware or OS change/upgrade	Not relevant

	20. Mean number of errors per software change/update	0
Installability	21. Mean number of errors per software install	0
	22. Mean number of errors per software uninstall	Not relevant

3.11. PIXEL Information Hub

3.11.1. Assessment scenario

We evaluated most of the KPIs of the Information Hub in D8.2, and decided to not include those that have not changed in D8.3 to not overcharge the evaluation. However, some of the KPIs listed in Table 14 of D8.2, mostly those linked to the deployment of the modules in an operational scenario, were expected to be evaluated for D8.3, once such scenario was available. Those KPIs are listed below, with descriptions taken from D6.3, which described the methodology to be followed. Additionally, in this section we also provide some implementation details about environments and development of custom evaluation modules.

Functional suitability has been evaluated by expert judgement based on available documentation, code and testing. We have developed several custom modules to evaluate *Reliability*, which have been deployed and data collected for the following pilots: GPMB, ThPA. Custom modules collected data for a specific time period (as described in the results section) and KPI have been calculated from collected data. *Maintainability* and *Portability* have been estimated based on the first few months of experience gained during pilot deployments and maintenance.

We evaluated all characteristics based on 2 different methods, expert judgement and custom module:

- Functional suitability: Expert judgement
- Reliability: Custom module
- Maintainability: Expert judgement
- Portability: Custom module/Expert judgement
- Portability: Custom module/Expert judgement

3.11.2. KPI Data Collection and Results (XLAB)

1. Straightforward task accomplishment: Yes
A process to add new data sources and the process to provide data (data extractor) will be analysed to verify that the process does not include unnecessary steps. Boolean response (Yes/No)

Since the publication of D8.2 both functionalities, as described previously, are still available in the IH and have been used in several test and pilot deployments. It has been reconfirmed that for both modules the value of the KPI is YES.

2. The portion of completed requirements: 95%

"Should have" and "Must have" requirements from deliverable D3.2 will be taken as input in order to extract all requirements specifically targeting T6.3.

Table 25. below lists all PIXEL requirements related to the IH that have the priority set to "Should have" or "Must have". It also lists other PIXEL software modules related to the requirements and the status of development in the IH. The status does not assess the fulfilment of the requirement in other modules.

Requirement	Addressed in additional modules	Implemented in IH
Common functional requirements		
Analyse historical data [81] *Status: data collected through DAL can be stored in the IH and extracted through IH or Elasticsearch REST API.*	OT	yes
Support for manually provided data [86] *Status: data collected through DAL can be stored in the IH and extracted through REST APIs. IH is agnostic in relation to the collection method.*	DAL	yes
Port of Bordeaux – Energy Management Use Case		
Access to traffic data [10] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Collect sensor data through Port Community System (VIGIEsip) [12] *Status: FAL Forms data collected.*	DAL	**yes**
Support Air Quality Sensors [14] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support wind speed sensors [16] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support weather sensor/service [17] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Monitoring l'Ostrea dredge environmental impact [20] *Status: Details of the integration of the PEI module in the overall information architecture has not yet been specified.*	DAL, PEI	no (not finally needed in the pilot)
Expose data to VIGIEsip system [82] *Status: all data in the IH is available either through IH or Elasticsearch REST API.*		(yes)
Port of Monfalcone – SDAG – Intermodal Transport Use Case		

Integration with the SILI Information System [23] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Integration with the PMIS Information System [24] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Integration with ASPM video monitoring system [25] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Integration with the SDAG Access Control System [27] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Port of Thessaloniki – Port City Integration Use Case		
Support noise sensors and data [50] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support real-time fuel consumption sensors [51] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support real-time gate surveillance sensors [52] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support wind and weather data provided by third party [53] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support air quality data provided by third party [54] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support traffic data provided by third party [55] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Port of Piraeus – Port City Integration Use Case		
Integration with the PMIS SPARC N4 [76] *Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.*	DAL	**yes**
Support pollution and traffic data provided by third party [88]	DAL	**yes**

Status: DAL-IH connection has been implemented. All data collected through DAL is available in the IH.		

The above "Implemented in IH" columns values correspond to the following:

- yes: Requirements that have been already implemented and reported in D8.2.
- (yes): requirements that are related to data acquisition for specific data sources in different use cases. The functionality is, in principle, available in the IH as it relates to a common functional requirement for data availability.
- **yes**. The functionality has been deployed and tested as part of WP7 activities.
- no: The functionality is not going to be implemented.

A total of 21 requirements are related to functionality provided by the IH.

Total requirements: 21

- Fulfilled requirements (functionality available): 20
- Portion of completed requirements: 95%.

3. % of update: 100%
Will be measured by reporting the level of success in software updates on the information hub module. It compares successfully completed updates versus all executed updates.

- The procedure to update Information Hub is as follows:
  - Stop the Docker container corresponding to the component which is going to be updated and remove it.
  - Pull the new Docker image.
  - Start the new image.

  Bash scripts are available for stopping and removing Information Hub containers as well as for pulling the latest version of all Information Hub images. New container(s) can be started using the usual PIXEL platform startup script.

In the following table a list of updates has been provided for the period of three months (January - March 2021). A total of 41 updates has been performed in the three months period, all of them have been successful. The percent of update rate for the observed period is 100%.

Table 26. PIXEL IH update log

Date	Version	Component	GPMB	ThPA	PPA	ASPM	Integration Platform
17.01.2021	1.3.0	Orion Data Collector	Done	/	/	/	Done
19.01.2021	1.4.0	Orion Data Collector	Done	/	/	/	Done
04.02.2021	1.2.0	Data Writer	Done	Done	Done	Done	Done

04.02.2021	1.2.0	Data Extractor	Done	Done	Done	Done	Done
04.02.2021	1.2.0	AIS Data Collector	/	/	/	/	Done
04.02.2021	1.5.0	Orion Data Collector	Done	Done	Done	Done	Done
16.02.2021	1.5.1	Orion Data Collector	Done	Done	Done	Done	Done
03.03.2021	1.2.1	Data Writer	Done	Done	Done	Done	Done
09.03.2021	1.2.1	AIS Data Collector	/	/	/	/	Done
11.03.2021	1.5.2	Orion Data Collector	Done	Done	Done	Done	Done
24.03.2021	1.2.2	Data Writer	Done	Done	Done	Done	Done
Total updates			**9**	**7**	**7**	**7**	**11**

4. % Monthly availability: 99.84 % (Data Writer), 99.84 % (Data Extractor), 100 % (Orion Data Collector). For health status, an availability probe must be defined with minimal impact on performance. A periodic process will check regularly (e.g. every hour) if the IH (e.g. sending the test input and getting an expected successful response). Statistics will be collected and be available per month. A custom health probe has been developed to periodically collect and store availability data from several IH components. The corresponding Docker image can be found in PIXEL's registry: http://docker.pixel-ports.eu/ih-health-probe:v0.2.1. The probe has been implemented as a script that calls the Information Hub Extractor REST API and stores success statistics in a CSV file. The probe runs on a schedule every hour. The probe retrieves the list of data sources, checks the status of the Orion Data Collector, Data Writer and Data Extractor. Reported values have been computed for one of the pilots (THPA).

5. Success rate: 99.46 %.
For each access to the PIXEL information hub, the success or failure will be recorded, serving as a statistical indicator. The success rate will be measured through the following log files provided by the Information HUB:

- Elasticsearch Proxy as part of Information Hub logs (docker logs core_elasticsearch-proxy_1). These are HTTP access logs and HTTP result codes are used to measure success (HTTP codes 200 to 299).
- Requests to Data Extractor and Elasticsearch Proxy APIs are proxied through the 'internal proxy' nginx (docker logs internal-proxy), access logs are available in docker logs. These are HTTP access logs and HTTP result codes are used to measure success (HTTP codes 200 to 299). However, it may be possible to call the docker container directly (depending on which docker networks models have access to) and bypass the internal proxy. So the access logs may not be complete.

- "Orion data collector" that gets data from DAL logs all actions to a log file. A tool has been implemented to parse those logs and analyse the success rate of the following actions: Data source registration request and Storing data to Information Hub.

This value was computed as the ratio of successful requests (i.e. with HTTP code 200 to 299) to the total number of requests done through the NGINX internal proxy to the Extractor and Elasticsearch Proxy IH components for a time period over three months (from 13 January to 28 April 2021) in one of the pilots (GPMB). The number of successful requests was 46783, while only 252 requests failed. Additionally, the success rate of the Orion Data Collector was evaluated for a period of three days. During this period, 8117 storage requests were recorded, all of which were successful.

6. Mean number of errors per hardware or OS change/ upgrade: Not relevant.
Will be measured by analysing the system & application error logs. This parameter is not relevant for the PIXEL Information Hub, as in principle it is not affected by hardware or software updates.

7. Mean number of errors per software change/ update: Same as % of update, 100%.
Will be measured by analysing the system & application error logs. For the Information hub, this equals the success rate provided in the *% of update* table.

8. Mean number of errors per software install: <1
Will be measured by analysing the system & application error logs. An analysis will be performed by expert judgement instead of using custom-made modules. The operator will note problems during installation of the four pilots and the test platform and provide an estimate of the mean number of errors per software install. The PIXEL Information Hub has been installed at the Integration platform and then for all four pilots (GPMB, ThPA, PPA, ASPM). We logged two issues during the installation at the Integration Platform (test environment) and no further issues during deployments for pilots. The mean number of errors per software install is less than 1.

9. Mean number of errors per software uninstall: 0
Will be measured by analysing the system & application error logs. Information Hub is distributed as a set of Docker images and installed using Docker Compose. Consequently, the uninstallation procedure is very simple. Using the single 'docker-compose down' command Information Hub can be removed. Using the '-v' switch the persisted data (state) is removed as well. The uninstallation procedure was tested a few times on the integration platform and it worked without any problems.

10. GUI module availability: No

Information Hub GUI is intended to be used only by system administrators. It is a backend service so there is no need to be available to others.

11. WCAG 2.0 Conformance Level: None

As there is no GUI module available, the PIXEL Dashboard is considered the only dashboard to be used by all user types and as such, it is the one that will receive the WCAG evaluation.

Results for all KPIs are summarized in the table below :

Table 27. KPI summary for the Information Hub

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	1. Straightforward task accomplishment	Yes
Functional completeness	2. Portion of completed requirements	95%
Reliability		
Availability	4. % Monthly availability	99.84 % (Data Writer), 99.84 % (Data Extractor), 100 % (Orion Data Collector).
	5. Success rate	99.46 %.
Operability		
Ease of Use	GUI module availability	No
Technical Accessibility	WCAG 2.0 Conformance Level	None
Maintainability		
Modifiability	3. % of update	100%
Portability		
Adaptability	6. Mean number of errors per hardware or OS change/ upgrade	Not relevant.
	7. Mean number of errors per software change/ update	Same as % of update, 100%.
Installability	8. Mean number of errors per software install	<1
	9. Mean number of errors per software uninstall	0

3.12. PIXEL Operational Tools

3.12.1. Assessment scenario

The PIXEL Operational Tools (OT) is an essential component of the PIXEL platform as it allows the management and execution of models and predictive algorithms within the PIXEL architecture. It uses a REST API so it allows automatic tests through development tools (e.g. Postman), it also includes a Swagger API for manual testing, and the PIXEL Dashboard is also able to interact with them to offer a final end user interface.

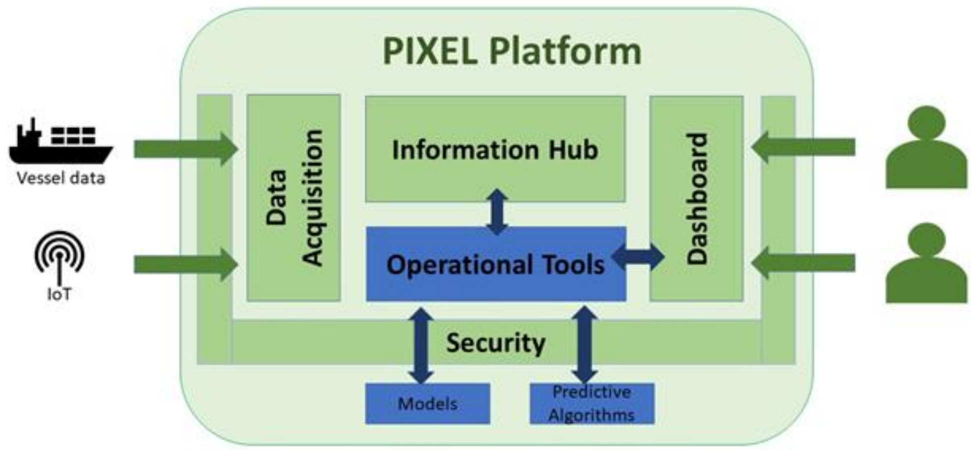

Figure 23. Operational Tools in the PIXEL platform

The Operational Tools is composed of a set of two Docker containers:

- **MongoDB database** to store specific information of the application.
- **Tomcat server** with a Java application for the **API REST** and **backend logic.**

The Operational Tools allows performing various tasks:

- **Publishing models and predictive algorithms**: this task is accomplished by entering the endpoint where the Docker image is available.
- **Executing models and predictive algorithms**: this task is accomplished by providing a JSON file with input and output configuration.
- **Scheduling models and predictive algorithms**: this task allows scheduling the execution of models and predictive algorithms (E.g. every hour/day/week/month).
- **Other**: OT also allows managing KPIs and includes utility functionalities, but for the purpose of this deliverable they are not relevant.

In the previous deliverable (D8.2) a set of KPIs were identified and the OT were evaluated in an early stage. Some aspects were possible to evaluate whereas others were delayed or postponed for this second phase. The summary table is provided below and updated from D8.2.

In the following section, JMeter and PerfMon are used as the custom method for measurement where an expert judgement cannot be used. The configuration used is the following:

- Server used: Intel Core i5-4430 @ 3 GHz (CPU), Windows 10 Pro (SO), 24 GB (RAM), 500 GB SSD-HD
- OT-API-Write Operation: **UPDATE api/models/update (ping, traffic-pa)**
- OT-API-Read Operation: **GET /api/models/list**
- OT-Model publication: **PUT /api/models/create**
- OT-Model Execution: **PUT /api/instances/create**

Models and PAs are now launched as independent Dockers; therefore, the execution does not reflect a big impact an is similar to a publication process

- Baseline memory**: 50%** (after start-up)
- Baseline CPU**: 40%** (after start-up)

Table 28. KPIs for Operational Tools

KPI	Measurement method
Functional suitability	
Straightforward task accomplishment	Expert judgement
The portion of completed requirements	Expert judgement
Performance efficiency	
Average latency	JMeter
Mean CPU Utilisation	JMeter, PerfMon
Mean memory usage	JMeter, PerfMon
Maximum memory usage	JMeter, PerfMon
Maximum processing power used	JMeter, PerfMon
Reliability	
Simultaneous requests	JMeter
% Monthly availability	Custom (new for D8.3)
Success rate	Custom (new for D8.3)
Maintainability	
% of modularity	Expert judgement
% of reusable assets	Expert judgement

% of update	Custom (new for D8.3)
Level of analysability	Expert judgement
Portability	
Mean number of errors per hardware or OS change/ upgrade	Custom (new for D8.3)
Mean number of errors per software change/ update	Custom (new for D8.3)
Mean number of errors per software install	Custom (new for D8.3)
Mean number of errors per software uninstall	Custom (new for D8.3)

3.12.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes
A process to add a new model or predictive algorithm will be analysed to verify that the process does not include unnecessary steps.

Boolean response (Yes/No)

A process to run/schedule a new model or predictive algorithm will be analysed to verify that the process does not include unnecessary steps.

Boolean response (Yes/No)

[Note: the configuration of events was merged within the PIXEL Dashboard and will not evaluated as part of the OT, as initially planned in D8.2]

Process to add a new model: Yes (100%)
The process only requires passing a JSON file with all necessary data. Basic UI available, but recommended to use the PIXEL Dashboard as UI.

Process to run a new model: Yes (100%)
The process only requires passing a JSON file with all necessary data. Basic UI available, but recommended to use the PIXEL Dashboard as UI.

Process to schedule a new model: Yes (100%)
The process only requires passing a JSON file with all necessary data (similar to execution, but adding schedule/timing info). Basic UI available, but recommended to use the PIXEL Dashboard UI.

Process to add, run and schedule a predictive algorithm: Yes (100%)
Predictive algorithms are treated in the same way as models, only a field in the JSON file changes to categorize both.

2. The portion of completed requirements: 100%

Deliverable D3.2 will be taken as input in order to extract all requirements specifically targeting T6.4. So far, specific requirements relate to:

- Interaction with models [41]
- Interaction with Catalogue [62]

[Note: the configuration of events and anomalies was merged within the PIXEL Dashboard and will not evaluated as part of the OT, as initially planned in D8.2 for requirements [44], [45] and [63]]

Following D3.2 requirements for the Operational Tools:

Interaction with models [41]: Yes (100%)
The OT Engine is able to publish and discover available models and predictive algorithms in the PIXEL platform, execute the involved models and obtain a valid response. Integration with the IH is directly done through the corresponding (logical) OT subcomponent embedded in the models/PAs supporting the connector.

Catalogue of models [62]: Yes (100%)
The OT framework allows converting WP4 models and predictive algorithms into (Docker) services pluggable into the PIXEL platform. Once published, the models/PAs can be listed through the OT

3. Average latency: 39ms (OT-API-Write Operation) - 48ms (OT-API-Read Operation) - 48s (OT-Model publication) With JMeter, requests to multiple services (encapsulating models and PAs) will be launched and average response time will be measured. We will differentiate 3 operational ranges:

- Low: least estimation of models and PAs
- Medium: average estimation of models and PAs
- High: worst-case estimation of models and PAs

OT-API-Write Operation: 39 ms
OT-API-Read Operation: 48 ms
OT-Model publication: 45 s (ping-count), 48 s (pa-traffic)

4. Mean CPU Utilisation: 84% **(**OT-API-Write Operation) **-** 80% (OT-API-Read Operation) **-** 85% (OT-Model deployment ping-count) - 91% (OT-Model deployment pa-traffic) The same approach as for average latency is valid. In order to get the mean CPU usage, PerfMon will be used for the same JMeter tests.

5. Mean memory usage: 52% (OT-API-Write Operation) - 51% (OT-API-Read Operation) - 54% (OT-Model deployment ping-count) - 56% (OT-Model deployment pa-traffic) The same approach as for average latency is valid. In order to get the mean CPU usage, PerfMon will be used for the same JMeter tests.

6. Maximum memory usage 52% (OT-API-Write Operation) - 52% (OT-API-Read Operation) - 56% (OT-Model deployment ping-count) - 57% (OT-Model deployment pa-traffic) The same approach as for average latency is valid. In order to get the mean CPU usage, PerfMon will be used for the same JMeter tests.

7. Maximum processing power used: 90% (OT-API-Write Operation) - 85% (OT-API-Read Operation) - 92% (OT-Model deployment ping-count) - 86% (OT-Model deployment pa-traffic) The same approach as for average latency is valid. In order to get the mean CPU usage, PerfMon will be used for the same JMeter tests.

8. Simultaneous requests: 21 to 38 (OT-API-Write Operation) - 23 to 36 (OT-API-Read Operation) - 9 to 13 (OT-Model publication ping-count) The same approach as for average latency is valid and JMeter will be used. Here JMeter probes will be defined to increase the number of concurrent requests progressively until the load arrives at a certain threshold.

OT-API-Write Operation: up to 21 with no effect on CPU/memory, not recommended more than 38
OT-API-Read Operation: up to 23 with no effect on CPU/memory, not recommended more than 36
OT-Model publication: up to 9 with no effect on CPU/memory, not recommended more than 13 (tested with ping-count)

9. % Monthly availability: >99%
For health status, an availability probe must be defined per each model and PA with minimal impact on performance. A test input might be provided by model/PA. A periodic process will check regularly (e.g. every hour) if a model/PA is available (e.g. sending the test input and getting an expected successful response). Statistics will be collected and be available per month. If there is unavailability from a service (model), it will try to recover automatically, otherwise, a notification (to the administrator) will be sent. Nagios (part of PIXEL internal monitoring service) was extended to log and persist host and service activity, thus providing a common way to all core architecture components to check and test availability. Partial unavailability was only detected during the upgrade of the platform, as some Docker instances need to be stopped, rebuilt and restarted. For the OT, there was no need to rebuild, and the only unavailable time was the one needed to download the new image (around 1 minute) and restart the Docker instance (less than a minute). Furthermore, the download time (docker pull) could even be performed offline. In terms of infrastructure, the 2 VMs dedicated to the PIXEL platform have not experienced any unavailability that may affect the service.

10. Success rate: 100%
For each execution of the service (model), the success or failure will be stored, serving as a statistics indicator. The latest version of the OT only needs to launch the Docker-compose file with the proper parameters. Only if the Docker container is not executed there is an error from the perspective of the OT. This situation is really unlikely to happen, as during the publication of models/PAs several checks are performed to avoid such types of failure. Any other errors due to problems during the execution are controlled by each model/PA and are logged in the Information Hub through a connector. This allows to check the success rate for every model/PA

11. % of modularity: 80%
Will be measured by reporting all the independent components that are part of the operational tools module and comparing them to the number of all components in the data acquisition module. Individual operation means that a component can offer a complete function with meaningful information in the context of PIXEL.

12. % of reusable assets: 90%
Will be measured by reporting all the reusable components that are part of the operational tools module and comparing them to the number of all components in the data acquisition module. A reusable component is considered any that can be applied in a different context of PIXEL with no modifications of the source code.

- OT-API. Reusable 100% (ad hoc component, structured in Java packages, but needs adaptation. As Docker container standalone, though recommended within a PIXEL platform).
- OT-UI. Reusable 60% (ad hoc component, based on VUE templates, needs adaptation).

13. % of updates: Custom (new for D8.3)
Will be measured by reporting the level of success in software updates on the operational tools module. It compares successfully completed updates versus all executed updates.

14. Level of analysability: 100%
Will be measured by reporting the ratio between the numbers of items inside the operational tools for which logging is implemented compared to the number of items for which the specifications require logging.

- OT-API. Logging supported (Log4j to file and console, thus able to use Docker logs).
- OT-UI. Logging not supported (not needed).

15. Mean number of errors per hardware or OS change/ upgrade: Not relevant
Will be measured by analysing the system & application error logs. Not relevant. The OT have been completely Dockerized for the PIXEL platform and therefore any change in hardware or OS change does not affect the Docker instances.

16. Mean number of errors per software change/ update: Not relevant
Will be measured by analysing the system & application error logs. Not relevant. The OT have been completely Dockerized for the PIXEL platform and therefore any change in hardware or OS change does not affect the Docker instances. The only moment when there could be an issue is if the Docker daemon and related libraries are changed; but this has not been the case during the pilots or, at least, no errors have been detected.

17. Mean number of errors per software install: 0
Will be measured by analysing the system & application error logs. No errors were detected during the software installation of the OT. It basically consists of a Docker-compose file with two Docker instances and a few short configuration files.

18. Mean number of errors per software uninstall: 0
Will be measured by analysing the system & application error logs. No errors were detected during the software uninstall, even if this is not the typical procedure in the PIXEL platform.

19. Dashboard availability: Yes
Besides an auto generated webpage (Swagger) to explore the API requests there is a basic VUE UI for standalone use; however, it is recommended to use the PIXEL Dashboard which fully exploits the OT API.

20. GUI module availability: No
There is no GUI intended for end user usage. There is only a basic UI for standalone development, but it was discontinued in favour of the Dashboard
OT is considered as a backend service. User types/roles are managed from the front-end (Dashboard) in cooperation with Keyrock, as part of the overall PIXEL architecture.

21. WCAG 2.0 Conformance Level: None
As there is no GUI module available, the PIXEL Dashboard is considered the only dashboard to be used by all user types and as such, it is the one that will receive the WCAG evaluation.

Results for all KPIs are summarized in the table below :

Table 29. KPI summary for the Operational Tools

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	Straightforward task accomplishment	Yes
Functional completeness	Portion of completed requirements	100%
Performance efficiency		
Time behaviour	Average latency	39ms (OT-API-Write Operation) 48ms (OT-API-Read Operation) 48s (OT-Model publication)
Resource utilization	Mean CPU Utilisation	84% (OT-API-Write Operation) 80% (OT-API-Read Operation) 85% (OT-Model deployment ping-count) - 91% (OT-Model deployment pa-traffic)
	Mean memory usage	52% (OT-API-Write Operation) 51% (OT-API-Read Operation) 54% (OT-Model deployment ping-count) 56% (OT-Model deployment pa-traffic)
	Maximum memory usage	52% (OT-API-Write Operation) 52% (OT-API-Read Operation) 56% (OT-Model deployment ping-count) 57% (OT-Model deployment pa-traffic)
	Maximum processing power used	90% (OT-API-Write Operation) 85% (OT-API-Read Operation) 92% (OT-Model deployment ping-count) 86% (OT-Model deployment pa-traffic)
Operability		
Ease of use	Dashboard availability	Yes
	GUI module availability	No

Technical Accessibility	WCAG 2.0 Conformance Level	None
Reliability		
Maturity	Simultaneous requests	21 to 38 (OT-API-Write Operation) 23 to 36 (OT-API-Read Operation) 9 to 13 (OT-Model publication ping-count)
Availability	% Monthly availability	>99%
	Success rate	100%
Maintainability		
Modularity	Percentage of modularity	80%
Reusability	Percentage of reusable assets	90%
Modifiability	% of update	Custom (new for D8.3)
Analysability	Level of analysability	100%
Portability		
Adaptability	Mean number of errors per hardware or OS change/upgrade	Not relevant
	Mean number of errors per software change/update	Not relevant
Installability	Mean number of errors per software install	0
	Mean number of errors per software uninstall	0

3.13. PIXEL Integrated Dashboard and Notifications

3.13.1. Assessment scenario

The tool for Integrated Dashboards and Notifications (hereinafter in this section, *dashboards*) represent the main entry point by the user and, at the same time, the last link in the data processing and analysis performed by the platform. This means that it is also the last element affected by any kind of change upwards in the waterfall, and, at the same time, each inconsistency or lack of usability is detected on this module immediately by the end-user.

Figure 24. PIXEL Dashboard in the Workflow

Thus, the assessment of the dashboards has to consider the elements upstream and, if it is the case, the version including changes; while also being specific for each end-user (in this project, the four pilot ports). To solve this complexity, the integration process included 1 or 2 development servers (depending on the project phase), as well as the 4 pilot deployment. To effectively introduce all the changes in an organized way, only one dashboard deployment to production (pilots) was done during the last period of integration. The process of validation of the dashboards has been performed following the criteria described in D8.2:

- Expert judgement: this has been done making a judgment based on skill, expertise, or specialized knowledge in a particular area. For the case of dashboards, the expert judgement has been performed by three different profiles: senior software architect, DevOps manager and usability expert.
- Automated testing: by using Apache Jmeter.
- Prospective assessment: by using PerfMon (in the case where it is indicated).

The expert judgement has been done at different phases of the integration, including a last assessment at the end of WP7 works. For the assessment performed with specialized tools, it has been done in planned time windows during the pilot execution.

3.13.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes
We measure multiple tasks:

- if there exists a process to add / configure widgets. The process does not include unnecessary steps. Here, visualizations are created using wizards tailored for specific models or generic for time series available on the IH.
- If there exists a process to add / configure widgets. The process does not include unnecessary steps.
- If there exists a process to send / receive notifications. The process does not include unnecessary steps. Notifications are composed via a specific wizard.
- If there exists a process to create new alerts. The process does not include unnecessary steps. Alerts are created in the same wizard as the notifications are configured.

2. % of completed requirements: 100%
Measured by reporting all the independent components that are part of the dashboard and notifications module. Individual operation means that a component can offer a complete function with meaningful information in the context of PIXEL.

Table 30. List of completed requirements for the PIXEL Integrated Dashboard and Notifications

Requirement	Addressed in additional modules	Implemented in Dashboard and completely fulfilled
Common non-functional requirements		

Import historical data [36]	Platform-level requirement	Yes. The dashboard includes a mechanism to upload historical data in different file formats.
Interaction with models [41]	Platform-level requirement	Yes. The dashboards tool includes user interfaces to manage model interaction.
Anomaly and event list [44]	Platform-level requirement	This requirement has been fulfilled by the development of the alerts engine and UI, as it allows to program anomaly detection based on thresholds and lists the triggered events on a list.
Anomaly and event detection [45]	Platform-level requirement	This requirement has been fulfilled by the development of the alerts engine and UI, as it allows to program anomaly detection based on thresholds and lists the triggered events on a list.
Homogenize Data [61]	Platform-level requirement	This is addressed in DAL
Catalogue of models [62]	Platform-level requirement	Yes. Dashboards include a UI to browse the available models.
Detection of anomalies [63]		Yes. This requirement has been fulfilled by the development of the alerts engine and UI
Feedback [64]	Platform-level requirement	Yes. Currently, the feedback is collected via JIRA or direct email to the platform development team, however, there is no in-app support for feedback. This is a Should-have requirement.
Centralized user administration system [65]	Platform-level requirement	Yes. A user management system is available on the UI of the platform.
Configurable Dashboard [66]	OT	Yes. One of the main interaction mechanisms of the UI are the wizards that ease the creation of model executions and visualizations for the results. All of them are grouped in dashboards that can be modified with drag-and-drop interaction.

UI Notification System [67]	Platform-level requirement	Yes. The platform includes a notifications area. Other notification channels (such as email) are available on the platform.
Port Operational KPI list [70]	OT	Yes (alerts)
Operational Interface [71]	OT	Yes. The dashboards include a graphic user interface to manage the operational tools.
Discovery service for data [104]	Platform-level requirements	Yes. The dashboards platform UI includes a tool for discovering the data sources attached to the PIXEL instance.
Visualization of data [105]	OT	Yes. Data can be visualized by using the dashboards tool.
Use-cases functional requirements		
Truck re-routing alerting system for operators [31]	OT	Yes (see 4.2.1.1)
Truck re-routing alerting system for final users [32]	OT	No. (Could)
Truck re-routing booking system [33]	OT	No. (Could)
Visualize the traffic status [106]	OT	Yes. The visualizations for the traffic status are available on the platform.
Visualize the pollution [107]	OT	Yes. The visualization for the pollution is available on the platform.
PEI Notification [94]		Outdated. The notifications feature is ready but there is no PEI functionality calling it.
PEI Dashboard [95]		Yes. Dashboards tool is available and functional for PEI.
PEI Dashboard – Time series [96]		Yes

3. % of modularity: 40%

Measured by reporting all the independent components that are part of the dashboard and notifications module. Individual operation means that a component can offer a complete function with meaningful information in the context of PIXEL.

Components of the dashboards & notifications tool:

- Back-End
- Front-End framework
- Front-End standard components
- Front-End PIXEL components
- Front-End database
- Alerts client
- Alerts back-end
- Alerts engine
- User front-end
- Security integration

In total there are 10 main components which provide 4 main features:

- Navigation
- Visualizations rendering
- Alerts & notifications
- Access control interfacing

The components that provide these features are interdependent between them but independent for the resulting tool, so we have 4 independent groups for a total of 10 components, modularity is then 40%.

4. % of reusable assets: 100%
Measured by reporting all the reusable components that are part of the dashboard and notifications module. A reusable component is considered any that can be applied in a different context of PIXEL with no modifications of the source code. The solution of the dashboards and notification is completely domain-agnostic as it is fully decoupled from the data sources. It can even work without the IH as far as it is replaced with Elastic or OpenDistro.

5. % of update: 100%
Measured by reporting the level of success in software updates on the dashboard and notifications module. It compares successfully completed updates versus all executed updates.
All software updates were successful.

6. Level of analysability: Yes
Measured by reporting the ratio between the numbers of items inside the dashboard and notifications for which logging is implemented compared to the number of items for which the specifications require logging.
Logging is only required once (SSO).

7. Mean CPU Utilisation: 0%

Dashboard server (REST API) consumes an average of 0.00% CPU. The server of the static files of the Dashboard consumes an average of 0.00% CPU. These metrics have been collected for 5 minutes with the system at rest, in a system that has a total of 4 virtual cores (vCPU).

8. Mean memory usage: 129MB

The Dashboard server (REST API) consumes an average of 115MB of RAM, so the system pre-allocates memory to have it available immediately if necessary. The server of the static files of the Dashboard consumes an average of 14MB of RAM. These metrics have been collected for 5 minutes with the system at rest, in a system that has a total of 16GB of RAM.

9. Maximum memory usage: 170MB

The Dashboard server (REST API) consumes an average of 120MB of RAM. The server of the static files of the Dashboard consumes an average of 50MB of RAM. These metrics have been collected for 5 minutes with the system at high load simulating 900 active simultaneous users, in a system that has a total of 16GB of RAM.

10. Maximum processing power used (CPU): 52.58%

Dashboard server (REST API) consumes an average of 42.53% CPU. The server of the static files of the Dashboard consumes an average of 10.05% CPU. These metrics have been collected for 5 minutes with the system in high load simulating 900 active simultaneous users, in a system that has a total of 4 virtual cores (vCPU).

11. Simultaneous requests : >900

Massive load tests have been carried out where the 2 main endpoints are requested simultaneously from the server (REST API), in addition to requesting the main static from the static server (necessary for clients to access the web from their web agents). The metrics have shown that the system is able to serve more than 900 simultaneous users (users who are interacting with the client at an average of 11 requests per minute).

12. % Monthly availability: 99.99%

The system of container-based deployments has been a key factor in reducing downtime, it has an uptime time (also known as SLA) of 99.99% which means a maximum downtime time of:

- Daily: 8s
- Weekly: 1m 0s
- Monthly: 4m 22s
- Quarterly: 13m 8s
- Yearly: 52m 35s

Since this downtime has only been a consequence of updates, it could be greater if the number of platform deployments is increased (currently it has been deployed 1 time per week). This means that if there are no software updates on the platform, the uptime time is 100%.

13. Mean number of errors per hardware or OS change/ upgrade: Not relevant

This indicator has not been tested as no hardware or OS changes/upgrades have been made during the pilot scope.

14. Mean number of errors per software install: Not relevant

The system is installed through containers; these have previously been generated by the development and maintenance team of the platform and then uploaded to a common repository (DockerHub) so that errors occur at the time of creation of the images and not during the installation or update time. Because if it has not been possible to generate a container, it is not uploaded to the repository and the system does not install / update it.

15. Mean number of errors per software uninstall: Not relevant
As in the previous case, when a piece of the platform is uninstalled, its container is deleted, since these encapsulate all its logic and dependencies, the system avoids having residual or poorly uninstalled processes.

16. Success rate: 100%

17. Dashboard availability: YesThe dashboard is available.

18. Notifications system availability: Yes

There are notifications in the task bar.

19. Average latency: 71ms (API-write), 70ms (API-read)

20. GUI module availability: Yes

This GUI is the PIXEL Dashboard itself. It covers all functionalities defined for the PIXEL project by redirecting to the specific models, as well as organizing flow between modules.

21. WCAG 2.0 Conformance Level: < A

The Web Content Accessibility Guidelines are a list of guidelines to follow to make the web content more accessible to people with disabilities. Those guidelines are ranked from A to AAA. By evaluating them, we can score our application based on the level of performance:

- Level A is the minimum level.
- Level AA includes all Level A and AA requirements. Many organizations strive to meet Level AA.
- Level AAA includes all Level A, AA, and AAA requirements.

For each guideline, we stated if the PIXEL Dashboard achieved it or not. Results of the WCAG evaluation are presented below:

Table 31. Results of the WCAG evaluation

Number	Guideline	Level	Summary	Done	Comments
Perceivable					
1.1.1	Non-text Content	A	Provide text alternatives for non-text content	No	
1.2.1	Audio-only and Video-only (Pre-recorded)	A	Provide an alternative to video-only and audio-only content	Yes	There is no video content on the platform.

1.2.2	Captions (Pre-recorded)	A	Provide captions for videos with audio	Yes	There is no video content on the platform.
1.2.3	Audio Description or Media Alternative (Pre-recorded)	A	Video with audio has a second alternative	Yes	There is no video content on the platform.
1.2.4	Captions (Live)	AA	Live videos have captions	Yes	There is no video content on the platform.
1.2.5	Audio Description (Pre-recorded)	AA	Users have access to audio description for video content	Yes	There is no video content on the platform.
1.2.6	Sign Language (Pre-recorded)	AAA	Provide sign language translations for videos	Yes	There is no video content on the platform.
1.2.7	Extended Audio description (Pre-recorded)	AAA	Provide extended audio description for videos	Yes	There is no video content on the platform.
1.2.8	Media Alternative (Pre-recorded)	AAA	Provide a text alternative to videos	Yes	There is no video content on the platform.
1.2.9	Audio Only (Live)	AAA	Provide alternatives for live audio	Yes	There is no audio content on the platform.
1.3.1	Info and Relationships	A	Logical structure	Yes	
1.3.2	Meaningful Sequence	A	Present content in a meaningful order	Yes	
1.3.3	Sensory Characteristics	A	Use more than one sense for instructions	No	

1.3.4	Orientation (WCAG 2.1)	AA	Content can be displayed in portrait and landscape orientation	Yes	
1.3.5	Identify Input Purpose (WCAG 2.1)	AA	Each input field must be able to be determined programmatically, a user should be able for example to autofill inputs	No	
1.3.6	Identify Purpose (WCAG 2.1)	AAA	Interface components, icons and landmarks (sections, article, main, etc.) must be able to be identified programmatically to help navigation for assistive technologies	No	
1.4.1	Use of Colour	A	Don't use presentation that relies solely on colour	Yes	
1.4.2	Audio Control	A	Don't play audio automatically	Yes	
1.4.3	Contrast (Minimum)	AA	Contrast ratio between text and background is at least 4.5:1	No	
1.4.4	Resize Text	AA	Text can be resized to 200% without loss of content or function	Yes	
1.4.5	Images of Text	AA	Don't use images of text	Yes	
1.4.6	Contrast (Enhanced)	AAA	Contrast ratio between text and background is at least 7:1	No	
1.4.7	Low or No Background Audio	AAA	Audio is clear for listeners to hear	Yes	No audio content
1.4.8	Visual Presentation	AAA	Offer users a range of presentation options	No	
1.4.9	Images of Text (No Exception)	AAA	Don't use images of text	No	Maps have text on tiles
1.4.10	Reflow (WCAG 2.1)	AA	User must be able to browse a website using a 320 pixel wide screen without having to	No	

			scroll horizontally (There are some exceptions)		
1.4.11	Non-Text Contrast (WCAG 2.1)	AA	Extend color contrast of at least 3:1 to non-text content such as infographics, diagrams, states, etc.	No	
1.4.12	Text Spacing (WCAG 2.1)	AA	Changing text style properties shouldn't break the page (line height, spacing after paragraph, letter spacing, word spacing)	Yes	
1.4.13	Content on Hover or Focus (WCAG 2.1)	AA	Elements that are being shown on focus or hover (skip navigation, tooltip) should be dismissible(Esc), hoverable, persistent	No	
Operable					
2.1.1	Keyboard	A	Accessible by keyboard only	No	
2.1.2	No Keyboard Trap	A	Don't trap keyboard users	No	
2.1.3	Keyboard (No Exception)	AAA	Accessible by keyboard only, without exception	No	
2.1.4	Character Key Shortcuts (WCAG 2.1)	A	If using single letter keyboard shortcut, the shortcut should be able to be turn off, or remap, or active only on focus	No	
2.2.1	Timing Adjustable	A	Time limits have user controls	No	
2.2.2	Pause, Stop, Hide	A	Provide user controls for moving content	Yes	
2.2.3	No Timing	AAA	No time limits	No	
2.2.4	Interruptions	AAA	Don't interrupt users	Yes	
2.2.5	Re-authenticating	AAA	Save user data when re-authenticating	Yes	
2.2.6	Timeouts (WCAG 2.1)	AAA	Users should be warned if user inactivity could cause data loss, unless data is preserved for more than 20h	No	

2.3.1	Three Flashes or Below	A	No content flashes more than three times per second	Yes	
2.3.2	Three Flashes	AAA	No content flashes more than three times per second	Yes	
2.3.3	Animation from Interactions (WCAG 2.1)	AAA	Motion animation triggered by interaction can be disabled	Yes	No motion animation on PIXEL
2.4.1	Bypass Blocks	A	Provide a ‘Skip to Content’ link	No	
2.4.10	Section Headings	AAA	Break up content with headings	No	
2.4.2	Page Titled	A	Use helpful and clear page titles	Yes	
2.4.3	Focus Order	A	Logical order	Yes	
2.4.4	Link Purpose (In Context)	A	Every link’s purpose is clear from its context	Yes	
2.4.5	Multiple Ways	AA	Offer several ways to find pages	No	
2.4.6	Headings and Labels	AA	Use clear headings and labels	No	
2.4.7	Focus Visible	AA	Ensure keyboard focus is visible and clear	Yes	
2.4.8	Location	AAA	Let users know where they are	Yes	
2.4.9	Link Purpose (Link Only)	AAA	Every link’s purpose is clear from its text	No	
2.5.1	Pointer Gestures (WCAG 2.1)	A	Complex gesture (Pinch, zooming, swiping) should have a simpler gesture alternative (Tap, double taps, long press)	No	
2.5.2	Pointer Cancellation (WCAG 2.1)	A	When using single pointer events, one of the following should be true, No Down-Event, Abort or Undo, Up Reversal, Essential	Yes	

2.5.3	Label in Name (WCAG 2.1)	A	Text in buttons or label should be readable by assistant technologies and can be used with Text-to-speech	Yes	
2.5.4	Motion Actuation (WCAG 2.1)	A	Functionalities trigger by moving the device should have a fallback without (Eg some apps use shake to undo)	Yes	No such case on PIXEL
2.5.5	Target Size (WCAG 2.1)	AAA	The size of the target for pointer inputs is at least 44 by 44 CSS pixels	No	
2.5.6	Concurrent Input Mechanisms (WCAG 2.1)	AAA	Inputs must to available to use with a different mechanism (Mouse, keyboard, stylus, touch, voice)	Yes	
Understandable					
3.1.1	Language of Page	A	Page has a language assigned	Yes	
3.1.2	Language of Parts	AA	Tell users when the language on a page changes	Yes	
3.1.3	Unusual words	AAA	Explain any strange words	No	
3.1.4	Abbreviation s	AAA	Explain any abbreviations	No	
3.1.5	Reading Level	AAA	Users with nine years of school can read your content	Yes	
3.1.6	Pronunciatio n	AAA	Explain any words that are hard to pronounce	No	
3.2.1	On Focus	A	Elements do not change when they receive focus	Yes	
3.2.2	On Input	A	Elements do not change when they receive input	No	
3.2.3	Consistent Navigation	AA	Use menus consistently	Yes	
3.2.4	Consistent Identification	AA	Use icons and buttons consistently	Yes	
3.2.5	Change on Request	AAA	Don’t change elements on your website until users ask	Yes	

3.3.1	Error Identification	A	Clearly identify input errors	Yes	
3.3.2	Labels or Instructions	A	Label elements and give instructions	No	
3.3.3	Error Suggestion	AA	Suggest fixes when users make errors	No	
3.3.4	Error Prevention (Legal, Financial, Data)	AA	Reduce the risk of input errors for sensitive data	No	
3.3.5	Help	AAA	Provide detailed help and instructions	No	
3.3.6	Error Prevention (All)	AAA	Reduce the risk of all input errors	No	
Robust					
4.1.1	Parsing	A	No major code errors	Yes	
4.1.2	Name, Role, Value	A	Build all elements for accessibility	No	
4.1.3	Status Messages (WCAG 2.1)	AA	Content that is updated dynamically must be notified to users of assistive technologies without getting visual focus	No	

From the above table, we can see that not all A characteristics have been implemented, preventing the PIXEL platform from reaching the A level.

Results for all KPIs are summarized in the table below :

Table 32. KPI summary for the PIXEL Integrated Dashboard and Notifications

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	Straightforward task accomplishment	Yes
Functional completeness	Portion of completed requirements	100%
Performance efficiency		

Time behaviour	Average Latency	71ms (API-write), 70ms (API-read)
Resource utilization	Mean CPU Utilisation	0%
	Mean memory usage	129MB
	Maximum memory usage	170MB
	Maximum processing power used	58.28%
Operability		
Ease of use	Dashboard availability	Yes
	Notifications system availability	Yes
	GUI module availability	Yes
Technical Accessibility	WCAG 2.0 Conformance Level	< A
Reliability		
Maturity	Simultaneous requests	>900
Availability	% Monthly availability	99.99%
	Success rate	100%
Maintainability		
Modularity	% of modularity	40%
Reusability	% of reusable assets	100%
Modifiability	% of update	100%
Analysability	Level of analysability	Yes
Portability		
Adaptability	Mean number of errors per hardware or OS change/upgrade	Not relevant
Installability	Mean number of errors per software install	Not relevant
	Mean number of errors per software uninstall	Not relevant

3.14. PIXEL Security and Privacy

3.14.1. Assessment scenario

KPIs assigned to the assessment (PIXEL Security assessment) have been described in D8.1, while the tools and methods for their collection have been defined in D6.3. In this section this methodology is further elaborated, and the assessment scenario is described in detail.

Functional suitability and **Maintainability** will be estimated using the expert judgment approach. **Automated measurements** are performed either by usage of existing evaluation software or by development of custom tools for this purpose. Part of the KPIs will be collected using **JMeter measurements.** The Apache JMeter™ application is an open-source software designed to load test functional behaviour and measure performance. **Performance efficiency** and **Reliability** have been measured using this approach. In order to assess the performance in the port area, measurements will be performed with a predefined set of realistic input data relevant to port operations. In the beginning, all measurements will be performed under laboratory conditions, and on the infrastructure, which will be defined in WP7 (cloud environment v.s. on-premises installation and other parameters).

Custom modules: reliability and **portability** are going to be measured using custom modules.

Reliability, portability and few other KPIs depend on the deployment of the modules in an operational scenario in order to measure them, as they are mostly statics related to an operational environment.

In the previous deliverable (D8.2) a set of KPIs were identified and the Security Layer was evaluated in an early stage. Some aspects were possible to evaluate whereas others were delayed or postponed for this second phase. The summary table is provided below and updated from D8.2.

Table 33. KPI for Security and Privacy

KPI	Measurement method	Reporting
Functional suitability		
Straightforward task accomplishment	Expert judgement	D8.2, D8.3
The portion of completed requirements	Expert judgement	D8.2, D8.3
Security		
Incidents of ownership changes and accessing prohibited data	Expert judgement	D8.3
Incidents of authentication mechanisms breaches	Expert judgement	D8.3
Level of User authenticity	Expert judgement	D8.3
Reliability		

Simultaneous requests	JMeter	D8.3
% Monthly availability	Custom module, Phase 2 based on NAGIOS Log	D8.3
Success rate	Custom module, Phase 2 based on Nagios Log	D8.3
Maintainability		
% of modularity	Expert judgement	D8.2
% of reusable assets	Expert judgement	D8.2
% of update	Expert judgement, Phase 2	D8.3
Level of analysability	Expert judgement	D8.2
Portability		
Mean number of errors per hardware or OS change/ upgrade	Custom module, Phase 2	D8.3
Mean number of errors per software change/ update	Custom module, Phase 2	D8.3
Mean number of errors per software install	Custom module, Phase 2	D8.3
Mean number of errors per software uninstall	Custom module, Phase 2	D8.3

For the purpose of data collection, a setup with two workstations has been used:

- PIXEL Security deployment in a workstation with:
  - Docker
  - Security (FIWARE KeyRock, Wilma and Authzforce)
  - Meter PerfMon
- Testing workstation: a workstation with installed JMeter probing and reporting tools.

The deployment workstation has the following specifications:

- CPU: Intel(R) Core (TM) i7-8559U CPU @ 3.40GHz
- RAM: 16 GB RAM
- SDD: 500 GB
- OS: Ubuntu 20.04 (64-bit)

3.14.2. KPI Data Collection and Results

1. Straightforward task accomplishment: Yes

"*Processes for authentication and authorization will be analyzed to verify that they do not include unnecessary steps. "* The process for authentication and authorization use the standard Oauth2 protocol and the mechanism developed by the FIWARE Foundation, with its Identity Management Components. Those protocols and mechanisms are compliant with the state of the art.

2. The portion of completed requirements: 100% (relevant)

"Should have" and "Must have" requirements from deliverable D3.2 will be taken as input in order to extract all requirements specifically targeting T6.6. Table below lists all PIXEL requirements related to the Security Layer that have the priority set to *"Should have"* or *"Must have"*. It also lists other PIXEL software modules related to the requirements and the status of development in the Security. The status does not assess the fulfilment of the requirement in other modules.

Table 34. PIXEL Security and Privacy Requirements ("Should have" and "Must have") and implementation status

Requirement	Addressed in additional modules	Implemented in Security component
Common non-functional requirements		
Security communications between components [68] *Status: The Security Layer provides API to manage user and role. It also provides components that can be used as API Gateway.*	DAL, IH , OT, DB	yes
Data source API connectivity [85] *Status: NGSI Agent that exposed an API are accessible through the PEP Proxy (OAuth2) using HTTPS.*	DAL	yes
Access Security [97] *Status: Security Layer provides components to secure the access to the PIXEL platform.*	DAL, IH , OT, DB	yes

A total of 3 requirements are related to functionality provided by the Security. Out of those, 1 are fully available, 2 in progress.

- Total requirements: 3
- Fulfilled requirements (functionality available): 3
- Portion of completed requirements: 100%

3. Incidents of ownership changes and accessing prohibited data: Not relevant

Even if the Security Layer provides mechanisms to control access resources, the architecture of the project (NGSI Agent implemented by trusted developers and deployed inside the infrastructure) and the user profile defined in the application UI that allows it to manage all the data, those controls were irrelevant. Only selected Data Source are able to push specifics data, and only fully identified users have been allowed to consult those data.

4. Incidents of authentication mechanisms breaches: 0

We don't notice authentication mechanism breaches. We notice wrong password usage, for example in Bordeaux in June and July we count 454 authentication requests with 68 failures. It is too low for an authentication attack. At the beginning of the Pilot we faced an SSH BotNet attack that tried to force the SSH root access of the servers. It was not directed against PIXEL and not directly relative to our security layer. As the SSH access is only possible using RSA key no breaches were detected. We changed the SSHD listen port to stop the attack.

5. Level of User authenticity: Yes

Authentication of the user is done using login/password. Only the admin of the application could grant access to the PIXEL application to a given user. The admin is in charge to ensure that it grants access to the subject that owns its credentials. Those credentials are used to identify the subject.

6. % Monthly availability: 100%

NAGIOS log provides information about all the components availability. A tool has been developed to parse that information. No incident has been detected during the active pilot phase.

7. % Success Rate: 100%

We didn't notice any incident with the security module. As far as we can see in the log analysis we don't detect requests that the security layer failed to manage. We notice some wrong access attempts (wrong password or expired token) but no failure on the treatment of the request.

8. % of update: 100%

Will be measured by reporting the level of success in software updates on the data acquisition layer module. It compares successfully completed updates versus all executed updates. The process to manage updates for the Security Layer as for the full platform relies on docker and docker-compose solutions. The advantage of those features guarantee the capability to reproduce an installation or update on all platforms, with the exact same process. All the installation and update files are shared on a GIT repository and all the platform specific parameters are stored in external files: .env and secrets. That way we can ensure that if an update works on the integration platform it will be successfully deployed on all platforms. All the security components were deployed in the state of the art version at the beginning of the pilots and no update has been needed.

Table 35. Update log

Date	**Version**	**Component**	**GPMB**	**ThPA**	**PPA**	**ASPM**	**Integration Platform**
10.03.2021	1.0.0	Keyrock + Wilma	Done	Done	Done	Done	Done
Total updates			**1**	**1**	**1**	**1**	**1**

9. Mean number of errors per hardware or OS change/upgrade: Not relevant

Will be measured by analysing the system & application error logs. This parameter is not relevant for the PIXEL Security Layer, as in principle it is not affected by hardware or software updates. (The use of docker prevent of this kind of issues)

10. Mean number of errors per software change/update: 0

Will be measured by analysing the system & application error logs. For the Security Layer, this equals the success rate provided in the *% of update* table. But as the security layer hasn't changed since the beginning of the pilot deployment the information isn't really relevant.

11. Mean number of errors per software install: 0
Will be measured by analysing the system & application error logs. An analysis will be performed by expert judgement instead of using custom-made modules. The operator noting problems during installation of the four pilots and the test of the platform provides an estimate of the mean number of errors per software installation. End of July no installation error were detected

12. Mean number of errors per software uninstall: 0
Will be measured by analysing the system & application error logs. Security Layer as the all platform is distributed as a set of Docker images and installed using Docker Compose. Consequently, the uninstallation procedure is very simple. Using the single 'docker-compose down' command Data Acquisition Layer can be removed. Using the '-v' switch the persisted data (state) is removed as well.

13. Mean CPU utilization: 34.72%

Memory CPU and memory utilization is provided in the table below. All values are provided in %.

Table 36. PIXEL Security Load Test result

Test ID	CPU mean	CPU max	Memory mean	Memory max	Throughput	Latency
1	35.72	49.33	75.47	76.42	630.03	646
2	33.73	40.45	76.21	76.57	622.57	53

14. Mean memory usage: 75.84MB

15. Maximum memory usage: 76.5MB

16. Maximum processing power used: 49.33%

17. Average latency: 55ms
The time response was quite constant

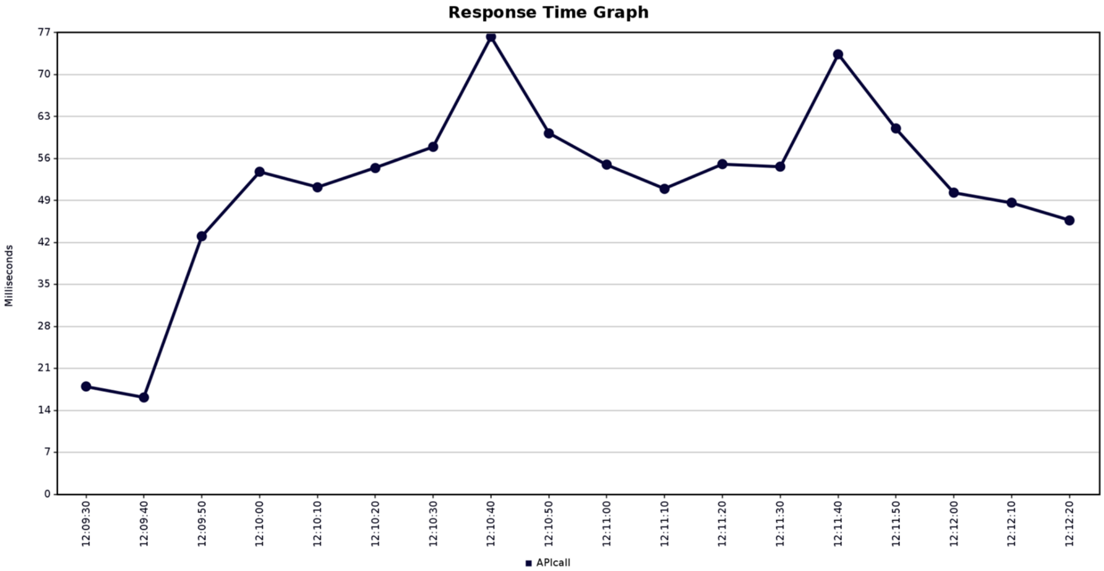

Figure 25. Response time analysis for the PIXEL Security and Privacy

18: Simultaneous requests: 100/s

We have run several tests to evaluate its usage in the PIXEL context. We have connected behind Wilma a mock server to measure the capabilities of Wilma to manage throughput and simultaneous requests. The test of the Mock server with the configuration shows that he can manage a lot more requests than Wilma: more than 2500 requests per seconds, with 400 simultaneous clients and a response time around 2ms. We run a first 3 minutes test with a linear augmentation of the number of clients to reach 400 after 2 minutes, each client trying to send up to 10 requests per second. Wilma can handle 600 requests per second, but the response time climbs to 1s. We run a second test with 100 clients sending 5 requests per second.

Table 37. PIXEL Security Load Tests

Test ID	Test setup				Test execution - achieved performance
	clients	req/s/client	Total req/s	Requests (3 min)	Requests (3 min)
1	=>400	~10			113 787
2	100	6.5	650	117000	112 095

In this configuration the Security Layer could handle 100 simultaneous requests with a good quality of service. But more than the number of simultaneous requests, the test shows that the server is able to manage up to 600 requests per seconds, after that the response time increases and the quality of service decreases.

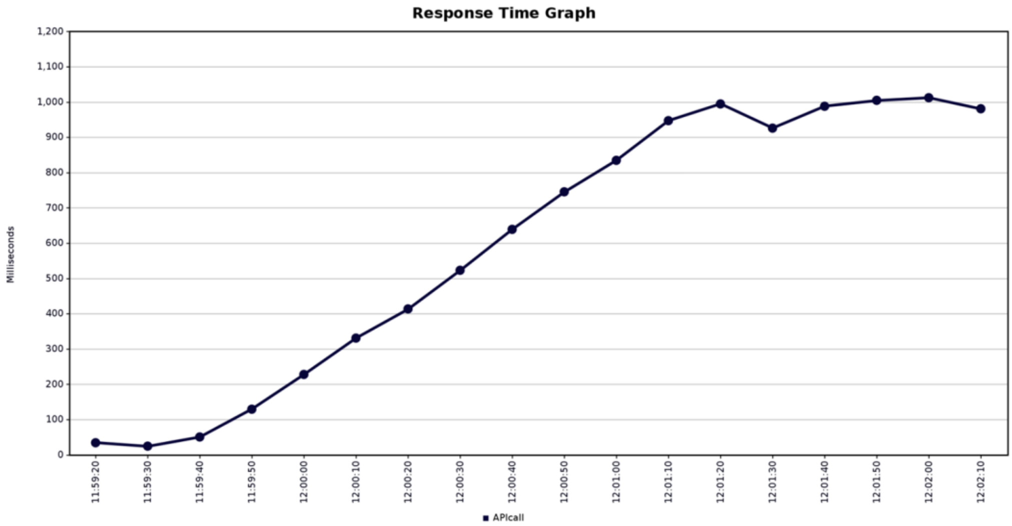

Figure 26. Simultaneous requests analysis for the PIXEL Security and Privacy

19. GUI module availability: Partially

Keyrock comes with an administrative GUI that allows to manage roles, permissions, users, organizations and applications.

PIXEL user/administrator should use the dashboard features to manage PIXEL users and roles. On the PIXEL project, only the integrator uses those GUI to configure special rights in order to access NGSI agents that expose an API.

20. WCAG 2.0 Conformance Level: None

As there is no GUI module available, the PIXEL Dashboard is considered the only dashboard to be used by all user types and as such, it is the one that will receive the WCAG evaluation.

21. % of modularity: 100%

Will be measured by reporting all the independent components that are part of the security module and comparing them to the number of all components in the security module. Individual operation means that a component can offer a complete function with meaningful information in the context of PIXEL.

As defined in WP6 deliverables, PIXEL Security Layer is composed of several components that provide different feature of the Security implementation. Those components are FIWARE Generics Enabler that implements Identity Management (Keyrock), Authorization (AuthZForce) and Access control (PEP Proxy Wilma).

22. % of reusable assets: 100%

Will be measured by reporting all the reusable components that are part of the security layer module and comparing them to the number of all components in the Security. A reusable component is considered any that can be applied in a different context of PIXEL with no modifications of the source code.

All modules in the PIXEL Security Layer are FIWARE Generic Enablers that could be reused on any FIWARE compatible projects.

Results for all KPIs are summarized in the table below :

Table 38. KPI summary for the Security and Privacy module

Sub-characteristics	KPI	Result
Functional suitability		
Functional appropriateness	Straightforward task accomplishment	Yes
Functional completeness	Portion of completed requirements	100% (relevant)
Performance efficiency		
Resource utilization	Mean CPU Utilisation	34.72%
	Mean memory usage	75.84MB
	Maximum memory usage	76.5MB
	Maximum processing power used	49.33%
Operability		
Ease of Use	GUI module availability	Partially
Technical accessibility	WCAG 2.0 Conformance Level	None
Reliability		
Maturity	Simultaneous requests	100/s
Availability	% Monthly availability	100%
	Success rate	100%
Security		
Confidentiality	Incidents of ownership changes and accessing prohibited data	Not relevant
Integrity	Incidents of authentication mechanisms breaches	0

Authenticity	Level of User authenticity	Yes
Maintainability		
Modularity	% of modularity	100%
Reusability	% of reusable assets	100%
Modifiability	% of update	100%
Portability		
Adaptability	Mean number of errors per hardware or OS change/upgrade	Not relevant
	Mean number of errors per software change/update	0
Installability	Mean number of errors per software install	0
	Mean number of errors per software uninstall	0

4. Technical Impact Assessment of the PIXEL Use Cases

The technical impact assessment of the PIXEL Use Cases aims to evaluate characteristics and sub-characteristics that are defined by 2 ISO/IEC norms, ISO/IEC 25010:2011 and ISO/IEC 25012:2008. Partners selected in D8.1 the sub-characteristics that were of interest for the PIXEL project and defined a way to calculate them. Calculation is mainly based on two questionnaires:

- The Quality In Use Model, to evaluate sub-characteristics of ISO/IEC 25010:2011, relates to the outcome of interaction when a product is used in a particular context of use.
- The Data Quality Model, to evaluate sub-characteristics of ISO/IEC 25012:2008, is a general Model for data retained in a structured format within a computer system.

Additional sub-characteristics are also calculated using quantification methods, like the number of implemented requirements or the number of completed user-stories based on what was defined in WP3. Questionnaires for the 2 above models are derived from 2 known questionnaires:

- The TAM3 which is an information model theory that aims to model how end-users of a system may come to accept and use it.
- The AIMQ which is a complete methodology for information quality assessment.

Each port listed the end-users that were impacted by the platform, and thus had to answer the questionnaires, in table 7 of D8.1 that is reminded below.

Table 39. Port users' classification - Table 7 of D8.1

Port	Primary Users	Secondary Users	Indirect Users
GPMB	Statistics Manager Energy Manager Port Manager	IT Manager Software Editor	Environmental Manager Port Agent/Operator
ASPM	Environmental Manager Parking area Manager	Software Editor	Gate/Access Manager
PPA	Environmental Manager Management team	IT Department	Quality Assessment
ThPA	Environmental Manager	IT Manager	Terminal Operator

Each user had to provide multiple answers:

- The Quality in use questionnaire had to be filled three times per user:
  - Once evaluating the old platform (if available)
  - Once evaluating the PIXEL platform for their use case
  - Once evaluating the PIXEL platform for the PEI use case
- The Data quality questionnaire had to be filled twice per user:

- Once evaluating the PIXEL platform for their use case
- Once evaluating the PIXEL platform for the PEI use case

Results of the quality in use-questionnaire were provided on a 1-7 likert scale, and results of the Data quality questionnaire on a 0-10 scale, which are converted to percentages when aggregated for the KPIs for better understanding. We also list, for every use-case, the list of implemented requirements and completed stories, that allow to calculate remaining sub-characteristics that were not answerable with the questionnaires.

It is important to mention that the successful deployment of the PIXEL platform got delayed due to COVID-19 pandemic outbreak consequences that led to an extension of the project, and was achieved at the end of WP7 in M38. Thus, many ports did not have enough time to gain an extensive experience using it: It was decided, realising the previous, that such evaluation started using a partial version of the system. At the end of the process, stakeholders in ports answered both questionnaires at the same time, which was great as it allowed a good comparison between the old system of the port and the PIXEL platform because they could directly compare them while answering the questionnaires.

Also, it must be considered that this represents a frozen version of the state of the PIXEL at the end of summer 2021, and that it tends to evolve quite quickly (and will continue to evolve thanks to the work by PIXEL Association).

The sections below present the results of the evaluations in the aggregated versions. Links to the view-only versions of the questionnaires and detailed results are available in the Annexes at the end of this document. For each use-case, we establish what can be deduced from the results. In the last section, we also compare the results between all use-cases.

For the below evaluations, we chose to use a color code associated with the calculated KPI for a better understanding. Here is the color convention:

- KPIs with value 75% or more are coded green ●, this is considered as a very good result.
- KPIs with value 50% up to 75% are coded yellow ●, this is considered as an acceptable result for a research project.
- KPIs with value less than 50% are coded red ●, this is considered as improvable.

As results from the questionnaires are quite high, we estimate that only those KPIs that have been scoring less than 75% should be explained.

For the Data Quality Model, we have also calculated in every section the PSP/IQ Model quadrants introduced in table 37 of D8.2. This allows us to show usability roles gaps in the 4.6 section.

4.1. Energy Management Use Case - GPMB

4.1.1. Data collection and user stories

4.1.1.1. Requirements completion

Table 40. Functional requirements of the Energy Management Use Case

Requirement	**Status (Done / Not fully done / not dOne)**	**Comments**
Common functional requirements		
Import historical data [36]	Done	Specific NGSI agents have been developed and deployed to import historical data of vessels calls, MARPOL data, ships air emissions, waste of terminals. This was a "must have" requirement.
Interaction with models [41]	Done	Results of the PAS model are used as inputs of PEI. This was a "must have" requirement.
Anomaly and event list [44]	Partially done	The Alerts/thresholds systems allows to set specific range of values above/below which an alert would be triggered.
Anomaly and event detection [45]	Done	
Homogenize data [61]	Done	
Catalogue of models [62]	Done	
Detection of anomalies [63]	Half done	The Alerts/thresholds systems allows to set specific range of values above/below which an alert would be triggered.
Feedback [64]	Not done	Technologically not included. Available via JIRA tickets and specific request to technical partners.
Centralized user administration system [65]	Done	
Configurable Dashboard [66]	Done	
UI Notification System [67]	Done	Through PIXEL Dashboard. Not straightforward for stakeholders.
Port Operational KPI list [70]	Partially done	Covered by OTs. Not straightforward for

		stakeholders
Operational Interface [71]	Done	
Analyze historical data [81]	Done	
Support for manually provided data [86]	Done	
Discovery service for data [104]	Done	
Visualization of data [105]	Done	
Use-case specific functional requirements		
Support electricity consumption sensors [9]	Not fully done.	An NGSI agent has been developed but due to a technical and administrative issue the raw data was not available.
Access to traffic data [10]	Done	PIXEL collects data of vessel calls both the historical one and the coming one. Two specific NGSI Agents have been developed.
Monitor expected port calls [11]	Done	PIXEL collects data of vessel calls both the historical one and the coming one. Two specific NGSI Agents have been developed.
Collect sensor data through Port Community System (VIGIEsip) [12]	Done	PIXEL gathers from VIGIEsip MARPOL data, vessel calls.
Support Air Quality Sensors [14]	Done	A NGSI agent has been developed to obtain data from the ATMO station. Data is available through the dashboard.
Modelling and analysis of energy consumption during ship handling procedures [15]	Done	The PAS model has been developed during WP4 and integrated during WP7 and allows the obtained and analysed energy consumption.
Support wind speed sensors [16]	Done	A NGSI agent has been developed to obtain data from the Sencrop weather station. Data is available through the dashboard.
Support weather sensor/service [17]	Done	A NGSI agent has been developed to obtain data from the Sencrop weather station. Data is available through the dashboard.

Support old sensors (gauge stations network [18]	Partially achieved.	No specific (physical) Gateway has been developed within the framework of PIXEL to add communication capacities, in particular on sea level sensors. However, this data is nevertheless gathered into PIXEL and accessible.
Optimization of photovoltaic energy production and consumption [19]	Done	In the context of WP4, data analysis has been performed and 3 possibilities of integration of PV systems has been tested and analysed.
Monitoring l'Ostrea dredge environmental impact [20]	Not done	PEI model exists but the impact of the dredge is currently not included. This is mainly explained by the fact that the dredge work is not available in the vessel calls list and that the set of environmental impacts selected for the PEI does not include dredging effects.
Monitor energy consumption of the port authority [22]	Done	The PAS model has been developed during WP4 and integrated during WP7 and allows the obtained and analysed energy consumption. The PAS model has been scheduled on a monthly basis and instant and cumulative visualisation of energy is available.
Expose data to VIGIEsip system [82]	Done	API for IH is available in order to expose PIXEL data to VIGIEsip system

4.1.1.2. User Stories completion

Table 41. User stories completion of the Energy Management Use Case

As a/an	I want to …	So that …	Completable (Yes/No/ Partial)	Comments
Statistics manager	Analyse the structure and periodicity of ships calls from the internal database including notions of time of call, goods, tonnages, berth, etc…	I could estimate the average call time of a targeted piece of goods taking into account the berth and potential seasonality	Yes	During WP4 a data analysis of vessel calls has been performed. The ETD model has been developed in order to predict the average time of a call and the ETD. A Gantt visualisation of a

	Be able to update this database annually by adding ships calls of the past year	I could anticipate the possible evolution inherent to a given piece of goods	Yes	vessel's calls is available in the dashboard with info associated to the vessel calls. Vessel calls are updated almost in real time and stored in the PIXEL Information Hub.
Energy manage r	Evaluate/Quantify the energy consumption of each logistic chain model by measuring the consumption of each element related to loading/unloading considering technical features	I could determine the relative share of each energy in the targeted logistic model and identify the potentially interesting elements for renewable energy injection	Yes	During WP4 the PAS model has been designed and developed in order to model the port activities. The PAS model has been developed based on this user story. The PAS model has been tested and deployed in the PIXEL platform in WP7
	Be able to update this database at any time by adding/substituting/erasin g components	I could obtain the most reliable data	Yes	During WP7, a specific user interface was developed in order to let the users define the port parameters and the supply chain specification. These PAS forms are available through the PIXEL dashboard and data are stored in the PIXEL Information Hub.
	Obtain an average value of energy consumption for each element whose actual data would not be available based on technical characteristics of the machines	I could substitute a missing item by an average value	Yes	The PAS model has been developed in order to let the user choose the level of accuracy for the modelling of supply chains. In WP4 a list of methods to substitute a missing item have been reviewed.
	Link the results calculated before, namely: ships call data analysis and energy consumption of the logistics chains	I could determine the energy consumption of any ship that has called to Bordeaux in the past and estimate the likely energy consumption of future ships whatever their goods.	Yes	The PAS model results provide the energy consumption, pollutants emissions per loading/unloading operation for a specific vessel call. Instantaneous and cumulative visualisation are available.
	Analyse the distribution structure of electrical energy on Bassens from sensors available on the entire terminal	I could study from a data collection platform the consumption structure of all or part of a targeted area in order to	No	An NGSI agent has been developed but due to a technical and administrative issue the raw data was not available.

		distribute the adequate proportion of electricity according to needs.		
	Measure the real conditions of sunshine from a weather station located in the port in Bassens	I could evaluate precisely the amount of solar energy produced and adapt the need of conventional electricity. I could accurately determine the handling downtime due to rain or wind type situations, too.	Partial	The real conditions of sunshine have not been measured. However, the potential of the photovoltaic production has been estimated for GPMB based on the PGVIS database.
IT manager	Reduce the cost of ownership of connected sensors in the port and simplify the addition of new sensors	I can multiply the number of connected sensors in the port area to raise the knowledge on the evolution of the environment and on ports operations	Partial	No specific Gateway has been developed within the framework of PIXEL to add communication capacities, in particular on sea level sensors. However, this data is nevertheless gathered into PIXEL and accessible.
	Have more calculation power (thanks to cloud computing)	I can get more useful data analysis and provide optimisation solutions	No	Cloud computing (high performance computing) has not been addressed in PIXEL.
Environmental manager	Promote my actions done for the protection of the environment and for the mitigation of port activities	The port-city relationships and acceptance of port activities could improve	yes	PEI has been installed, deployed and run IN GPMB. PEI results can be used to promote GPMB actions.
Port manager	Estimate if the investment in solar panels on the rooftops of the port's warehouses is valuable	I can decide whether to invest or not	Yes	During WP4 an analysis of the photovoltaic production potential of GPMB has been performed based on 3 different scenarios.
	Assess the relevance to add new functionalities in PCS (Port Community Systems) such as port environmental index,	I can define an appropriate roadmap for VIGIEsip	Yes	PAS and PEI results can be integrated in the PCS using the Information Hub API.

	outcomes of cloud computation…			
Software editor	Assess the relevance to add new functionalities in PCS (Port Community Systems) such as port environmental index, outcomes of cloud computation	I can define an appropriate roadmap for VIGIEsip	Yes	PAS and PEI results can be integrated in the PCS using the Information Hub API.
Port agent/operator	Master my energies consumptions	I can save money by optimising these consumptions	Yes	PAS model allows us to understand and model the energy consumption of GPMB. These results can be used as a decision making tools to save energy (test of different scenarios)
	Buy cheaper green electricity	I can save money and contribute to actions for climate change	Partially	No PV systems have been installed during the PIXEL project but GPMB will deploy a PV system after PIXEL project

4.1.1.3. Other data collection

Table 42. Other data collection for the Energy Management Use Case

Requested data	Value
Number of end-users that were planned to use the platform	7
Number of end-users that at least tried to use the platform	7
Number of end-users that are really using the platform	1
Number of sensors connected to the local IoT platform	12 (9 waves, 1 particle, 1 sound, 1 light)
Number of different sensors connected to the local IoT platform	4

4.1.2. Data analysis (CATIE)

4.1.2.1. Results of the Quality in use model

As GPMB doesn't have an old system, we only evaluate the PIXEL platform.

Table 43. Results of the Quality in use model for the Energy Management Use Case

Sub-characteristic	KPIs	Old System	PIXEL platform
Effectiveness			
Effectiveness	% of completed user stories	-	🟡- 68,75%
	Output Quality	-	🟢- 97,14%
Efficiency			
Efficiency	Efficiency level (Uses the Number of end-users KPI)	-	🔴- 11,85% (summer holidays and users unreachability)
Satisfaction			
Usefulness	Usefulness level	-	🟡- 64,12%
	Perceived usefulness	-	🟢- 81,43%
Trust	Trust level		🟢- 90,48%
	System Serenity	-	🟢- 100%
Context coverage			
Context completeness	Completeness level	-	🟢 - 98,81%
Flexibility	Flexibility level	-	🟢- 100%

We obtained 7 different answers to the questionnaires: all evaluating the PIXEL platform, composed of 3 for the primary users, 2 for the secondary users and 2 for the indirect users. Those answers were provided by 7 different people. PIXEL platform was recently implemented in GPMB and it was summer holidays for many of the collaborators, as such not all end-users had a chance to fully use the platform (*Efficiency level* low KPI). Indeed, even if there are functionalities in the platform, users did not know how to fully exploit them yet (due to at that moment not-enough skilled users were available and the documentation of the models and views was not finished), impacting the *% of completed user stories* and completed requirements (reflected by the *Usefulness level*) felt by the end users. To improve this, GPMB and technical partners will continue to exchange information about the platform until the end of the project.

4.1.2.2. Results of the Data Quality Model

Table 44. Results of the Data quality model for the Energy Management Use Case

Sub-characteristic	KPIs	Calculation Type	
Information Accuracy			
Currentness	Timeliness	AIMQ questionnaire	🟢- 94.8%

Correctness	Free of errors	AIMQ questionnaire	🟢- 93.5%
Credibility	Believability	AIMQ questionnaire	🟢- 97%
Precision	Precision	AIMQ-like questionnaire	🟢- 96.5%
Traceability	Traceability	AIMQ-like questionnaire	🟢- 79.8%
Information Accessibility			
Accessibility	Accessibility	AIMQ questionnaire	🟢- 80%
Information Appropriateness			
Understandability	Understandability	AIMQ questionnaire	🟡- 71.5%
Value Added	Advantage	AIMQ-like questionnaire	🟢- 100%
	Relevancy	AIMQ questionnaire	🟢- 92.5%
Representational Adequacy	Concise representation	AIMQ questionnaire	🟢- 98%
	Interpretability	AIMQ questionnaire	🟢- 82.8%
Consistency	Consistent representation	AIMQ questionnaire	🟢- 98%
Completeness	Number of sensors / devices connected to the local IoT platform	Count the number of sensors connected to the local IoT platform.	12
	Number of types of data (sensors) connected to the local IoT platform.	Count the number of different sensors connected to the local IoT platform.	4
	Completeness	AIMQ questionnaire	🟢- 95.7%
Efficiency			
Efficiency	Ease of Operation	AIMQ questionnaire	🟢- 100%
Availability			
Availability	Availability	AIMQ-like questionnaire (reworked from Security)	🟢- 100%
	Security	AIMQ questionnaire	🟢- 100%
Portability			
Portability	Portability level	AIMQ-like questionnaire	🔴- 43.5%

We obtained 7 different answers to the questionnaires: composed of 3 for the primary users, 2 for the secondary users and 2 for the indirect users. Those answers were provided by 7 different people. GPMB is the port with the highest number of sensors, this shows early adoption of the platform and a volunteer effort to pursue using it. Understandability could be better, and reflects what has already been highlighted by the quality in use model that end-users have to gain experience using the platform to understand how to fully use the data. As all data is integrated within PIXEL, the Portability KPI has not been a priority in PIXEL and the interpretation of this result is that (whenever using PIXEL), a stakeholder may necessitate technically skilled staff to port data and results.

We can use the above to calculate the PSP/IQ Model quadrants introduced in table 37 of D8.2:

Table 45. PSP/IQ Model quadrants for the Energy Management Use Case

PSP/IQ Quadrants	Global	PRIMARY	SECONDARY	INDIRECT
Sound information	97.2%	99.4%	-	93.9%
Useful information	82.3%	83.7%	-	80.1%
Dependable information	100%	100%	-	-
Usable information	90%	90%	-	-

4.2. Intermodal Transport Use Case - ASPM / SDAG

4.2.1. Data collection and user stories

4.2.1.1. Requirements completion

Table 46. Functional requirements of the Intermodal Transport Use Case

Requirement	Status (Done / Not fully done / not DOne)	Comments
	Common functional requirements	
Import historical data [36]	Done	Historical data concerning SILI and pollution (for PEI calculation) have been imported. NGSI agents are able to feed IH with existing data.
Interaction with models [41]	Done	
Anomaly and event list [44]	Not realised (half done)	The Alerts/thresholds systems allows to set specific range of values above/below which an alert would be triggered.
Anomaly and event detection [45]	Done	
Homogenize data [61]	Done	
Catalogue of models [62]	Done	
Detection of anomalies [63]	Not realised (half done)	The Alerts/thresholds systems allows to set specific range of

		values above/below which an alert would be triggered.
Feedback [64]	Done	By means of integration between the PIXEL infrastructure and the port's website.
Centralized user administration system [65]	Done	
Configurable Dashboard [66]	Done	
UI Notification System [67]	Done	
Port Operational KPI list [70]	Done	
Operational Interface [71]	Done	
Analyze historical data [81]	Done	
Support for manually provided data [86]	Done	
Discovery service for data [104]	Done	Not straightforward for the stakeholder. The user can visualise all data integrated in the platform by creating a custom View on the different IH ElasticSearch indices existing in the database.
Visualization of data [105]	Done	
Use-case specific functional requirements		
Integration with the SILI Information System [23]	Done	
Integration with the PMIS2 Information System [24]	Partially done	Due to unavailability of the PMIS2 Information System for administrative and security reasons, Vessel Calls have been collected by integrating the PIXEL platform with the ASPM's website (by means of an ad hoc NGSI agent)
Integration with ASPM video monitoring system [25]	Done	

Traffic peak and congestion monitoring at the port facility [26]	Done	Done thanks to smart capabilities provided by ASPM video monitoring system.
Integration with the SDAG Access Control System [27]	Done	
Integration with data provided by sensors, cameras and feeds by third parties [28]	Done	
Cooperation with railway authorities [29]	Not Done	Could have priority.
Provide a common access for management and monitoring of ADR [30]	Not Done	Should have priority.
Truck re-routing alerting system for operators [31]	Done	
Truck re-routing alerting system for final users [32]	Not Done	Due to privacy issues
Truck re-routing booking system [33]	Not Done	
Port congestion forecasting [34]	Done	
Port congestion simulation [35]	Done	
Port - SDAG highway congestion forecasting [37]	Done	

4.2.1.2. User stories completion

Table 47. User stories completion of the Intermodal Transport Use Case

As a/an	I want to …	So that …	Completable (Yes/No/Partial)	Comments
Port of Montfalcone				

Gate/Access Manager	Have automatic predictions of parking occupancy in the port entry parking area using the actual parking occupancy, the port gate flows and the vessels scheduling and historical traffic data on a daily basis with "some" hour range	Truck operators can be notified of congestion of port access and parking availability / predictions if they overpass certain threshold, as well as other stakeholder (municipality, police), in order to evaluate proper actions to minimize the issue and port-city interference	Partial	Not specific within the visualisation, but the Otools include the possibility to establish thesholds and alerts. User could select proper information index in the database, establish thresholds and be notified.
		In case a parking is full (or almost) and the automatic predictions of parking occupancy forecast an increment in traffic flows and parking needs, truck drivers/operators can be notified of it and linked to SDAG in order to reroute their parking destination towards the interport or delay their arrival to the port/parking area	yes	The availability of SDAG parking area is shared on the Monfalcone Port website on hourly basis.
Environmental Manager	Be able to collect and analyse environmental data	I can support the activities of the Regional Environment and Health Observatory and plan future investments/procedures to promote a greener port	yes	2 new type of environmental data have been added by new sensor in order to make more complete the monitoring of the PEI as additional and available tool for the Regional territory.
Software editor	Assess the relevance to add new functionalities in SILI	I can define an appropriate roadmap for SILI	no	No additional comments. SILI used for traffic prediction. This functionality was disregarded during the pilot execution.

SDAG				
Parking area manager	receive automatic alerts of trucks diverted to SDAG from the port of Monfalcone and confirm slot availability	I can check automatically the availability of parking slots and/or reserve + address trucks to the different parking areas	Partially	Availability of parking slots is verified by SDAG control access system and shared on the Monfalcone Port website on hourly basis were there is a link to SDAG parking area website from which the parking slots can be reserved, but no automatic alert has been developed.
	have an automatic booking system for trucks that are diverted to SDAG	I can reduce/optimize manual work from the internal personnel and the use of resources	yes	No additional comments.
	receive automatic alerts of ADR (dangerous) transport coming to SDAG	I can reinforce the security (in a dedicated parking area) related to ADR transport and/or divert trucks in other infrastructures	no	No additional comments. No additional comments. SDAG parking areas can monitor ADR transport in dedicated areas. No automatic alerts have been developed
	have anticipations or simulation of the traffic congestion in the port/surrounding areas (through	"I can estimate the n° of trucks coming to SDAG and I can evaluate the use of all available resources (ports, inland ports and railway) to address the traffic towards other multimodal transport and	Partially	The availability of the Traffic Prediction and Intermodal model allows estimating the traffic on the regional territory based on 2 perspectives (vessels calls and road traffic).

	a system that put in communication more stakeholders thanks to SILI platform, for example: Autovie Venete and other authorities)	support the decision making in addressing the trucks towards Monfalcone or other infrastructures. Furthermore, I can improve all the services offered to the truck drivers in the inland ports"		

4.2.1.3. Other data collection

Table 48. Other data collection for the Intermodal Transport Use Case

Requested data	Value
Number of end-users that were planned to use the platform	4
Number of end-users that at least tried to use the platform	12
Number of end-users that are really using the platform	9
Number of sensors connected to the local IoT platform	3
Number of different sensors connected to the local IoT platform	No additional sensors, but 3 external information systems

4.2.2. Data analysis

Extended from table 8 (Quality In Use Model evaluation criteria) of D8.1.

4.2.2.1. Results of the Quality in use model

Table 49. Results of the Quality in use model for the Intermodal Transport Use Case

Sub-characteristic	KPIs	Old System	PIXEL platform
Effectiveness			

Effectiveness	% of completed user stories	-	🔴 - 37,5% (rationale above and reduced due to the introduction of a new, whole different use-case within the pilot).
	Output Quality	72,79%	🟢- 77,29%
Efficiency			
Efficiency	Efficiency level (Uses the Number of end-users KPI)	-	🟢- 115,08%
Satisfaction			
Usefulness	Usefulness level	-	🟢 - 79,46%
	Perceived usefulness	72,96%	🟢- 83,24%
Trust	Trust level	78,57%	🟡- 69,05%
	System Serenity	100%	🟢- 88,10%
Context coverage			
Context completeness	Completeness level	71,43%	🟢- 85,12%
Flexibility	Flexibility level	71,43%	🟢- 84,52%

We obtained 23 different answers to the questionnaires:

- 8 evaluating the old system, composed of 2 for the primary users, 1 for the secondary users and 5 for the indirect users. Those answers were provided by 7 different people.
- 15 evaluating the PIXEL platform, composed of 6 for the primary users, 2 for the secondary users and 7 for the indirect users. Those answers were provided by 12 different people.

Some people, involved in two different roles within the organisation, responded multiple times. Analysing those results helps us to detect some ideas that are of interest:

- ASPM/SDAG/INSIEL used to have a legacy system, but we can see here with the number of answers that new people are coming to use the PIXEL platform. This is also confirmed by the high *Efficiency level* KPI, and can be explained by the increased Perceived usefulness (from 72,96% to 83,24%) which means that people tend to find PIXEL useful, and may then attract some other people to use it.
- *Trust level* and *System serenity*, however, decreased a bit. This seems legit for a system which is quite new for the port and for which they did not have much time to gain experience with it.
- *Context completeness* and *Flexibility level* show great scores and improved a bit, which means that PIXEL works in most of the specified contexts of use and even beyond.
- The low *% of completed user stories*, along with the *usefulness level*, show that even with the maturity they have using the platform, either the required functionality has not been implemented or users do not know how to use it (the most cases). This also responds to the evolution of the pilot. During the pandemic outbreak, the possibility of establishing a new use-case within the pilot was studied. A COVID-19 social distance simulation tool based on the results of the PAS model was included. **According to was agreed via Amendment #2**, this meant a slight decrease on the expectations over the original pilot (T7.2) for the sake of T7.7.

It is also worth to note that this was the first port to complete the questionnaires, showing a great maturity of using the platform.

4.2.2.2. Results of the Data Quality Model

Table 50. Results of the Data quality model for the Intermodal Transport Use Case

Sub-characteristic	KPIs	Calculation Type	
Information Accuracy			
Currentness	Timeliness	AIMQ questionnaire	🟢- 79.3%
Correctness	Free of errors	AIMQ questionnaire	🟢- 76.3%
Credibility	Believability	AIMQ questionnaire	🟢- 77.1%
Precision	Precision	AIMQ-like questionnaire	🟢- 75%
Traceability	Traceability	AIMQ-like questionnaire	🟢- 76.3%
Information Accessibility			
Accessibility	Accessibility	AIMQ questionnaire	🟢- 86.7%
Information Appropriateness			
Understandability	Understandability	AIMQ questionnaire	🟡- 70.8%
Value Added	Advantage	AIMQ-like questionnaire	🟢- 83.3%
	Relevancy	AIMQ questionnaire	🟢- 79.6%
Representational Adequacy	Concise representation	AIMQ questionnaire	🟢- 90%
	Interpretability	AIMQ questionnaire	🟡- 61.3%
Consistency	Consistent representation	AIMQ questionnaire	🟢- 78.3%
Completeness	Number of sensors / devices connected to the local IoT platform	Count the number of sensors connected to the local IoT platform.	3
	Number of types of data (sensors) connected to the local IoT platform.	Count the number of different sensors connected to the local IoT platform.	3
	Completeness	AIMQ questionnaire	🟢- 79.3%
Efficiency			
Efficiency	Ease of Operation	AIMQ questionnaire	🟢- 82%
Availability			

Availability	Availability	AIMQ-like questionnaire (reworked from Security)	●- 80%
	Security	AIMQ questionnaire	●- 94.2%
Portability			
Portability	Portability level	AIMQ-like questionnaire	●- 75.6%

We obtained 7 different answers to the questionnaires: composed of 3 for the primary users, 1 for the secondary users and 3 for the indirect users. Those answers were provided by 7 different people.

Results of the data quality model show that the data inside the platform is suitable for the use case. The medium values of Understandability and Interpretability are not critical and only show that users may incorporate the provided documentation of the platform within their work.

We can use the above to calculate the PSP/IQ Model quadrants introduced in table 37 of D8.2:

Table 51. PSP/IQ Model quadrants for the Intermodal Transport Use Case

PSP/IQ Quadrants	Global	PRIMARY	SECONDARY	INDIRECT
Sound information	82.7%	88.9%	-	71.7%
Useful information	70.6%	77.4%	-	49.7%
Dependable information	94.2%	94.2%	-	-
Usable information	84.3%	84.3%	-	-

4.3. Port City Integration Use Case - THPA

4.3.1. Data collection and user stories

4.3.1.1. Requirements completion

Table 52. Functional requirements of the Port City Integration (THPA) Use Case

Requirement	Status (Done / Not fully done / not DOne)	Comments
Common functional requirements		
Import historical data [36]	Done	
Interaction with models [41]	Done	

Anomaly and event list [44]	Partially Done	The Alerts/thresholds systems allows to set specific range of values above/below which an alert would be triggered.
Anomaly and event detection [45]	Done	
Homogenize data [61]	Done	
Catalogue of models [62]	Done	
Detection of anomalies [63]	Partially done	Anomalies can be detected by the end user, through the overview widgets and the Alerts/thresholds systems allows to set specific range of values above/below which an alert would be triggered.
Feedback [64]	Done	
Centralized user administration system [65]	Done	
Configurable Dashboard [66]	Done	
UI Notification System [67]	Done	
Port Operational KPI list [70]	Done	
Operational Interface [71]	Done	
Analyze historical data [81]	Done	
Support for manually provided data [86]	Done	
Discovery service for data [104]	Done	Not straightforward for the stakeholder. The user can visualise all data integrated in the platform by creating a custom View on the different IH ElasticSearch indices existing in the database.
Visualization of data [105]	Done	
Use-case specific functional requirements		
Integration with ThPA	Done	

information system [46]		
Support wind and weather sensors [47]	Done	
Support air quality sensors [48]	Done	
Support water quality sensors and data [49]	Done	
Support noise sensors and data [50]	Done	
Support real-time fuel consumption sensors [51]	Not Done	These sensors were not needed for the pilot objectives and were not acquired during the project.
Support real-time gate surveillance sensors [52]	Not Done	
Support wind and weather data provided by third party [53]	Done	Data used for air and noise dispersion calculations
Support air quality data provided by third party [54]	Done	
Support traffic data provided by third party [55]	Done	
Estimate air pollution impact of handling cargo [56]	Done	
Estimate noise pollution impact of handling cargo [57]	Done	
Estimate air pollution impact of bulk cargo operations [58]	Partially done	PAS model execution offers estimations of the air pollution impact of the operations defined via the forms and from vessel calls data.
Visualize the traffic status [106]	Done	
Visualize the pollution [107]	Done	

4.3.1.2. User stories completion

Table 53. User stories completion of the Port City Integration (THPA) Use Case

As a/an	I want to …	So that …	Completed (Yes/No)	Comments
Terminal Operator	To estimate the impact of the current inbound / outbound flow of trucks entering /exiting the port, considering the actual traffic in the nearby (city)	We can regulate the number of working entry/exit gates to optimize the inbound/outbound traffic without impacting too much in the city traffic (alleviate bottleneck).	Yes	
Environmental Manager	Estimate the air pollution impact of bulk cargo operations to the city due to specific/bad forecasted weather conditions, for the next day	We can make decisions to decrease the impact (sprinkling, reduce the number of operations, etc.)	Yes	
	Estimate the air pollution impact of handling cargo (loading / unloading) to the city due to specific/bad forecasted weather conditions, for the next day	I can have a clear picture of the quantity that adds to the pollution of the city	Yes	
	Estimate the air pollution impact of handling cargo (loading / unloading) to the city due to specific/bad forecasted weather conditions, for the next day	I can have a clear picture of the parts of the city that are affected by port operations (as result of air pollution dispersion models)	Yes	
	Estimate the amount of noise from operating machinery for handling cargo (loading / unloading) to the city	I can have a clear picture of the quantity that adds to the noise of the city	Yes	

	Estimate the amount of noise from operating machinery for handling cargo (loading / unloading) to the city	I can have a clear picture of the parts of the city that are affected by port operations (as result of air pollution dispersion models)	Yes	

4.3.1.3. Other data collection

Table 54. Other data collection for the Port City Integration (THPA) Use Case

Requested data	Value
Number of end-users that were planned to use the platform	6
Number of end-users that at least tried to use the platform	9
Number of end-users that are really using the platform	3
Number of sensors connected to the local IoT platform	3
Number of different sensors connected to the local IoT platform	2. No additional sensors, but 2 external information systems (Gates and wind sensors that are not actually connected to the platform).

4.3.2. Data analysis (THPA)

Extended from table 8 (Quality In Use Model evaluation criteria) of D8.1.

4.3.2.1. Results of the Quality in use model

Table 55. Results of the Quality in use model for the Port City Integration (THPA) Use Case

Sub-characteristic	KPIs	Old System	PIXEL platform
Effectiveness			
Effectiveness	% of completed user stories	-	●- 100%
	Output Quality	-	●- 92,38%
Efficiency			

Efficiency	Efficiency level (Uses the Number of end-users KPI)	-	🔴 - 27,26%
Satisfaction			
Usefulness	Usefulness level	-	🟢- 78,44%
	Perceived usefulness	-	🟢- 87,14%
Trust	Trust level	-	🟢- 92,86%
	System Serenity	-	🟢- 89,29%
Context coverage			
Context completeness	Completeness level	-	🟢- 98,21%
Flexibility	Flexibility level	-	🟢- 94,64%

We obtained 8 different answers to the questionnaires: all evaluating the PIXEL platform, composed of 2 for the primary users, 3 for the secondary users and 3 for the indirect users. Those answers were provided by 8 different people *Efficiency level* also shows that the platform has not been adopted by the port yet.

4.3.2.2. Results of the Data Quality Model

Table 56. Results of the Data quality model for the Port City Integration (THPA) Use Case

Sub-characteristic	KPIs	Calculation Type	
Information Accuracy			
Currentness	Timeliness	AIMQ questionnaire	🟢- 97.6%
Correctness	Free of errors	AIMQ questionnaire	🟢- 93.5%
Credibility	Believability	AIMQ questionnaire	🟢- 99%
Precision	Precision	AIMQ-like questionnaire	🟢- 94%
Traceability	Traceability	AIMQ-like questionnaire	🟢- 93.50%
Information Accessibility			
Accessibility	Accessibility	AIMQ questionnaire	🟢- 100%
Information Appropriateness			
Understandability	Understandability	AIMQ questionnaire	🟢- 96%
Value Added	Advantage	AIMQ-like questionnaire	🟢- 100%
	Relevancy	AIMQ questionnaire	🟢- 100%
Representational Adequacy	Concise representation	AIMQ questionnaire	🟢- 96.5%
	Interpretability	AIMQ questionnaire	🟢- 96.8%
Consistency	Consistent representation	AIMQ questionnaire	🟢- 98%
Completeness	Number of sensors / devices connected to the	Count the number of sensors connected to the local IoT platform.	3

	local IoT platform		
	Number of types of data (sensors) connected to the local IoT platform.	Count the number of different sensors connected to the local IoT platform.	2
	Completeness	AIMQ questionnaire	●- 98.3%
Efficiency			
Efficiency	Ease of Operation	AIMQ questionnaire	●- 100%
Availability			
Availability	Availability	AIMQ-like questionnaire (reworked from Security)	●- 98.8%
	Security	AIMQ questionnaire	●- 100%
Portability			
Portability	Portability level	AIMQ-like questionnaire	●- 60%

We obtained 8 different answers to the questionnaires:composed of 2 for the primary users, 3 for the secondary users and 3 for the indirect users. Those answers were provided by 8 different people.Even with the low number of sensors compared to other use cases, users feel confident while using the data of the platform. Only drawback is the *Portability level*, which shows that data would be easier to export to traditional tools.

Table 57. PSP/IQ Model quadrants for the Port City Integration (THPA) Use Case

PSP/IQ Quadrants	Global	PRIMARY	SECONDARY	INDIRECT
Sound information	97.6%	100%	-	96%
Useful information	97.6%	100%	-	96%
Dependable information	100%	100%	-	-
Usable information	100%	100%	-	-

4.4. Port City Integration Use Case - PPA

4.4.1. Data collection and user stories

4.4.1.1. Requirements completion

Table 58. Functional requirements of the Port City Integration (PPA) Use Case

Requirement	Status (Done / Not fully done / not done)	Comments
Common functional requirements		
Import historical data [36]	Done	Historical data for PAS have been imported.
Interaction with models [41]	Done	PAS interacts with models.
Anomaly and event list [44]	Partially done	Anomalies can be detected by the end user, through the overview widgets and the Alerts/thresholds systems allows to set specific range of values above/below which an alert would be triggered.
Anomaly and event detection [45]	Done	
Homogenize data [61]	Done	
Catalogue of models [62]	Done	
Detection of anomalies [63]	Not done	Same rationale as [44] and [45]
Feedback [64]	Partially done	Currently, the feedback is collected via JIRA or direct email to the platform development team, however, there is no in-app support for feedback. This is a Should-have requirement.
Centralized user administration system [65]	Done	
Configurable Dashboard [66]	Done	
UI Notification System [67]	Done	

Port Operational KPI list [70]	Done	Some elements are being provided in OT/KPIs (can be added).
Operational Interface [71]	Done	
Analyze historical data [81]	Done	
Support for manually provided data [86]	Done	
Discovery service for data [104]	Not fully done	There is not a catalogue to display all port's sources (at least not straightforward to the port). There is the possibility to observe the data sources by creating custom visualisations.
Visualization of data [105]	Done	
Use-case specific functional requirements		
Support air quality sensors [73]	Done	
Support water quality data [75]	Done	(Through MARPOL)
Integration with the PMIS SPARC N4 [76]	Not done	The priority of this requirement was reduced during the execution of the pilot.
Estimate air pollution impact of cruise and passengers ships related activities [78]	Done	
Measure real-time air pollution impact of cruise and passengers ships related activities [79]	Not done	The priority of this requirement was reduced during the execution of the pilot.
Support noise sensors and data [87]	Done	
Support pollution and traffic data provided by third party [88]	Done	
Measure real-time noise pollution impact of cargo ships related activities [89]	Partially done.	No real time, only daily measurement (per minute, daily average, etc…)

Estimate noise pollution impact of cargo ships related activities [90]	Partially done	Noise pollution model (static, steady-state) was provided but not directly relating (through the platform) with cargo ships' related activities. Only indirect interpretation can be achieved checking custom Views and PAS simulation.
Port-City road congestion forecasting [91]	Done	

4.4.1.2. User stories completion

Table 59. User stories completion of the Port City Integration (PPA) Use Case

As a/an	I want to …	So that …	Completable (Yes/No)	Comments
Environ mental operator	Estimate the influence of the air emissions related with the port activities (cruise and passenger terminals) at the city	The port will develop air pollution dissemination models based on evaluation of air quality measurements and meteorological data	Yes/No	Isolated/Steady state model is available. However, air quality sensor requests are not available for specific time periods and, according to the stakeholder, platform results could have been shown in tabular excel formats.
	Have predictions for the effect at the city (emissions alarm mapping) by the air emissions associated with the Cruise and passenger Terminal	The port will enhance further the Air Quality monitoring network by establishing an additional air quality monitoring station 24/7 in the Cruise Terminal.	Yes	The port has installed an additional air quality monitoring station 24/7 in the Cruise Terminal.
	Co-evaluation with results of LAeq indicator measurements in order to have adequate data for noise mapping and	Establishment of a permanent noise monitoring network with suitable sensors for 24/7 measurements of Lden indicator in	Yes	The port has installed a suitable noise sensor for 24/7 measurements of the Lden indicator in the Container Terminal area.

		the Container Terminal area		
	Accurate estimation of the noise level impact of the Container Terminal in the residential nearby area	Establishment of a permanent noise monitoring network with suitable sensors for 24/7 measurements of Lden indicator in the Container Terminal area	No	Noise dispersion is based on a third party model run by MEDRI. The model is steady state, therefore the port needs the support of technical partners of the project each time it wants to run the model

4.4.1.3. Other data collection

Table 60. Other data collection for the Port City Integration (PPA) Use Case

Requested data	Value
Number of end-users that were planned to use the platform	7
Number of end-users that at least tried to use the platform	7
Number of end-users that are really using the platform	3
Number of sensors connected to the local IoT platform	3
Number of different sensors connected to the local IoT platform	3

4.4.2. Data analysis (PPA)

Extended from table 8 (Quality In Use Model evaluation criteria) of D8.1.

4.4.2.1. Results of the Quality in use model

Table 61. Results of the Quality in use model for the Port City Integration (PPA) Use Case

Sub-characteristic	KPIs	Old System	PIXEL platform
Effectiveness			

Effectiveness	% of completed user stories	-	🟡- 50%
	Output Quality	53.97%	🟢- 82.54%
Efficiency			
Efficiency	Efficiency level (Uses the Number of end-users KPI)	-	🟡 - 28.40% (summer holidays and global delays on adopting the full platform)
Satisfaction			
Usefulness	Usefulness level	-	🟡- 73.33%
	Perceived usefulness	52.38%	🟢- 80.95%
Trust	Trust level	57.14%	🟢- 78.57%
	System Serenity	73.21%	🟢- 82.14%
Context coverage			
Context completeness	Completeness level	57.14%	🟢- 80.36%
Flexibility	Flexibility level	51.79%	🟢- 83.93%

We obtained 10 different answers to the questionnaires:

- 5 evaluating the old system, composed of 2 for the primary users, 2 for the secondary users and 1 for the indirect users. Those answers were provided by 5 different people.
- 5 evaluating the PIXEL platform, composed of 2 for the primary users, 2 for the secondary users and 1 for the indirect users. Those answers were provided by 5 different people.

The medium value for the *% of completed user stories* contrasts with the 100% requirement completion of the corresponding models of the use-cases, analysed in the above section 3 of this deliverable D8.3. Once again, the late deployment of the platform is responsible for the unyet adopted platform, and reflected in the *Efficiency level*. *Usefulness level* is discussed in below section 4.6.

4.4.2.2. Results of the Data Quality Model

Table 62. Results of the Data quality model for the Port City Integration (PPA) Use Case

Sub-characteristic	**KPIs**	**Calculation Type**	
Information Accuracy			
Currentness	Timeliness	AIMQ questionnaire	🟢- 79.30%
Correctness	Free of errors	AIMQ questionnaire	🟢- 96.70%
Credibility	Believability	AIMQ questionnaire	🟢- 96.70%
Precision	Precision	AIMQ-like questionnaire	🟢- 94.20%
Traceability	Traceability	AIMQ-like questionnaire	🟢- 99.20%
Information Accessibility			
Accessibility	Accessibility	AIMQ questionnaire	🟢- 98.80%
Information Appropriateness			
Understandability	Understandability	AIMQ questionnaire	🟢- 98.30%

Value Added	Advantage	AIMQ-like questionnaire	🟢- 98.30%
	Relevancy	AIMQ questionnaire	🟢- 96.70%
Representational Adequacy	Concise representation	AIMQ questionnaire	🟢- 90%
	Interpretability	AIMQ questionnaire	🟢- 92.70%
Consistency	Consistent representation	AIMQ questionnaire	🟢- 98.30%
Completeness	Number of sensors / devices connected to the local IoT platform	Count the number of sensors connected to the local IoT platform.	3
	Number of types of data (sensors) connected to the local IoT platform.	Count the number of different sensors connected to the local IoT platform.	3
	Completeness	AIMQ questionnaire	🟢- 97.80%
Efficiency			
Efficiency	Ease of Operation	AIMQ questionnaire	🟡- 64%
Availability			
Availability	Availability	AIMQ-like questionnaire (reworked from Security)	🟢- 93.80%
	Security	AIMQ questionnaire	🟢- 91.30%
Portability			
Portability	Portability level	AIMQ-like questionnaire	🟡 - 71.70%

We obtained 4 different answers to the questionnaires composed of 2 for the primary users, 1 for the secondary users and 1 for the indirect users. Those answers were provided by 4 different people.

It is the only use case for which the *Efficiency* KPI is medium. This is directly related to the associated model. Data extracts, reflected by the *Portability* KPI, have to be done by someone technical.

Table 63. PSP/IQ Model quadrants for the Port City Integration (PPA) Use Case

PSP/IQ Quadrants	Global	PRIMARY	SECONDARY	INDIRECT
Sound information	95.4%	94.7%		96.7%
Useful information	95.9%	97.2%		93.3%
Dependable information	91.3%	91.3%		
Usable information	81.4%	81.4%		

4.5. Port Environmental Index Use Case

4.5.1. Data collection

4.5.1.1. Requirements completion

Table 64. Functional requirements of the PEI Use Case

Requirement	**Status (Done / Not fully done / not DOne)**	**Comments**
Common functional requirements		
Import historical data [36]	Done	
Interaction with models [41]	Done	
Anomaly and event list [44]	Partially done	Anomalies can be detected by the end user, through the overview widgets and the Alerts/thresholds systems allows to set specific range of values above/below which an alert would be triggered.
Anomaly and event detection [45]	Done	
Homogenize data [61]	Done	
Catalogue of models [62]	Done	
Detection of anomalies [63]	Partially done	Same as above.
Feedback [64]	Done	
Centralized user administration system [65]	Done	
Configurable Dashboard [66]	Done	
UI Notification System [67]	Done	
Port Operational KPI list [70]	Done	
Operational Interface [71]	Done	
Analyze historical data [81]	Done	

Support for manually provided data [86]	Done	
Discovery service for data [104]	Done	
Visualization of data [105]	Done	
Use-case specific functional requirements		
PEI Evaluation [92]	Done	
PEI Data Sources [93]	Done	
PEI Notification [94]	Done	
PEI Dashboard [95]	Done	
PEI Dashboard - Time series [96]	Done	

4.5.2. Data analysis

Extended from table 8 (Quality In Use Model evaluation criteria) of D8.1.

4.5.2.1. Results of the Quality in use model

Table 65. Results of the Quality in use model for the PEI Use Case

Sub-characteristic	KPIs	Old System	PIXEL platform
Effectiveness			
Effectiveness	% of completed user stories	-	-
	Output Quality	-	● - 92.33%
Efficiency			
Efficiency	Efficiency level (Uses the Number of end-users KPI)	-	-
Satisfaction			
Usefulness	Usefulness level	-	● - 85.52%
	Perceived usefulness	-	● - 75.99%
Trust	Trust level	-	● - 91.07%
	System Serenity	-	● - 100%
Context coverage			
Context completeness	Completeness level	-	● - 97.77%
Flexibility	Flexibility level	-	● - 97.77%

We obtained 26 different answers to the questionnaires:

- composed of 8 for the primary users, 8 for the secondary users and 10 for the indirect users. Those answers were provided by 26 different people.

4.5.2.2. Results of the Data Quality Model

Table 66. Results of the Data quality model for the PEI Use Case

Sub-characteristic	KPIs	Calculation Type	
Information Accuracy			
Currentness	Timeliness	AIMQ questionnaire	●- 88.4%
Correctness	Free of errors	AIMQ questionnaire	●- 87.20%
Credibility	Believability	AIMQ questionnaire	●- 89%
Precision	Precision	AIMQ-like questionnaire	●- 85.70%
Traceability	Traceability	AIMQ-like questionnaire	●- 83.30%
Information Accessibility			
Accessibility	Accessibility	AIMQ questionnaire	●- 87.20%
Information Appropriateness			
Understandability	Understandability	AIMQ questionnaire	●- 77.30%
Value Added	Advantage	AIMQ-like questionnaire	●- 92.70%
	Relevancy	AIMQ questionnaire	●- 85.20%
Representational Adequacy	Concise representation	AIMQ questionnaire	●- 84.20%
	Interpretability	AIMQ questionnaire	●- 77.50%
Consistency	Consistent representation	AIMQ questionnaire	●- 90%
Completeness	Number of sensors / devices connected to the local IoT platform	Count the number of sensors connected to the local IoT platform.	-
	Number of types of data (sensors) connected to the local IoT platform.	Count the number of different sensors connected to the local IoT platform.	-
	Completeness	AIMQ questionnaire	●- 88.90%
Efficiency			
Efficiency	Ease of Operation	AIMQ questionnaire	●- 84.20%
Availability			
Availability	Availability	AIMQ-like questionnaire (reworked from Security)	●- 95.30%
	Security	AIMQ questionnaire	●- 97.80%

Portability			
Portability	Portability level	AIMQ-like questionnaire	🟡- 60.20%

We obtained 21 different answers to the questionnaires composed of 8 for the primary users, 6 for the secondary users and 7 for the indirect users. Those answers were provided by 21 different people.

Table 67. PSP/IQ Model quadrants for the PEI Use Case

PSP/IQ Quadrants	**Global**	**PRIMARY**	**SECONDARY**	**INDIRECT**
Sound information	91%	94%		87.6%
Useful information	80%	83.3%		76.2%
Dependable information	97.8%	97.8%		
Usable information	85.7%	85.7%		

4.6. All Use cases

4.6.1. Data collection and user stories

4.6.1.1. Non functional and legal requirements completion

Table 68. Non functional and legal requirements completion

Requirement	**Status (Done / Not fully done / not done)**	**Comments**
Non-functional requirements		
Compliance [38]	Done	PIXEL platform has a security layer with authentication and different types of roles are available. This was a “must have” requirement.
Multi Language support [43]	Done	English, Spanish, Italian and Greek are supported. This was a “must have” requirement.
Interoperability [59]	Done	PIXEL integrates information provided by each port. This was a “must have” requirement.
Scalability [60]	Done	The PIXEL architecture (with docker)

		allows it to be scalable in terms of IT resources. This was a “must have” requirement.
Security Communications between components [68]	Done	A specific security layer has been integrated in the PIXEL platform. This was a “must have” requirement.
Data source API documentation [83]	Partially done	Some data sources have been documented but not all. Some data sources have no API available. This was a “should have” requirement and is being completed during the last days of the project.
Data source API versioning [84]	Partially done	API versioning of data sources connected to the PIXEL platform has not been done. However, there is versioning for NGSI agents. This was a “should have " requirements.
Data source API connectivity [85]	Done	
Access security [97]	Done	
Availability [98]	Done	
Integrity [99]	Done	
Web UI [100]	Done	
Deployment environments [101]	Done	
Open Source licensing [102]	Done	
Portability [103]	Done	
Legal requirements		
Use Case Regulatory Context Compliance [118]	Done	

4.6.2. Data Analysis

4.6.2.1. Results of the Quality in use Model

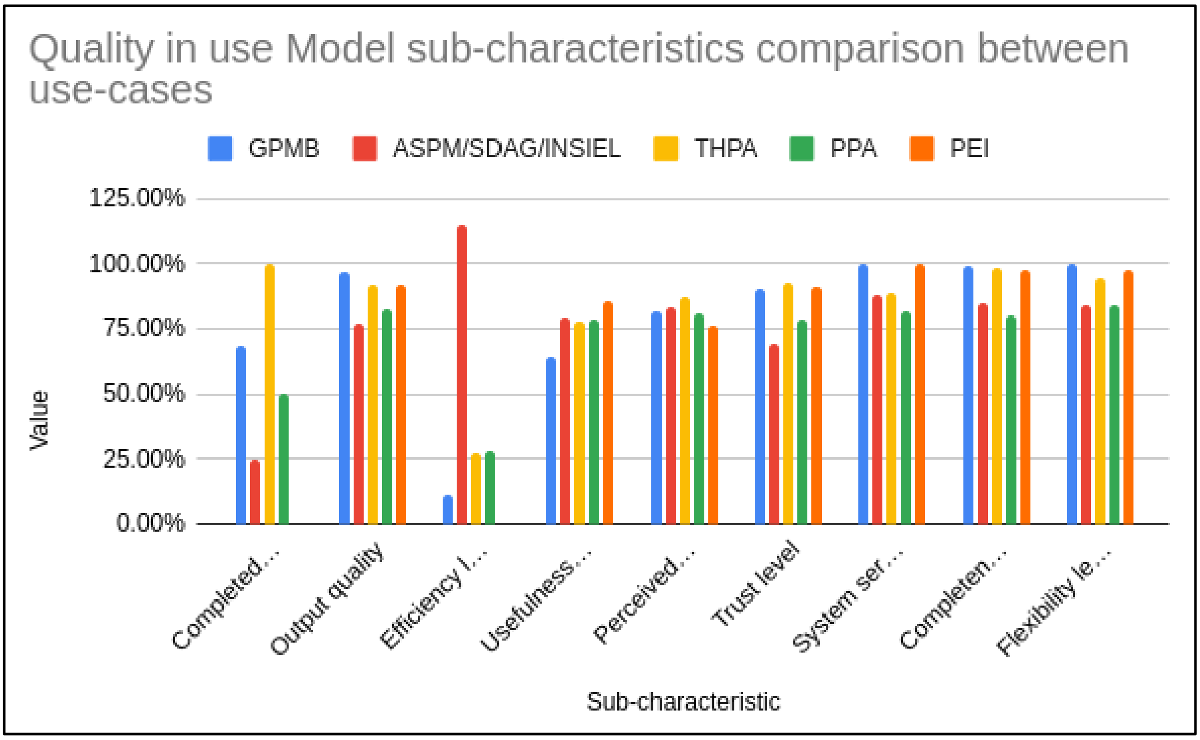

Figure 27. Quality in use Model sub-characteristics comparison between use-cases

Overall results for the Quality in use Model were very good, with most of the KPIs above 75%. As we stated per use-cases, most of the low-end results are coming from:

- The % of completed user stories, however most of them have been either compensated or reduced in priority during the execution of the pilots. In addition, compared with the requirements implemented by the platform and evaluated in section 3, most of the relevant requirements have been achieved. So this KPI can be explained by the "late" deployment of the PIXEL platform and the learning and adoption curve of innovative tools in ports. Indeed, as the platform is a professional tool, with a lot of different functionalities, it takes time to fully master it.
- However, even with that, we see that the platform is useful and complete, those KPIs being above 75%.
- The efficiency, for the same reason that it takes time for people to switch to the platform if they use to work with another software.

Best results are achieved for the PEI, as it is the tool that no ports had before the PIXEL platform.

4.6.2.2. Results of the Data Quality Model

We show below the KPIs results for all ports, allowing for an easy comparison.

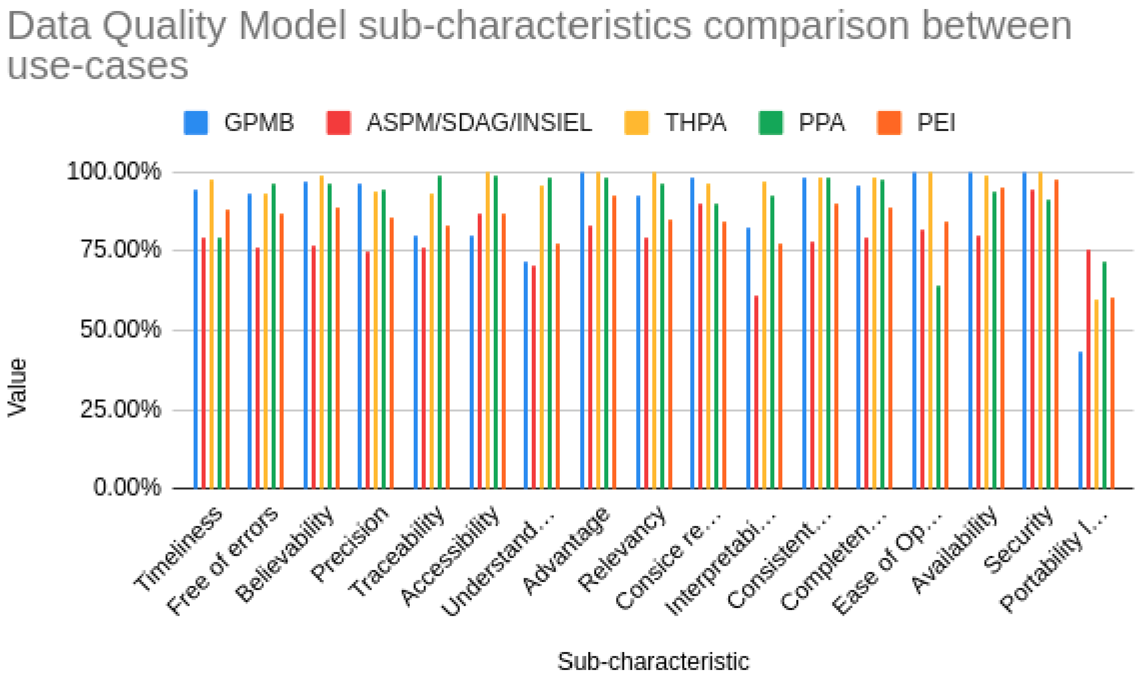

Figure 28. Data Quality Model sub-characteristics comparison between use-cases

Overall results for the Data Quality Model were very good, with most of the KPIs above 75%.

As we stated per use-cases, most of the improvable results are coming from:

- The portability, which is the drawback of having a platform with a lot of functionalities. It is needed to have someone technical to help the end-users exporting their results. However, this does not affect all the other KPIs, which shows that end-users are still confident with the platform data.

As for the Quality in use Model, best results are achieved for the PEI, as it is the tool that no ports had before the PIXEL platform.

D8.2 for the Data Qquality Model also introduced two analysis techniques used to identify IQ (Information Quality) problem areas, the IQ Benchmark Gap and the IQ Role Gap. Calculating the IQ Benchmark Gap per use-case, using the answers deciles, would not make sense because of the reduced number of answers. However, we introduce in this section the IQ Role Gap, comparing all use-cases between the Primary and the Indirect users.

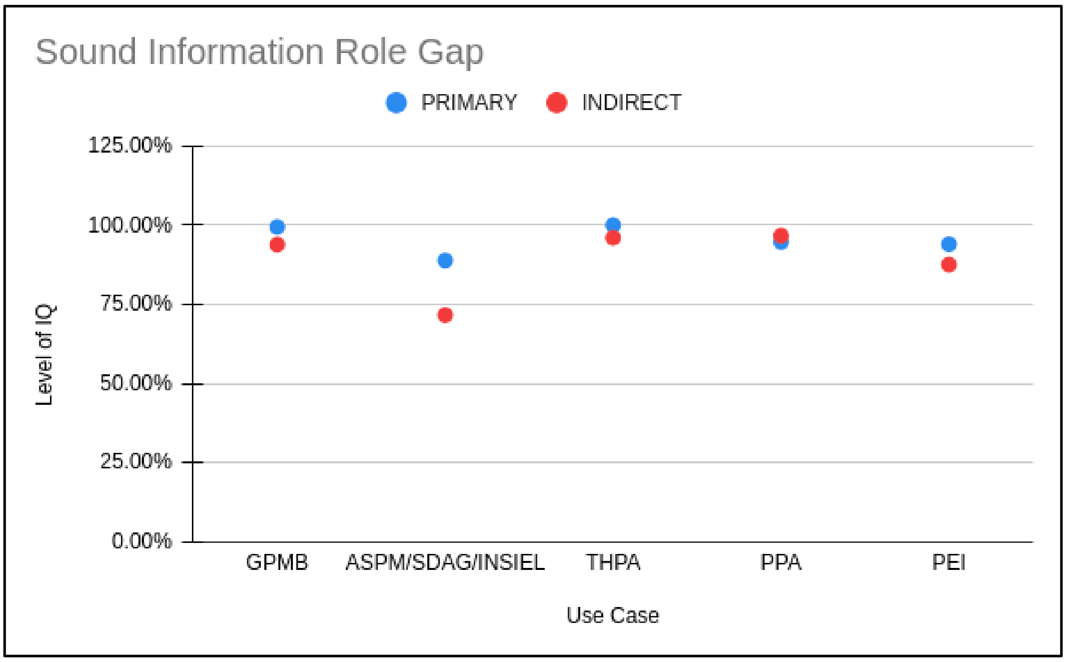

Figure 29. Sound Information Role Gap

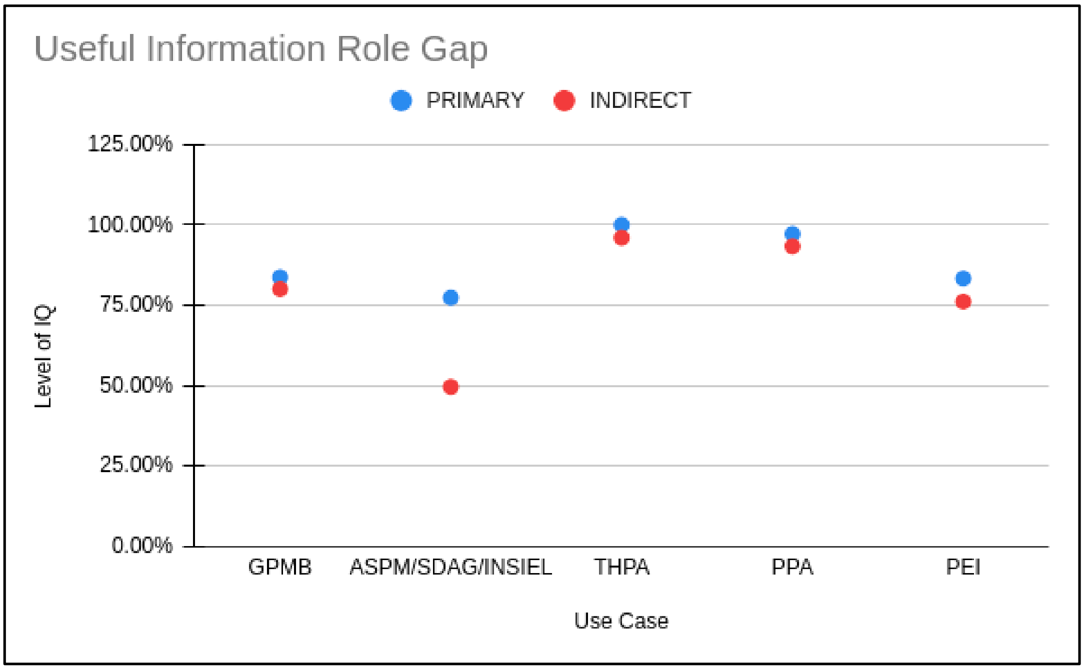

Figure 30. Useful Information Role Gap

IQ Role GAP aims to model the difference between the data needs of the consumer and the offer of the provider. In PIXEL, the provider can be identified as the PRIMARY user as he is the one that creates the added value of the data, and the customer can be identified as the INDIRECT user as he is the one consuming the data. What we see is that the PRIMARY user has better outcomes from the data produced by PIXEL than the INDIRECT user as he is the one responsible for it and as he has better control over it. We evaluate here only the PSP/IQ characteristics (derived by the quadrant in the above sections) that are common to those 2 users profiles. Evaluating the others wouldn't make sense as those 2 profiles do not have the complete same needs and expectations.

Conclusion

Following the work done in D8.2, this D8.3 deliverable achieved evaluating the PIXEL platform both regarding the technical performance and the user acceptance in the different use cases.

Overall results were very good, and reflects that the platform is both:

- Technically efficient, with computation needs that allow the different ports to host the platform on site;
- Able to answer the requirements of end-users;
- Comfortable for end-users;
- Computing data of high quality.

Although PIXEL platform has been developed and deployed following a specific structure, the different modules of the platform have been designed to handle adding new models. It is thus expected that many more use cases will be achievable and that, together, we are building the ports of the future.

Annexes

Annexe A1: Quality In Use Questionnaire

Anonymised responses of the questionnaires are available by following this link: https://docs.google.com/spreadsheets/d/1E2Evci82_KbZRV7gjuDS4N3qcRIwv2YIHQwxxnD-ohc/edit?usp=sharing

Questions of the questionnaires are provided below.

Quality In-Use questionnaire

* Required

1. What is your organisation ? *

Mark only one oval.

- GPMB
- ThPA
- PPA
- ASPM / SDAG

2. What is your name ? *

3. What is your function ? *

Port user's classification

Table 7: Port users' classification

Port	Primary Users	Secondary Users	Indirect Users
GPMB	Statistics Manager Energy Manager Port Manager	IT Manager Software Editor	Environmental Manager Port Agent/Operator
ASPM	Environmental Manager Parking area Manager	Software Editor	Gate/Access Manager
PPA	Environmental Manager Management team	IT Department	Quality Assessement
ThPA	Environmental Manager	IT Manager	Terminal Operator

4. What is your function category ? (Use the image above to help you) *

Mark only one oval.

- Primary User: a person who interacts with the system to achieve the primary goals.
- Secondary User: a person who provides support.
- Indirect User: a person who receives output but does not interact with the system.
- I don't know

5. Which use-case are you evaluating ? *

Mark only one oval.

- Energy Management
- Intermodal Transport
- Port City Integration
- Port Environmental Index (PEI)
- I don't know

Evaluation context

Please remember that you should fill in this questionnaire three times:
* Once evaluating your old system for your main use-case (i.e. Energy Management, Intermodal Transport or Port City Integration).
* Once evaluating the PIXEL platform for your main use-case (i.e. Energy Management, Intermodal Transport or Port City Integration).
* Once evaluating the PIXEL platform for the PEI use-case.

6. For this session, you are evaluating... *

Mark only one oval.

- Your old system for your main use-case (i.e. Energy Management, Intermodal Transport or Port City Integration)
- The PIXEL platform for your main use-case (i.e. Energy Management, Intermodal Transport or Port City Integration)
- The PIXEL platform for the PEI use-case

Effectiveness, Effectiveness

Accuracy and completeness with which users achieve specified goals.

Most of the answers allow selection on a 7-point Likert scale, where :
- 1 : Strongly disagree
- 2 : Moderately disagree
- 3 : Somewhat disagree
- 4 : Neutral (neither disagree nor agree)
- 5 : Somewhat agree
- 6 : Moderately agree
- 7 : Strongly agree

Output Quality

7. The quality of the output I get from the platform is high. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

8. I have no problem with the quality of the platform's output. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

9. I rate the results from the platform to be excellent. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

Satisfaction, Usefulness	Degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use. Most of the answers allow selection on a 7-point Likert scale, where : - 1 : Strongly disagree - 2 : Moderately disagree - 3 : Somewhat disagree - 4 : Neutral (neither disagree nor agree) - 5 : Somewhat agree - 6 : Moderately agree - 7 : Strongly agree

Perceived usefulness

10. Using the platform improves my performance in my job *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

11. Using the platform in my job increases my productivity. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

12. Using the platform enhances my effectiveness in my job. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

13. I find the platform to be useful in my job. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

Satisfaction, Trust

Degree to which a user or other stakeholder has confidence that a product or system will behave as intended.

Most of the answers allow selection on a 7-point Likert scale, where :
- 1 : Strongly disagree
- 2 : Moderately disagree
- 3 : Somewhat disagree
- 4 : Neutral (neither disagree nor agree)
- 5 : Somewhat agree
- 6 : Moderately agree
- 7 : Strongly agree

Trust level

14. The platform behaves as expected when I use it. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

System Anxiety

15. Using the platform does not scare me at all. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

16. Working with the platform makes me nervous. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

17. The platform makes me feel uncomfortable. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

18. The platform makes me feel uneasy. *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	Strongly agree

Context coverage, Context completeness

Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in all the specified contexts of use.

Most of the answers allow selection on a 7-point Likert scale, where :
- 1 : Strongly disagree
- 2 : Moderately disagree
- 3 : Somewhat disagree
- 4 : Neutral (neither disagree nor agree)
- 5 : Somewhat agree
- 6 : Moderately agree
- 7 : Strongly agree

Completeness level

19. I think that the platform, in all its specified contexts of use, is : *

Mark only one oval per row.

	Strongly disagree	Moderately disagree	Somewhat disagree	Neutral (neither disagree nor agree)	Somewhat agree	Moderately agree	Strongly agree
Effective	◯	◯	◯	◯	◯	◯	◯
Efficient	◯	◯	◯	◯	◯	◯	◯
Useful	◯	◯	◯	◯	◯	◯	◯
Trustworthy	◯	◯	◯	◯	◯	◯	◯

Context coverage, Flexibility

Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements.

Most of the answers allow selection on a 7-point Likert scale, where :
- 1 : Strongly disagree
- 2 : Moderately disagree
- 3 : Somewhat disagree
- 4 : Neutral (neither disagree nor agree)
- 5 : Somewhat agree
- 6 : Moderately agree
- 7 : Strongly agree

Flexibility level

20. I think that the platform, in contexts beyond those initially defined, is : *

Mark only one oval per row.

	Strongly disagree	Moderately disagree	Somewhat disagree	Neutral (neither disagree nor agree)	Somewhat agree	Moderately agree	Strongly agree
Effective	⬭	⬭	⬭	⬭	⬭	⬭	⬭
Efficient	⬭	⬭	⬭	⬭	⬭	⬭	⬭
Useful	⬭	⬭	⬭	⬭	⬭	⬭	⬭
Trustworthy	⬭	⬭	⬭	⬭	⬭	⬭	⬭

References

Most of the questions were taken and/or adapted from the TAM 3 questionnaire (Venkatesh, V. & Bala, H. (2008). TAM 3: Advancing the Technology Acceptance Model with a Focus on Interventions. Decision Sciences, 39, 273-315.)

Annexe A2: Data Quality Questionnaire

Anonymised responses of the questionnaires are available by following this link:

https://docs.google.com/spreadsheets/d/1z_hHZwqcVgJl81GAz5vbRAgHb63D8TIFXQt6cyW1R70/edit?usp=sharing

Questions of the questionnaires are provided below.

Data Quality questionnaire

Most of the questions are taken and/or adapted from the AIMQ questionnaire (Lee, Yang & M. Strong, Diane & Kahn, Beverly & Y. Wang, Richard. (2002). AIMQ: A methodology for information quality assessment. Information & Management. 40. 133-146. 10.1016/S0378-7206(02)00043-5.)

Most of the items are measured on a 0 to 10 scale where 0 is "not at all" and 10 is "completely".

* Required

1. What is your organisation ? *

Mark only one oval.

- GPMB
- ThPA
- PPA
- ASPM / SDAG

2. What is your name ? *

3. Which use-case are you evaluating ? *

Mark only one oval.

- Energy Management
- Intermodal Transport
- Port City Integration
- Port Environmental Index (PEI)

Function category

Every following sections will require an answer only if you are part of the targeted "function category". You can find your function category in the table below.

Table 7: Port users' classification

Port	Primary Users	Secondary Users	Indirect Users
GPMB	Statistics Manager Energy Manager Port Manager	IT Manager Software Editor	Environmental Manager Port Agent/Operator
ASPM	Environmental Manager Parking area Manager	Software Editor	Gate/Access Manager
PPA	Environmental Manager Management team	IT Department	Quality Assessement
ThPA	Environmental Manager	IT Manager	Terminal Operator

4. What is your function ? *

____

5. What is your function category ? *

Mark only one oval.

- ( ) Primary User: a person who interacts with the system to achieve the primary goals.
- ( ) Secondary User: a person who provides support.
- ( ) Indirect User: a person who receives output but does not interact with the system.
- ( ) I don't know

Information Accuracy, Currentness

The degree to which data has attributes that are of the right age in a specific context of use.

Timeliness

Primary User
Indirect User

6. The data is sufficiently current for our work.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	( )	( )	( )	( )	( )	( )	( )	( )	( )	( )	( )	Completely

7. The data is not sufficiently timely.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

8. The data is not sufficiently current for our work.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

9. The data is sufficiently timely.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

10. The data is sufficiently up-to-date for our work.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information Accuracy, Correctness

The extent to which information is reliable in the sense of being free of errors.

Free of errors
Primary User
Indirect User

11. The data is correct.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

12. The data is incorrect.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

13. The data is accurate.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

14. The data is reliable.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information Accuracy, Credibility	The degree to which data has attributes that are regarded as true and believable by users in a specific context of use. Credibility includes the concept of authenticity (the truthfulness of origins, attributions, commitments).

Believability

Primary User
Indirect User

15. The data is believable.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

16. The data is of doubtful credibility.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

17. The data is trustworthy.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

18. The data is credible.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

Information Accuracy, Precision

The degree to which data has attributes that are exact or that provide discrimination in a specific context of use.

Precision

Primary User
Indirect User

19. The data is precise.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

20. The data is imprecise.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

21. The data is rigorous.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

22. The data is marked by exactness and accuracy of expression or detail.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information Accuracy, Traceability

The degree to which data has attributes that provide an audit trail of access to the data and of any changes made to the data in a specific context of use.

Traceability

Primary User
Secondary User

23. I am able to get the origin of the data.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

24. I am not able to know the origin of the data.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

25. The data has an indication of its origin

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

26. I am able to know from which sensor the data is coming.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information Accessibility, Accessibility

The degree to which data can be accessed in a specific context of use, particularly by people who need supporting technology or special configuration because of some disability.

Accessibility

Primary User

27. The data is easily retrievable.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

28. The data is easily accessible.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

29. The data is easily obtainable.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

30. The data is quickly accessible when needed.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information Appropriateness, Understandability	The degree to which data has attributes that enable it to be read and interpreted by users, and are expressed in appropriate languages, symbols and units in a specific context of use.

Understandability

Primary User
Indirect User

31. The data is easy to understand.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

32. The meaning of this data is difficult to understand.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

33. The data is easy to comprehend.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

34. The meaning of the data is easy to understand.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information appropriateness, Value Added

The extent to which data or information are beneficial and provide advantages from their use

Advantage

Primary User
Indirect User

35. This data provides an advantage.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

36. This data provides a weakness.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

37. This data provides a convenience.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

38. This data put in a favorable or more favorable position.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Relevancy

Primary User
Indirect User

39. The data is useful to our work.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

40. The data is relevant to our work.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

41. The data is appropriate for our work.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

42. The data applicable to our work.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information Appropriateness, Representational Adequacy

The extent to which data or information is represented in a concise, flexible and organized way with due relevancy to the users' goals to help user to achieve their specified goals.

Concise representation

Primary User
Indirect User

43. The data is formatted compactly.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

44. The data is presented concisely.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

45. The data is presented in a compact form.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

46. The representation of this data is compact and concise.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

Interpretability

Primary User
Indirect User

47. It is easy to interpret what the data means.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

48. The data is difficult to interpret.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

49. It is difficult to interpret the data.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

50. The data is easily interpretable.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

51. The measurement units for the data are clear.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Information Appropriateness, Consistency

The degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use. It can be either or both among data regarding one entity and across similar data for comparable entities.

Consistent representation

Primary User
Indirect User

52. The data is consistently presented in the same format.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

53. The data is not represented consistently.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

54. The data is presented consistently.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

55. The data is represented in a consistent format.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

Completeness, Completeness	The degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use.

Completeness
Primary User
Indirect User

56. The data includes all necessary values.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

57. The data is incomplete.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

58. The data is complete.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

59. The data is sufficiently complete for our needs.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

60. The data covers the needs of our tasks.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

61. The data has sufficient breadth and depth for our task.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Efficiency, Efficiency

The degree to which data has attributes that can be processed and provide the expected levels of performance by using the appropriate amounts and types of resources in a specific context of use.

Ease of Operation

Primary User

62. The data is easy to manipulate to meet our needs.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

63. The data is easy to aggregate.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

64. The data is difficult to manipulate to meet our needs.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

65. The data is difficult to aggregate.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

66. The data is easy to combine with other data.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Availability, Availability

The degree to which data has attributes that enable it to be retrieved by authorized users and/or applications in a specific context of use.

Availability

Primary User

67. The data can be retrieved by authorized users.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

68. The data cannot be retrieved by authorized users.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all												Completely

69. The data can be exploited by authorized users.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all												Completely

70. The data can be retrieved by users with enough privileges.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all												Completely

Security

Primary User

71. The data is protected against unauthorized access.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all												Completely

72. The data is not protected with adequate security.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

73. Access to this information is sufficiently restricted.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

74. This information can only be accessed by people who should see it.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

Portability, Portability	The degree to which data has attributes that enable it to be installed, replaced or moved from one system to another preserving the existing quality in a specific context of use.

Portability level
Primary User
Secondary User

75. The data is easily transportable to another system.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely

76. The data can't be easily transported to another system.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

77. The data can be easily mooved to another system.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely

78. The data can easily be exported from the system and imported to another.

Mark only one oval.

	0	1	2	3	4	5	6	7	8	9	10	
Not at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely