RECLAIM

Refurbishment and re-manufacturing
of large industrial equipment

Machinery Operation Profiling

March 2022 - M30

AUTHORS :
OSCAR GARCIA PERALES (ICE)
OSVALDO DIAZ (ICE)
CESAR MARIN (ICE)

DATE :
30.09.2022

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Technical References

Project Acronym	RECLAIM
Project Title	RE-manufaCturing and Refurbishment LArge Industrial equipMent
Project Coordinator	HARMS & WENDE GMBH & CO KG
Project Duration	1/10/2019 - 30/09/2023

Deliverable No.	D3.2
Dissemination level [1]	PU
Work Package	WP3 – Digital Retrofitting for Advanced Predictive Maintenance
Task	T3.2 – Machinery Operation Profiling
Lead beneficiary	ICE
Contributing beneficiary(ies)	ICE, HWH, GORENJE, ZORLUTEKS, SUPSI, PODIUM, SCM GROUP, FLUCHOS, CTCR, UNI
Due date of deliverable	31/05/2020 (Advanced internal version) 30/09/2022 (Final version)
Actual submission date	15/08/2020 (Advanced internal version) 30/09/2021 (Advanced internal version 2) 30/09/2022 (Final version)

[1] PU = Public

PP = Restricted to other programme participants (including the Commission Services)

RE = Restricted to a group specified by the consortium (including the Commission Services)

CO = Confidential, only for members of the consortium (including the Commission Services)

Document history

V	Date	Beneficiary	Author	Updates
0.01	08/05/2020	ICE	Oscar Garcia Perales	Initial version
0.02	19/06/2020	HWH; GORENJE; ZORLUTEKS	Alexander Maisuradze; Majda Meža; Ceyda Akkoyun	Info from machines from HWH, GORENJE and Zorluteks
0.1	20/07/2020	ICE	Oscar Garcia Perales	Definition of Indexes

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

0.11	23/07/2020	SUPSI; PODIUM	Elias Montini, Andrea Barni	Info from machines from PODIUM (inputted by SUPSI)
0.12	24/07/2020	SCM Group	Enrico Callegati	Info from SCM Group
0.13	28/07/2020	FLUCHOS, CTCR;	Rodrigo Fernández, Saúl Íñiguez	Info from machines from FLUCHOS (inputted by CTCR)
0.2	31/07/2020	ICE	Oscar Garcia Perales, Chris Orton	Description and specification of DRyICE
0.25	12/08/2020	ICE	Oscar Garcia Perales	Internal version ready
0.30	30/09/2020	ICE	Oscar Garcia Perales	Minor adjustments to indexes
0.40	23/12/2020	ICE; UNI	Oscar Garcia Perales, Adriano Ferrara	Research on UNI standards
0.45	04/08/2021	GORENJE	Jiří Balhar	Info from machines from MORA (3rd Party of GORENJE)
0.50	30/09/2021	ICE	Oscar Garcia Perales, Chris Orton	Input regarding creation of indexes
0.60	30/06/2022	ICE	Osvaldo Diaz Orozco, Oscar Garcia Perales	Input regarding DRyICE Access and API documentation
0.9	16/09/2022	ICE	Osvaldo Diaz Orozco, Cesar Marin	Final version ready for internal review
1.0	29/09/2022	ICE	Cesar Marin	Final reviewed version to submit formally to the EC

Contributors list

Partner nº	Organisation name	Name	Country
1	Information Catalyst for Enterprise, Ltd	Oscar Garcia Perales, Chris Orton, Osvaldo Diaz, Cesar Marin	UK / Spain
2	Harms & Wende GmbH	Alexander Maisuradze	Germany
3	GORENJE	Majda Meža, Jiří Balhar	Slovenia
4	ZORLUTEKS	Ceyda Akkoyun	Turkey
5	CTCR	Rodrigo Fernández, Saúl Íñiguez	Spain
6	SUPSI	Elias Montini, Andrea Barni	Switzerland
7	SCM Group	Enrico Callegati	Italy
8	LINKS Foundation	Rosaria Rossini	Italy

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

9	UNI	Adriano Ferrara	Italy

Reviewers List

n°	Name	Surname	Beneficiary
1	Alexander	Maisuradze	HWH
2	Rosaria	Rossini	LINKS

Abbreviations and Acronyms

Abbreviation	
AM	Autonomous Maintenance
DOA	Description of Action
DSF	Decision Support Framework
DRyICE	Distributed data storage and exploratory analytics for enterprises
ERP	Enterprise Resource Planning
FPM	Frequent Pattern Mining
HMI	Human Machine Interface
IIoT	Industrial Internet of Things
IoT	Internet of Things
JSON	JavaScript Object Notation - file format
KPI	Key Performance Indicator
kA	Kiloamperes
Kbyte	Kilobytes
KPI	Key Performance Indicator
LCC	Life Cycle Cost
Min	Minutes
ML	Machine Learning
MPI	Manufacturing Performance Index
MRT	Mean Repair Time
MTBF	Mean-Time-Between-Failures
MTTF	Mean-Time-to-Failure
MTTR	Mean-Time-to-Repair
OEE	Overall Equipment Effectiveness
Pcs	Pieces
PHA	Physical Assets

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

PHM	Prognostic and Health Management
PLC	Programmable Logic Controller
SPC	Statistical Process Control
SW	Software
TPM	Total Productive Maintenance
UNI	Ente Nazionale Italiano di Unificazione
WI	Whiteness Index

Summary

The vision of RECLAIM is to demonstrate technologies and strategies to support a new paradigm for refurbishment and re-manufacturing of large industrial equipment in factories, paving the way to a circular economy. Its ultimate goal is to save valuable resources by reusing equipment instead of discarding them. RECLAIM will support legacy industrial infrastructures with advanced technological solutions with built-in capabilities for in-situ repair, self-assessment, and optimal re-use strategies. It will establish new concepts and strategies for repair and equipment upgrade and factory layouts' redesign to gain economic benefits to the manufacturing sector.

The technological core of RECLAIM is a novel Decision Support Framework (DSF) that guides the optimal refurbishment and re-manufacturing of electromechanical machines and robotics systems.

This task ensures that all necessary information and specifications for the industrial machines under examination is collected. For this purpose, three types of indexes will be elaborated for machinery: Health, Performance and Production indexes, which will be populated with historic and real-time data. The correlation of these indexes will provide the profile of the machinery. This profile will be reused in further tasks of the project.

The present report is the deliverable D3.2 of the project, RECLAIM's Machinery Operation Profiling. Its purpose is to fully describe the operational information that can be collected from the machinery by the RECLAIM framework.

The document is structured as follows:

- Section 1: Provide the current review of the State-of-the-Art for defining the indexes and how they are calculated according to such sources. In addition to these sources, several UNI standards have been checked.
- Section 2: Covers the definition of the three Indexes that the RECLAIM project is defining for creating the profile of the machines.
- Section 3: Lists the specifications gathered for each machinery to be used by the use case pilots that will be used for the calculation of the three indexes.
- Section 4: Describes the repository infrastructure, based on DRyICE tool, which complements the information provided in D2.3.

The present document is the 1st version, containing an initial definition, for internal use, of the indexes and the repository. It also includes a set of templates providing the specifications of the machinery that will be supported for the industrial pilots foreseen in RECLAIM.

Disclaimer

This publication reflects only the author's view. The Agency and the European Commission are not responsible for any use that may be made of the information it contains.

Table of Contents

Table of Contents

Table of Figures

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

1 Indexes: State of the Art

1.1 Introduction

Unscheduled downtimes of machinery and equipment reduce product and service turnaround times, leading to higher production costs and potential loss of revenue. IoT and the assessment of the machines and production lines increase the efficiency throughout the life cycle of a machine. An Industrial IoT (IIoT) platform – with condition monitoring features – helps protect against this unwanted downtime. RECLAIM, in general, and this task, in particular, will help manufacturers to turn to IoT-driven machine condition monitoring to identify equipment issues that can affect the quality of production and to replace and/or refurbish equipment before it gets worse. Machinery from the RECLAIM's pilots' partners will be needed to collect information and combine it in the three indexes that RECLAIM has committed to be calculate. The information and specifications most likely to be extracted from these machines range from machinery specifications and functionalities to sensor reading, performance and capacity. It may also include information on production history and maintenance data as well as estimated future changes in the production demand and machinery usage.

These indexes are RECLAIM's efforts to log the machine failures and monitor of more relevant parameters as it has been realized that there is a need for either an automatic system which predicts failures before they cause damage and breakdown or a fault tolerant system which is insensitive to failures [1].

These indexes are part of the Decision Support Framework (DSF) that RECLAIM is entitled to develop for the timely and accurate machine's health forecast. As such, there are three main challenges that need to be addressed during the decision-making process concerning the refurbishment of a machine. First, is the machine worth refurbishing? Second, what is the best time to perform refurbishment at the least cost? Finally, how should the machine be refurbished? To determine if a machine is worth refurbishing, RECLAIM's DSF implements well-established Prognostics and Health Management (PHM) techniques to estimate the Life Cycle Cost (LCC) associated with the refurbishment of the machine. And these techniques are built on top of the indexes that RECLAIM is providing, namely health, production, and performance.

The next sections provide the current review of the State-of-the-Art for defining these indexes and how they are calculated according to such sources. In addition to these sources, several standards have been checked which are also reported below.

1.2 Health index

Machine "health" monitoring is a term used for observation of machinery condition using sensors on either continuous basis (on-line condition monitoring) or off-line basis (off-line condition monitoring). Most of the machines or systems have tolerance on their operating parameters [2].

The health of a machine can be measured based on various parameters all related to the amount of time that the machinery and/or the equipment has been ready to be working or actively working.

The health index machine is directly related to the reduction (or prevention) of failures of machinery, production lines and/or equipment. RECLAIM partners understand also that the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

health-index relates to the quality of the product manufactured (low ratio of defects) and it is correlated with the outcome (product) of the machine.

In some cases, the health-index may also be related to the readiness to address potential problems, to increase reliability of the equipment and to rapid recover in case of failures. In this sense, it could also be understood that the index is related to anomaly detection not related to the quality of the product (e.g., classification of products A++, A+, B, C…).

In technical terms, the health-index will be related to the failure metrics such as Mean-Time-to-Failure (MTTF), Mean-Time-to-Repair (MTTR) or Mean-Time-Between-Failures (MTBF). In practical terms, any of these metrics will be defined in T2.5 (Holistic Life-Cycle Machinery models for facilitation Refurbishment & Re-manufacturing).

Informally speaking, the health-index is a categorized index calculated based on the information available and/or collected from the different machinery. The illustration below shows the level of health-index for a 'Failing Bearing Degradation Curve'.

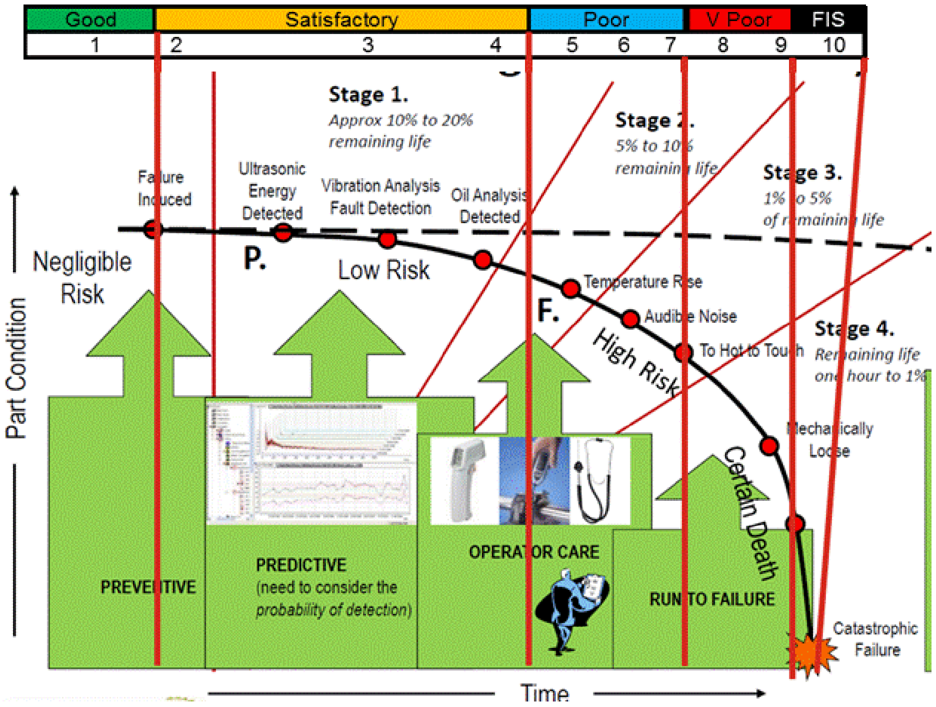


Figure 1: Health-index scale

In the figure, the health-index [7] is a holistic, fact-based, subjective, health assessment method using the simple linguistic terms good, satisfactory, poor, very poor and failed-in-service and the numbers 1 – 10, where 1= Good health (G), 2-4 = Satisfactory (S), 5-7 = Poor (P), 8-9 = Very Poor (VP), 10=Failed-in-Service (FIS).

The health-index recognises that there is a dependency hierarchy which extends from the simplest function level, such as dirty oil, to complete and catastrophic total plant loss. This progression to failure enables assessments to be carried out, and dependency links to be made, at and between sub-function, function, component, equipment, machine, subsystem, system, and process and risk levels.

The assessment of the variables to calculate the health-index is performed based on several physical tests such as vibration, acoustic emission, oil sampling, valve monitoring, and even the most common of all monitoring methods – tactile senses and intuition – as information sources.

The DOA defines the Machine Health Index as follows (N: number of components in machine; i: sensing line; W weight of component; X: the component level of sensing line):

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

$$\sum_{i=1}^{N} W_i * X_i$$

1.3 Performance index

Manufacturers have historically used a number of metrics to drive performance improvement, with Overall Equipment Effectiveness (OEE) being the most popular.

OEE is a foundational metric used for process analysis and root cause problem solving. It measures the percentage of the theoretical maximum productivity that a manufacturing process is achieving, with an equation multiplying quality by performance (speed) by availability (uptime).

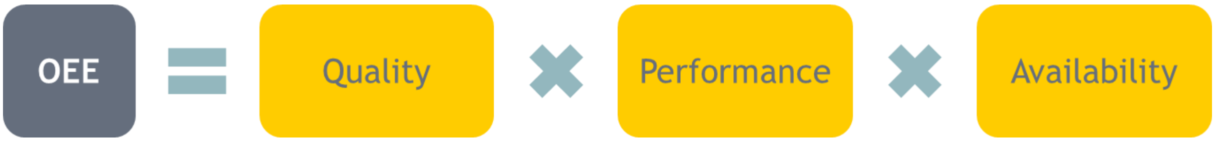


Figure 2: OEE calculation

The performance index machine is related to the throughput of the machine (e.g., products per minute, amount of debris produced, etc) depending to a certain extent on the quality of the product produced as well. In other words, how fast a machine performs a work and how good it is (velocity vs quality).

In this sense, RECLAIM partners understand also that the performance-index relates to the optimization of quality to number of pieces produced since intense production leads to reduction of reliability and wear. As such, this index should aim to identify and eliminate bottlenecks.

Sight Machine have been working in defining a new metric that extends beyond OEE’s areas of focus, which could be of use for RECLAIM, the Manufacturing Performance Index (MPI). MPI is an easy-to-understand metric that defines factory performance as a ratio of actual production achieved compared to the maximum production the factory is designed for. MPI is holistic in nature as it measures performance across the entire factory and can clearly link increases in performance to profitability.

In technical terms, the performance-index will be related to metrics such as actual production, maximum production capacity, products per batch (including the cycle time), how many production lines, amount of scrap, working hours and shifts, etc. In practical terms, these metrics will be defined as part of the pilots and in T2.5 (Holistic Life-Cycle Machinery models for facilitation Refurbishment & Re-manufacturing).

1.4 Production index

Like for Performance, manufacturers have historically used a number of metrics to derive production measurements; however, this measurement is not only bound to the Overall Equipment Effectiveness (OEE) like the Performance. Besides OEE, other KPIs can be used to assess the improvements made to OEE. The most typical one is the classical throughput, which can be defined as:

Throughput is the amount of a product or service that a company can produce and deliver to a client within a specified period of time.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

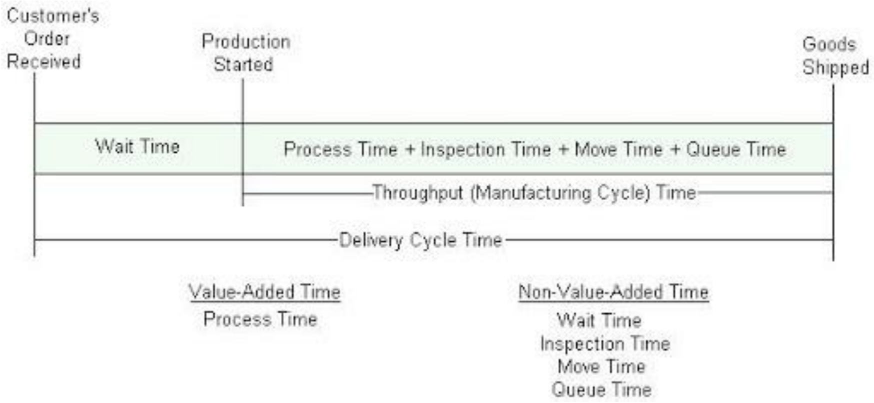


Figure 3: Relationship between throughput and other KPI-times

Other interesting KPIs can be Statistical Process Control (SPC), Total Production Maintenance (TPM), or Autonomous Maintenance (AM):

- Statistical Process Control (SPC):
  - Statistical method to monitor and control the production performance as well as continuously improve the quality of the product [3],
  - Its purpose is to improve the product quality, improve productivity, reduce wastes, reduce defects and improve customer values [4].
- Total Production Maintenance (TPM):
  - Aims to maximize the effectiveness of equipment throughout its entire life by the participation and motivation of the entire workforce [5],
  - The maintenance activities can be grouped into three categories which are reactive or corrective maintenance, preventive maintenance, and predictive maintenance [6].
- Autonomous Maintenance (AM):
  - It can be understood as the restoration and prevention of accelerated deterioration and has a major positive effect on OEE,
  - It is a step-by-step improvement process, rather than production teams taking on maintenance tasks [8].

In this sense, RECLAIM partners understand also that the production-index relates to the possibility of reusing or refurbishing an equipment which will lead to the implementation of easy-to-reuse mechanisms. This possibility will affect the production efficiency and a probable reduction of wear leading to a reduction of production costs. All these combined with the quality of the product manufactured (i.e., the less defects the better) and the level of usability of a production line/equipment will influence the OEE.

1.5 Standards Review

UNI is a standardisation body in Italy. Several of their standards are relevant to RECLAIM and thus they have been reviewed looking for definitions of the different indexes that are going to be used within RECLAIM. According to UNI rules, citing and reproducing their standards, it is necessary to request access in writing to such standards. A copy of this authorization can be seen in Annex A. The subsections below provide the outcomes after analysing these UNI norms.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

1.5.1 UNI CEN/TS 17385:2019 - Method for condition assessment of immobile constructed assets

This standard [9] deals with the methodologies of condition assessment for all types of structures by describing a method to assess the physical condition of all types of immobile constructed assets in a uniform and objective way. It describes how to achieve the condition class, based on non-destructive observation of defects of any asset or part thereof by using a predefined breakdown structure.

The methodology proposed by this document is based on the following variables:

- Defect severity, classified in Minor, Serious and Critical.
- Degradation level, classified in Level 1 (Low), Level 2 (Medium) and Level 3 (High).
- Extent, classified in Class 1 (Minimal), Class 2 (Insubstantial), Class 3 (Substantial), Class 4 (Significant), Class 5 (Widespread).
-

The description of these three variables is included in the table below:

Variable	Level	Description
Defect Severity	Related to its influence on the functioning of the element and the classified into three levels:	
	Minor	Does not affect the functionality of the element directly.
	Serious	Influences the secondary function of an element, which will lead to impact on the primary function.
	Critical	Influences the primary function of the element.
Degradation Level	It is expressed in terms of the visible detrimental change in physical condition, with time, use or external cause:	
	L1 (Low)	The degradation is hardly discernible / superficial.
	L2 (Medium)	The degradation is clearly discernible / significant.
	L3 (High)	The degradation is severe.
Extent	Assessed according to the percentage of the total area or volume of the element affected:	
	Class 1	Extent of defect is minimal, less than 2%.
	Class 2	Extent of defect is insubstantial, between 2% and 10%.
	Class 3	Extent of defect is substantial, between 10% and 30%.
	Class 4	Extent of defect is significant, between 30% and 70%.
	Class 5	Extent of defect is widespread, above 70%.

Table 1: Variables used for the assessment of the Condition Class of an Asset

Then, according to a combination of these three variables, an asset is classified from Condition Class 1 (Excellent Condition) to Condition Class 6 (Very Bad), according to Table 2 below.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Severity	Degradation Level	Degradation Extent				
		C1: Minimal	C2: Insubstantial	C3: Substantial	C4: Significant	C5: Widespread
Minor	L1: Low	1	1	1	1	2
	L2: Medium	1	1	1	2	3
	L3: High	1	1	2	3	4
Serious	L1: Low	1	1	1	2	3
	L2: Medium	1	1	2	3	4
	L3: High	1	2	3	4	5
Critical	L1: Low	1	1	2	3	4
	L2: Medium	1	2	3	4	5
	L3: High	2	3	4	5	6

Table 2: Condition class classification

The standard defines these six condition classes as follows:

- Condition Class 1 – Excellent condition, where no or very limited degradation has been identified.
- Condition Class 2 – Good condition, where an initial degradation is observed.
- Condition Class 3 – Fair condition, where the degradation is identifiable in places.
- Condition Class 4 – Poor condition, where the degradation is widespread and where reliability is compromised since a number of (severe) defects can lead to a loss of function.
- Condition Class 5 – Bad condition, where the degradation is serious, and components have significant defects in finish and function and reliability has reached a critical stage.
- Condition Class 6 – Very Bad condition, where the degradation is advanced and affects nearly all elements.

However, this classification can only be considered for elements with a single defect type. Where exist elements, or assets, with more than one single defect type the procedure stated below shall be applied:

- All individual defects are in the same section of the element, then the condition class is determined by the defect that leads to the highest class.
- All individual defects are of the same severity and degradation level BUT are in different sections of the element, then the condition class is determined by adding the extents of all the individual defects and proceed as usual calculations.
- All individual defects are of different severity or degradation levels and are in different sections, then the element shall be divided into sections by having one defect per section and one section without defects. Each "defective" section is given a 5 in the extent and the "non-defective" section is given a 1. Then, the methodology of Annex B of this standard must be applied for calculating the condition class of the element.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

1.5.2 UNI EN 15341:2019 - Maintenance - Maintenance Key Performance Indicators

This standard [10] lists Key Performance Indicators (KPIs) of the Maintenance Function and gives guidelines to define a set of suitable indicators, to appraise and to improve effectiveness, efficiency, and sustainability in the maintenance of the existing physical assets either industrial, infrastructures, facilities, civil buildings, or transportation systems, etc. in the framework of the external and internal influencing factors.

Within this document, several KPIs are worth to be considered and brought into the technical discussions of calculating the RECLAIM Indexes. In this sense, the following KPIs are the ones selected to be further investigated:

ID	KPI	Factors
Physical Assets Management related KPIs		
PHA4	Utilization rate of production capacity (%)	Actual Production Output
		Standard Production Capacity
PHA6	Total equipment effectiveness[1] R1xR2xR3	R1 – Maintenance Effectiveness (%)
		R2 – Manufacturing Effectiveness (%)
		R3 – Quality Effectiveness (%)
PHA8	Operational availability due to maintenance	Total Operating Time
		Total Operating time + Downtime
PHA15	Impact of maintenance on standard technical output (output units)	Annual Standard Technical Output
PHA19	Proportion of maintenance time on annual planned time	Down time due to maintenance works
		Total annual planned time
Maintenance Engineering related KPIs		
E5	MTBF: Meantime Between Failures (hours)	Total operating time
		Number of failures
E6	MRT: Mean Repair Time (%)	Total time to repair
		Number of failures
E8	Rate of failures (#/Year)	Annual number of failures
		Annual operating time
E9	Down time due to corrective maintenance (%)	Down time due to corrective maintenance
		Total down time due to maintenance reasons
E10	Down time due to condition-based maintenance (%)	Down time due to condition-based maintenance
		Total down time due to maintenance reasons

[1] Named also as Overall Equipment Effectiveness (OEE)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

E11	Down time due to predetermined maintenance (%)	Down time due to predetermined maintenance
		Total down time due to maintenance reasons
E12	Down time due to preventive maintenance (%)	Down time due to preventive maintenance
		Total down time due to maintenance reasons

Table 3: KPIs defined in UNI EN 15341:2019 useful for the definition of the RECLAIM Indexes

1.5.3 UNI EN 17007:2018 - Maintenance process and associated indicators

This European Standard provides a generic description of the maintenance process [11]. It specifies the characteristics of all the processes, parts of maintenance process, and establishes a maintenance model to gives guidelines for defining indicators. The purpose of the breakdown into processes and the representation of their inter-relationships is to help maintenance personnel, and particularly management at different levels, to:

- Clearly identify the actions to be taken to meet the overall objectives set by Management in terms of maintenance.
- Delegate responsibilities that ensure the realization of the actions with the required performance levels.
- For each process, clearly determine:
    - the necessary inputs and their origin,
    - the required results and their intended uses.
- Monitor and quantitatively assess the performance obtained at various levels of the breakdown into processes.
- Improve the collection and the distribution of data.
-

After checking UNI EN 17007:2018 standard it is clear that this document is devoted to the definition of the different maintenance processes and their breakdown instead of defining the different KPIs that may be checked and/or calculated to either start a maintenance process or to schedule a preventative maintenance in the near future of the machine.

2 Machinery Specifications

Pilot partners of RECLAIM have initially identified the machinery that will be supported for the industrial pilots foreseen in RECLAIM. For the benefit of the generation and calculation of the indexes, a template specifying these machines and the data that can be obtained from these is provided. These details are given in the following sections.

2.1 Machinery Overview

The following table provides an overview of the different machinery to be used and piloted during the RECLAIM project aggregated by pilot owner. A detailed description of each machinery is given in the sections below.

No.	Machinery	Responsible
1	MACH.HWH.01.RSM401	Harms & Wende
2	MACH.GORENJE.01.CELL_A	Gorenje
3	MACH.GORENJE.02.CELL_B_C	Gorenje
4	MACH.GORENJE.03.CELL_OBC	Gorenje
5	MACH.GORENJE.04.CELL_D_E	Gorenje
6	MACH.FLUCHOS.01.CZ/M	Fluchos
7	MACH.ZORLUTEKS.01.BLEACHING_MACHINE	Zorluteks

Table 4: Overview of the machinery to be used in RECLAIM

For collecting this information, the following template has been prepared aiming not only at the different user partners who should be providing the information about their machines, but also to the technical partners willing to collect, analyse and reuse the information shared.

MACH.PILOT_PARTNER.01.MACHINE_NAME	
Machine Identification	
Model	*Model of the machine*
Manufacturer	*Manufacturer of the Machine*
Activity	*Activity performed by the machine*
General Data	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Age	*Age of the machine*	
Workload	*Normal workload of the machine, measured in hours, days, or cycles*	
Usage	*Usage of the machine*	
Maintenance	*Which type of maintenance has been performed (ID, associated component and duration, category of the maintenance, worker, etc.)*	
Indexes		
Health	*Formula/Expression used to calculate the health of the machine*	
Performance	*Formula/Expression used to calculate the performance of the machine*	
Production	*Formula/Expression used to calculate the production of the machine*	
List of Errors		
Type of error and when it occurred	*Type of error and when it occurred*	
Metrics		
MTTF	*Mean Time To Failure internal formula*	
MTTR	*Mean Time To Repair internal formula*	
MTBF	*Mean Time Between Failures internal formula*	
Max Production	*Maximum Production per year*	
Max Capacity	*Maximum Capacity per year*	
Hours down (planned)	*Number of planned hours that the machine has been down*	
Hours down (unplanned)	*Number of unplanned hours that the machine has been down*	
<Other Metrics>	*Other metrics may be added as needed*	
Parameters		
<Param 1>	Type of param	*Type of parameter, meaning Integer, String, Boolean, etc.*
	Frequency of update	*Which is the frequency of the update related to the parameter*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	Amount of data	*Amount of data related to the parameter*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
<Param N>	Type of param	*Type of parameter, meaning Integer, String, Boolean, etc.*
	Frequency of update	*Which is the frequency of the update related to the parameter*
	Amount of data	*Amount of data related to the parameter*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
Processes		
<Process 1>	Type of process	*Identify the type of process*
	List of parameters affected	*List the parameters from the section above that are affected by this process*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
<Process N>	Type of process	*Identify the type of process*
	List of parameters affected	*List the parameters from the section above that are affected by this process*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
Data sources		
<Data Source 1>	Type of data	*Identify the type of data source*
	Frequency of update	*Which is the frequency of the update related to the data source*
	Amount of data	*Amount of data related to the data source*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
<Data Source N>	Type of data	*Identify the type of data source*
	Frequency of update	*Which is the frequency of the update related to the data source*
	Amount of data	*Amount of data related to the data source*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*

2.2 Harms & Wende

MACH.HWH.01.RSM401	
Machine Identification	
Model	RSM401
Manufacturer	Harms & Wende
Activity	Friction welding of small parts aviation industry
General Data	
Age	11 Years (constructed in 2008)
Workload	About 100 pieces in an hour
Usage	“Lufthansa” machine
Maintenance	*Every week: Visual test for damage and wear, test for hydraulic leaks, clean sample-holder and vacuum, check cooling system, check welding frequency at the end of maintenance.* *Every 50’000 weldings: Replace optical sensor* *Every 100’000 weldings: Replace sample-holder*
Indexes	
Health	1 Repair in a year.
Performance	
Production	See Workload above in “General Data”
List of Errors	
There exists no automatic log-book for machine errors	*About once in a year welding head (motor+spindle) is damaged*
Metrics	
MTTF	Every 50000 weldings (500 hours at 100 pieces an hour)
MTTR	Defect piece replacement time 0.5 - 4 hours (if new piece is available) Defect piece repair time: 3 Days. Order of new piece: 3-5 Days.
MTBF	500 – 600 hours (see details above)
Max Production	1 welding in 4 seconds
Max Capacity	2 million weldings
Hours down (planned)	Every 50000 weldings sample detection optical sensor should be replaced. Replacement time: 0.5 hour. Every 100’000 weldings sample holding piece should be replaced. Replacement time: 10 min.

Hours down (unplanned)	3-5 Days	
Parameters		
<Param 1>	Type of param	Flowmeter
	Frequency of update	Not logged, activates alarm at certain threshold
	Amount of data	
	<Other metadata worth to mention>	
<Param 2>	Type of param	Temperature of cooling water
	Frequency of update	Not logged, activates alarm at certain threshold
	Amount of data	
	<Other metadata worth to mention>	
<Param 3>	Type of param	Optical sample detector
	Frequency of update	Not logged, activates alarm at certain threshold
	Amount of data	
	<Other metadata worth to mention>	
<Param 4>	Type of param	Pressure detector
	Frequency of update	Not logged, activates alarm at certain threshold
	Amount of data	
	<Other metadata worth to mention>	
<Param 5>	Type of param	Angular velocity of motor during welding (curve)
	Frequency of update	Every welding
	Amount of data	Up to 20 KByte (2 – 20 KByte)
	<Other metadata worth to mention>	
<Param 6>	Type of param	Axial force on sample during welding (curve)
	Frequency of update	Every welding
	Amount of data	Up to 20 KByte (2 – 20 KByte)
	<Other metadata worth to mention>	
<Param 7>	Type of param	Mechanical motor torque during welding (curve)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	Frequency of update	Every welding
	Amount of data	Up to 20 KByte (2 - 20 KByte)
	<Other metadata worth to mention>	
<Param 8>	Type of param	Axial piece displacement during welding (curve)
	Frequency of update	Every welding
	Amount of data	Up to 20 KByte (2 - 20 KByte)
	<Other metadata worth to mention>	
<Param N>	Type of param	
	Frequency of update	
	Amount of data	
	<Other metadata worth to mention>	
Processes		
<Process 1>	Type of process	Welding
	List of parameters affected	Param-1 to Param-8. All parameters are recorded during single welding operation and packed into a single process dataset.
	<Other metadata worth to mention>	
<Process N>	Type of process	
	List of parameters affected	
	<Other metadata worth to mention>	
Data sources		
<Data Source 1>	Type of data	Internal sensors described in section "Data Parameters"
	Frequency of update	
	Amount of data	
	<Other metadata worth to mention>	
<Data Source N>	Type of data	
	Frequency of update	
	Amount of data	
	<Other metadata worth to mention>	

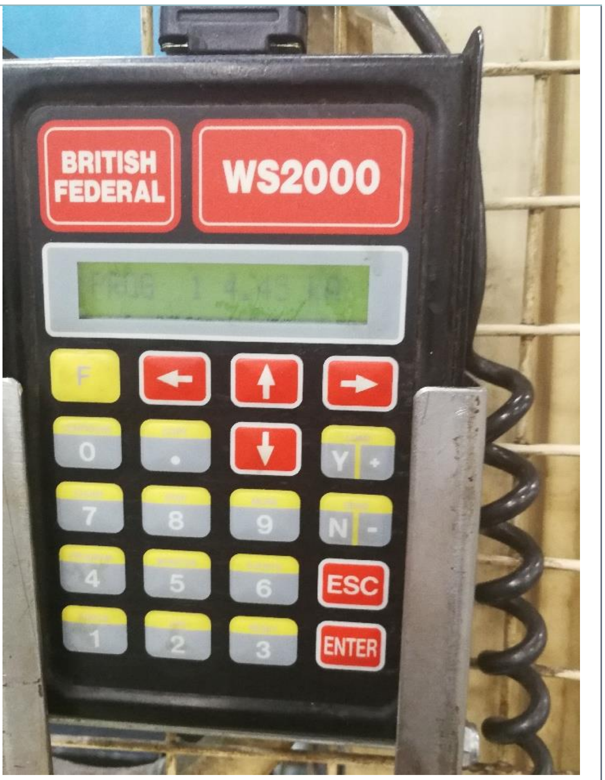
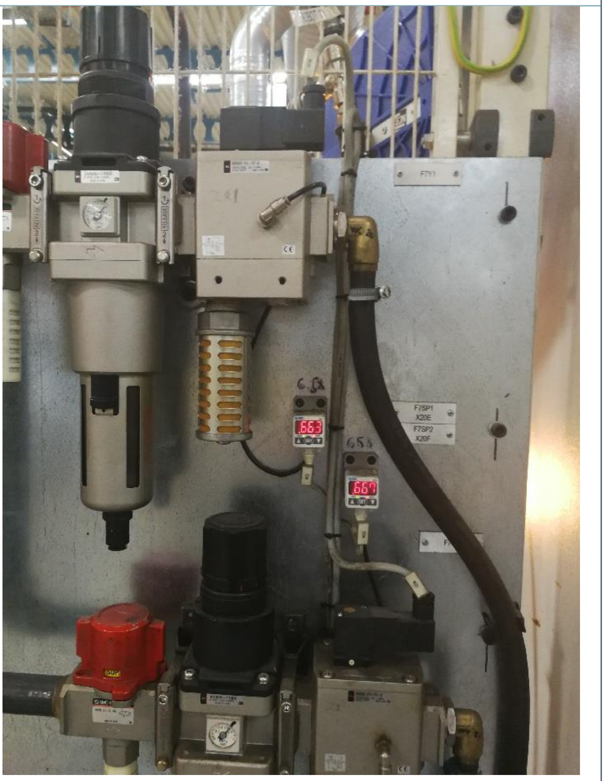
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

2.3 Gorenje

MACH.GORENJE.01.CELL_A	
Machine Identification	
Model	**Cell A**: feeder, 4 robots, rotary table, spot welding machine, 2 punching machines, 2 presses
Manufacturer	4 robots: KUKA (1), ABB (3) 1 spot welding machine: British Federal/Veldstar 2 presses: Meccanica Patron NB: In the Cell A are more machines and equipment - different manufacturers, but they are all working in cell A. Robots, spot welding machine and presses are most important (parameters).
Activity	The basic element for making the dishwasher is a flat sheet of stainless steel, called L and U sheet, on which various small parts are installed, further welded to the necessary semi-finished products, then the necessary holes are cut. Further the U and L plate are bent into the corresponding U and L shape.
General Data	
Age	Cell A: 1993 Robots: 2003, 2008, 2008, 2015
Workload	2-3 shifts
Usage	For manufacturing DW tubs
Maintenance	Preventive maintenance (monthly), 2 times/year main service
Indexes	
Health	rated capacity: 430 pcs/shift; actively working capacity: 380 pcs/shift Index= 88,4%
Performance	actively working capacity: 380 pcs/shift; scap: 3 pcs/shift Index=0,8%
Production	4 (on the scale 1-5); after refurbishment we expect better health and performance, evaluated from 1 to 5
List of Errors	
Mechanical, electrical, and SW failures	*Stop of machine because of wrong position, fuse control, overload motor control…; all failures are shown on HMI.*
Metrics	
MTTF	Nr. of operating hours / Nr. of produced units. Average value for 2019: 0,017
MTTR	Internally calculated by SAP ERP. Average value for 2019: 105 min

MTBF	Internally calculated by SAP ERP. Average value for 2019: 9,7 working days	
Max Production	380 pcs/shift	
Max Capacity	410 pcs/shift	
Hours down (planned)	50 min/shift	
Hours down (unplanned)	15 min/shift	
Parameters		
Pressure of hydraulic oil	Type of param	Mechanical / manometer
	Frequency of update	For every/last part
	Amount of data	No data
	Picture	
Welding current	Type of param	Sequence
	Frequency of update	For every part
	Amount of data	On switch (digital information)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

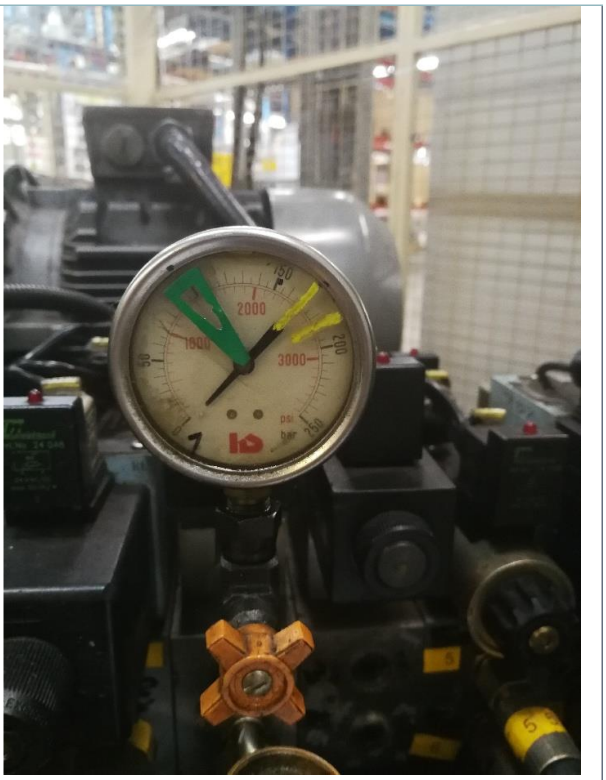
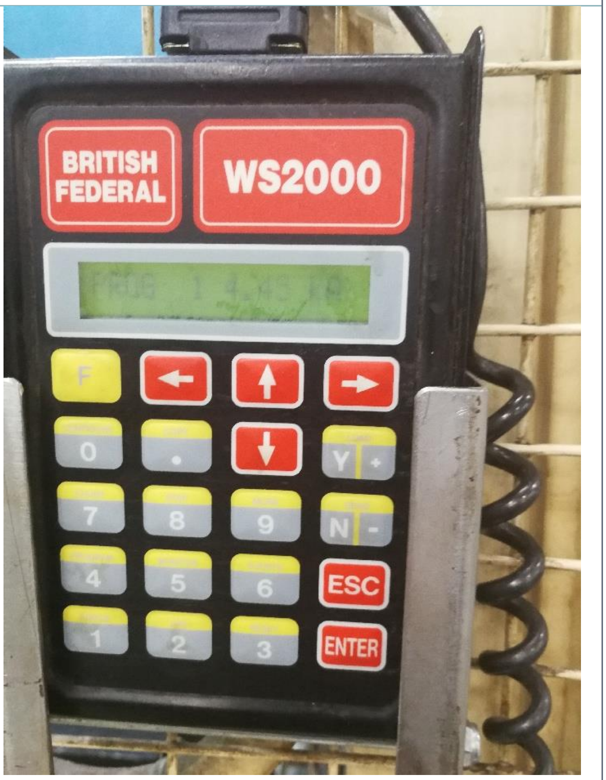
	Picture	
Air pressure	Type of param	Sequence
	Frequency of update	For every part
	Amount of data	On display (for information)
	Picture	
Processes		
Hole punching	Type of process	Punching (tool)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

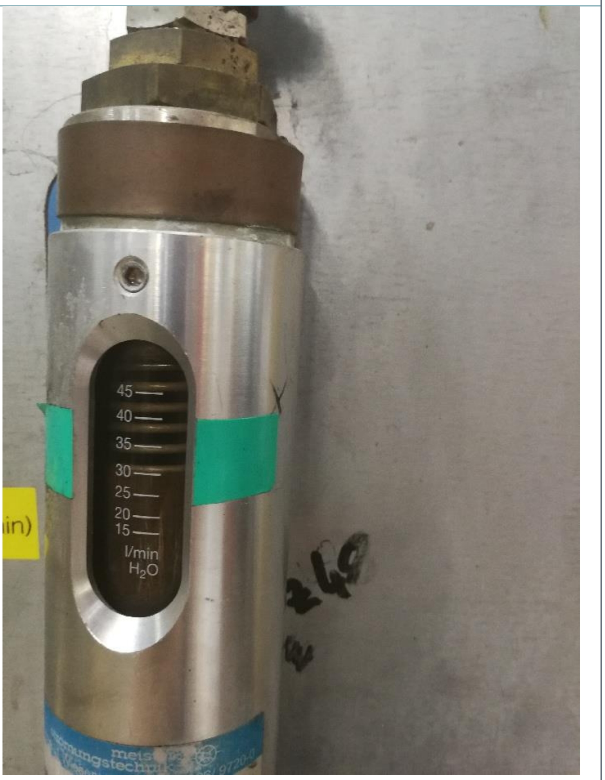
	List of parameters affected	None, just 'confirmation of operation done'
	<Other metadata worth to mention>	
Welding with robots	Type of process	Spot welding
	List of parameters affected	Welding current Air pressure Pressure of hydraulic oil
	<Other metadata worth to mention>	
Bending	Type of process	Bending tool, press
	List of parameters affected	Pressure of hydraulic oil
	<Other metadata worth to mention>	
Data sources		
Current loop	Type of data	kA
	Frequency of update	For every/last piece
	Amount of data	On internal instrument
	<Other metadata worth to mention>	
Manometer	Type of data	bar
	Frequency of update	Constant data
	Amount of data	On internal mechanical manometer
	<Other metadata worth to mention>	For air pressure For pressure of hydraulic oil

MACH.GORENJE.02.CELL_B_C	
Machine Identification	
Model	**Cell B, C**: 6 robots, rotary table, 5 welding machines (spot and seam welding), 2 double bending machines, cooling water system, transport rollers
Manufacturer	Robots: ABB 5 welding machines: British Federal/Veldstar 2 bending machines: Meccanica Patron
Activity	In the B-C robotic cells L and U dishwashers' plates are joined.

	The basic elements for the production of dishwasher tubs are the flat stainless steel sheets/plates of two dimensions (L and U), which previously came out of the A-cell and are manually loaded by the worker onto a 3-station turning table. Joining is done by bending, spot and seam welding. Additional punching is carried out if necessary.	
General Data		
Age	Cell B, C: 1993 Robots: 2007-2009	
Workload	2-3 shifts	
Usage	For manufacturing DW tubs	
Maintenance	*Preventive maintenance (monthly), 2 times/year main service*	
Indexes		
Health	rated capacity: 430 pcs/shift; actively working capacity: 380 pcs/shift Index= 88,4%	
Performance	actively working capacity: 380 pcs/shift; scrap: 3 pcs/shift Index=0,8%	
Production	4 (on the scale 1-5); after refurbishment we expect better health and performance, evaluated from 1 to 5	
List of Errors		
Mechanical, electrical and SW failures	*Stop of machine because of wrong position, fuse control, overload motor control…; all failures are shown on HMI.*	
Metrics		
MTTF	0,017 – same formula as in the previous machine applies	
MTTR	67 min – same formula as in the previous machine applies	
MTBF	8,3 days – same formula as in the previous machine applies	
Max Production	380 pcs/shift	
Max Capacity	410 pcs/shift	
Hours down (planned)	50 min/shift	
Hours down (unplanned)	15 min/shift	
Parameters		
Pressure of hydraulic oil	Type of param	Mechanical / manometer
	Frequency of update	For every/last part
	Amount of data	No data

	Picture	
Welding current	Type of param	Sequence
	Frequency of update	For every part
	Amount of data	On switch (digital information)
	Picture	 BRITISH FEDERAL WS2000 F 0 7 8 9 4 5 6 1 2 3 Y N ESC ENTER
Air pressure	Type of param	Sequence
	Frequency of update	For every part
	Amount of data	On display (for information)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	Picture	
Water flow	Type of param	
	Frequency of update	Constant
	Amount of data	Mechanical instrument
	Picture	
Water Temperature	Type of param	
	Frequency of update	
	Amount of data	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

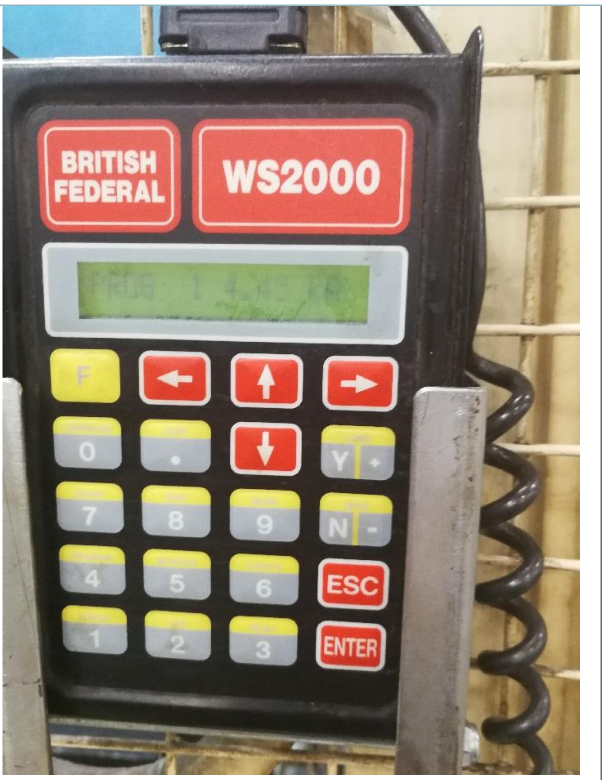
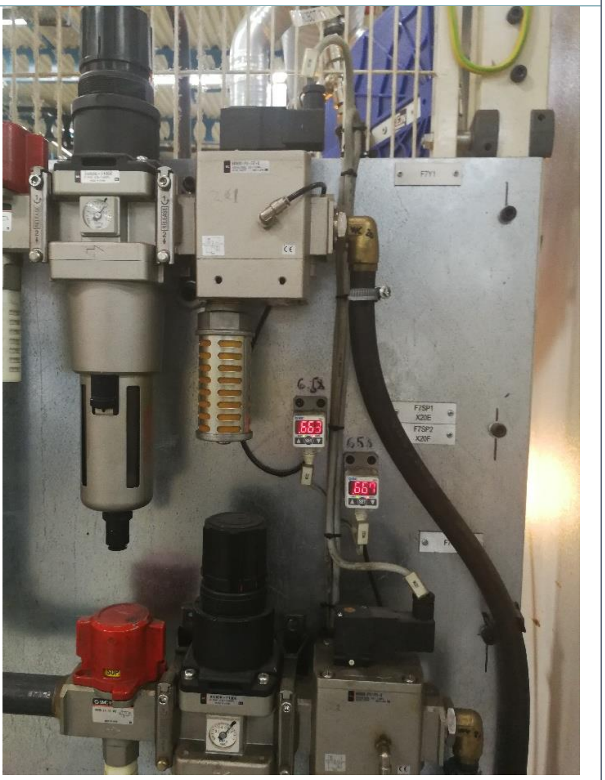
	Picture	
Processes		
Welding	Type of process	Spot welding, seam welding, robots for welding, robots for manipulation
	List of parameters affected	Water temperature Water flow Welding current Air pressure
	<Other metadata worth to mention>	
Double bending	Type of process	Double bending with press
	List of parameters affected	Air pressure Pressure of hydraulic oil
	<Other metadata worth to mention>	
Data sources		
Current loop	Type of data	kA
	Frequency of update	For every/last piece
	Amount of data	On internal instrument.
	<Other metadata worth to mention>	
Manometer	Type of data	bar
	Frequency of update	Constant data

	Amount of data	On internal mechanical manometer
	<Other metadata worth to mention>	For air pressure For pressure of hydraulic oil
Thermometer	Type of data	°C
	Frequency of update	Constant data
	Amount of data	On internal mechanical thermometer
	<Other metadata worth to mention>	For cooling water
Flowmeter	Type of data	l/min
	Frequency of update	Constant data
	Amount of data	On internal mechanical flowmeter
	<Other metadata worth to mention>	For cooling water

MACH.GORENJE.03.CELL_OBC	
Machine Identification	
Model	Cell OBC: 1 robot, 1 spot welding machine, 1 press
Manufacturer	1 robot: KUKA 1 spot welding machine: British Federal 1 press: Meccanica Patron
Activity	OB cell is a cell for reshaping the flat pre-formed sheet of the dishwasher base.
General Data	
Age	Cell OBC: 1993 Robots: 2009
Workload	2-3 shifts
Usage	For manufacturing DW tubs
Maintenance	*Preventive maintenance (monthly), 2 times/year main service*
Indexes	
Health	rated capacity: 430 pcs/shift; actively working capacity: 380 pcs/shift Index= 88,4%
Performance	actively working capacity: 380 pcs/shift; scrap: 3 pcs/shift Index=0,8%

Production	4 (on the scale 1-5); after refurbishment we expect better health and performance, evaluated from 1 to 5	
List of Errors		
Mechanical, electrical and SW failures	*Stop of machine because of wrong position, fuse control, overload motor control…; all failures are shown on HMI.*	
Metrics		
MTTF	0,017 – same formula as in the previous machine applies	
MTTR	135 min - same formula as in the previous machine applies	
MTBF	78 days - same formula as in the previous machine applies	
Max Production	380 pcs/shift	
Max Capacity	410 pcs/shift	
Hours down (planned)	50 min/shift	
Hours down (unplanned)	15 min/shift	
Parameters		
Pressure of hydraulic oil	Type of param	Mechanical / manometer
	Frequency of update	For every/last part
	Amount of data	No data
	Picture	
Welding current	Type of param	Sequence
	Frequency of update	For every part
	Amount of data	On switch (digital information)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	Picture	
Air pressure	Type of param	Sequence
	Frequency of update	For every part
	Amount of data	On display (for information)
	Picture	
Processes		
Bending	Type of process	Bending (tool)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

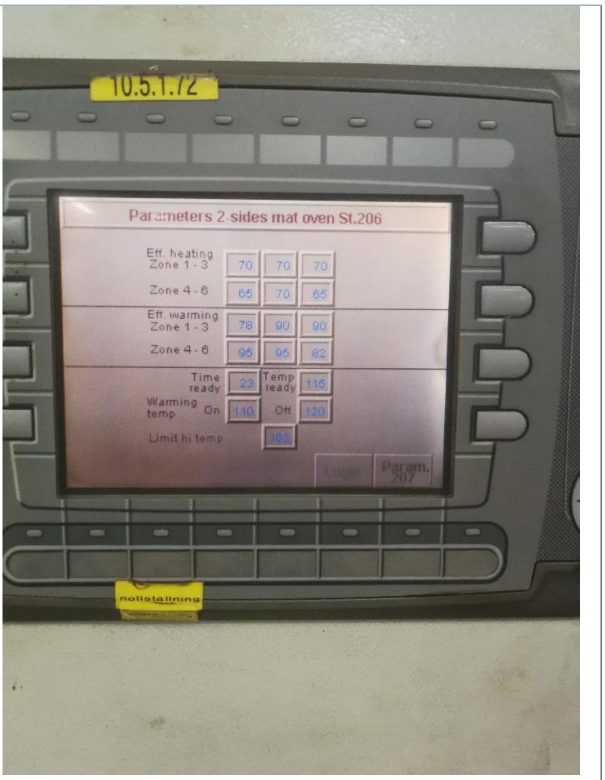
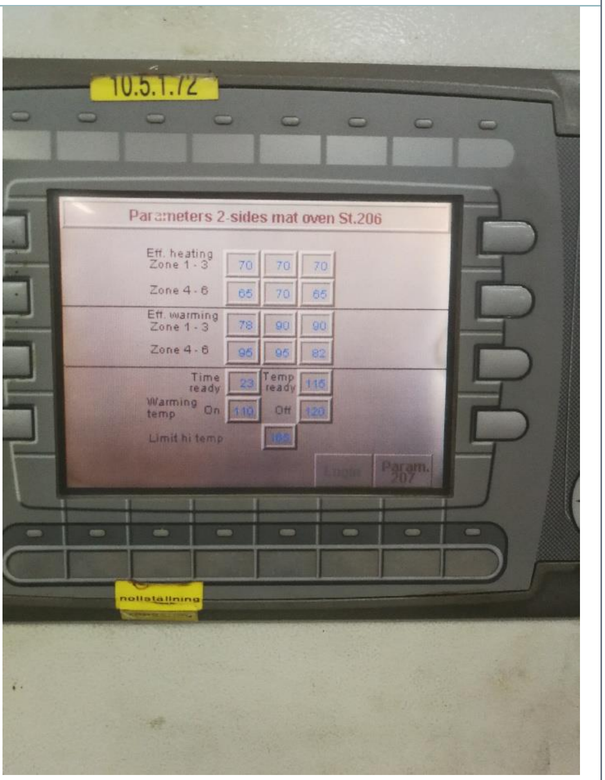
	List of parameters affected	Pressure of hydraulic oil Air pressure
	<Other metadata worth to mention>	
Welding	Type of process	Spot welding
	List of parameters affected	Welding current Air pressure
	<Other metadata worth to mention>	
Data sources		
Current loop	Type of data	kA
	Frequency of update	For every/last piece
	Amount of data	On internal instrument
	<Other metadata worth to mention>	
Manometer	Type of data	bar
	Frequency of update	Constant data
	Amount of data	On internal mechanical manometers
	<Other metadata worth to mention>	For air pressure For pressure of hydraulic oil

MACH.GORENJE.04.CELL_D_E	
Machine Identification	
Model	**Cell D, E**: 4 robots, 2 rotary tables, 4 IR ovens, alignment punching table, transport rolles, measuring equipment
Manufacturer	4 robots: ABB 5 IR ovens: Ircon Measuring equipment: Mitutoyo
Activity	Cell D: The heated bitumen sheet is pressed and melted on the outer part of tub. Containers are transported by robots throughout the cell and data on the type of container is also transmitted through the transfer. Cell E: Additional manual small operations and dimension control are carried out.
General Data	
Age	Cell D, E: 1993 Robots: 2007

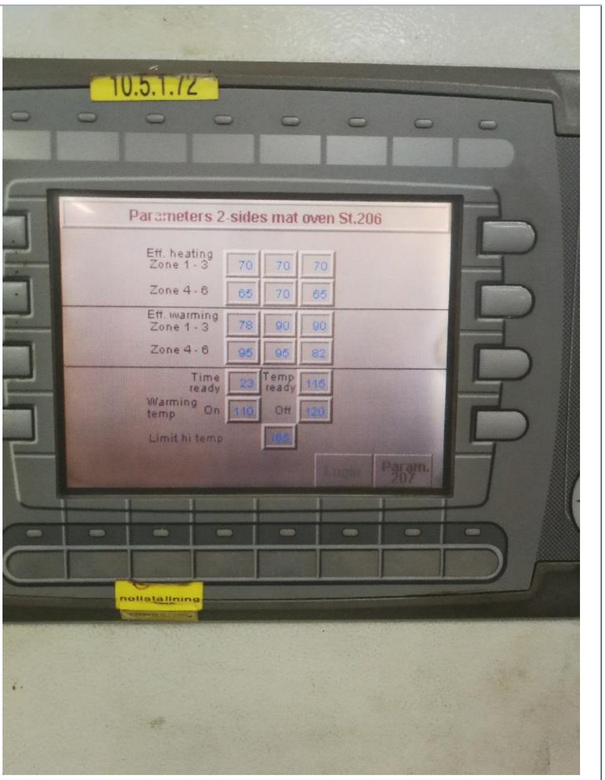
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Workload	2-3 shifts	
Usage	For manufacturing DW tubs	
Maintenance	*Preventive maintenance (monthly), 2 times/year main service*	
Indexes		
Health	rated capacity: 430 pcs/shift; actively working capacity: 380 pcs/shift Index= 88,4%	
Performance	actively working capacity: 380 pcs/shift; scrap: 3 pcs/shift Index=0,8%	
Production	4 (on the scale 1-5); after refurbishment we expect better health and performance, evaluated from 1 to 5	
List of Errors		
Mechanical, electrical and SW failures	*Stop of machine because of wrong position, fuse control, overload motor control…; all failures are shown on HMI.*	
Metrics		
MTTF	0,017 - same formula as in the previous machine applies	
MTTR	73 min - same formula as in the previous machine applies	
MTBF	15,5 days - same formula as in the previous machine applies	
Max Production	380 pcs/shift	
Max Capacity	410 pcs/shift	
Hours down (planned)	50 min/shift	
Hours down (unplanned)	15 min/shift	
Parameters		
Temperature of heaters	Type of param	Data to set the temperature
	Frequency of update	Constant data
	Amount of data	No data

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	Picture	 10.5.1.72 Par::meters 2-sides mat oven St.206 Eff. heating Zone 1 - 3: 70 70 70 Zone 4 - 6: 65 70 65 Eff. warming Zone 1 - 3: 78 90 90 Zone 4 - 6: 95 95 82 Time ready 23 Temp ready 115 Warming temp On 110 Off 120 Limit hi temp 165 Login Param. 207 nollställning
Power of heaters	Type of param	Data to set the power
	Frequency of update	Constant data
	Amount of data	No data
	Picture	 10.5.1.72 Par::meters 2-sides mat oven St.206 Eff. heating Zone 1 - 3: 70 70 70 Zone 4 - 6: 65 70 65 Eff. warming Zone 1 - 3: 78 90 90 Zone 4 - 6: 95 95 82 Time ready 23 Temp ready 115 Warming temp On 110 Off 120 Limit hi temp 165 Login Param. 207 nollställning
Time of IR heating	Type of param	Data to set the time
	Frequency of update	Constant data
	Amount of data	No data

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

	Picture	 10.5.1.72 Parameters 2-sides mat oven St.206 Eff. heating Zone 1 - 3: 70 70 70 Zone 4 - 6: 65 70 65 Eff. warming Zone 1 - 3: 78 90 90 Zone 4 - 6: 95 95 82 Time ready 23 Temp ready 115 Warming temp On 110 Off 120 Limit hi temp 165 Login Param. 207 nollställning
Air pressure	Type of param	Sequence
	Frequency of update	For every part
	Amount of data	On display (for information)
	Picture	
Processes		
	Type of process	Ovens with IR heaters

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Heating the isolation on tab	List of parameters affected	Temperature of heaters Power of heaters Time of heating
	<Other metadata worth to mention>	
Control measuring of tub	Type of process	Control equipment
	List of parameters affected	Air pressure
	<Other metadata worth to mention>	
Data sources		
Pyrometer	Type of data	°C on HMI/OP
	Frequency of update	Constant data
	Amount of data	Internal on HMI
	<Other metadata worth to mention>	
Time of heating	Type of data	sec
	Frequency of update	Constant data
	Amount of data	Internal on HMI
	<Other metadata worth to mention>	
Power of IR heater	Type of data	Power of heater in %
	Frequency of update	Constant data
	Amount of data	Internal on HMI
	<Other metadata worth to mention>	
Manometer	Type of data	bar
	Frequency of update	Constant data
	Amount of data	On internal mechanical manometers
	<Other metadata worth to mention>	For air pressure For pressure of hydraulic oil

MACH.GORENJE_MORA.01.SPRAYING_CABIN	
Machine Identification	
Model	--
Manufacturer	*NORDSON*

Activity	*Cabin with spraying guns and reciprocators for applying enamel powder with help of electrostatic high voltage*
General Data	
Age	*22 years (1998)*
Workload	*2484 pcs per day in one shift*
Usage	*Enamelling of white parts for freestanding cookers (cooktops) - powder coating*
Maintenance	*Preventive maintenance (monthly), 2 times/year main service*
Indexes	
Health	*Maximum capacity: 2484 pcs/shift, daily capacity: 4300 pc/shift*
Performance	*OEE = 76%*
Production	*Formula/Expression used to calculate the production of the machine*
List of Errors	
Small fails of guns function (no spraying - blended, no high voltage, crash on conveyor)	
Metrics	
MTTF	*51,5 hours = real working time / 26 fails = 235*7,5*76% / 26*
MTTR	*Spraying gun - 21 minutes (for 12 cases in 2020)* *Conveyor - 22 minutes (for 14 cases in 2020)*
MTBF	*Spraying gun - 25,6 days (from 2days to 122 days), 12 cases in 2020* *Conveyor - 22 days (from 1 to 60 days), 14 cases in 2020*
Max Production	*2484 pcs of cooktops / day*
Max Capacity	*4300 pcs of cooktops in two shifts*
Hours down (planned)	*150*
Hours down (unplanned)	*20*
Parameters	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Temperature	Type of param	
	Frequency of update	*Automatically controlled*
	Amount of data	
Humidity	Type of param	
	Frequency of update	*Automatically controlled*
	Amount of data	
High voltage for guns	Type of param	
	Frequency of update	*Automatically controlled*
	Amount of data	
Quantity of pressured air	Type of param	
	Frequency of update	*Automatically controlled*
	Amount of data	
Processes		
<Process 1>	Type of process	*Identify the type of process*
	List of parameters affected	*List the parameters from the section above that are affected by this process*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
<Process N>	Type of process	*Identify the type of process*
	List of parameters affected	*List the parameters from the section above that are affected by this process*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
Data sources		
<Data Source 1>	Type of data	*Identify the type of data source*
	Frequency of update	*Which is the frequency of the update related to the data source*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	Amount of data	*Amount of data related to the data source*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
<Data Source N>	Type of data	*Identify the type of data source*
	Frequency of update	*Which is the frequency of the update related to the data source*
	Amount of data	*Amount of data related to the data source*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*

MACH.GORENJE_MORA.02.BURNING_FURNACE	
Machine Identification	
Model	*VGT/U18,6*
Manufacturer	*VGT*
Activity	*Furnace with burning temperature about 830°C, with its heating / cooling zone and high temperature zone*
General Data	
Age	*22 years*
Workload	*2848 pcs per day in one shift*
Usage	*Enamelling of white parts for freestanding cookers (cooktops) - burning of powdered parts*
Maintenance	*Weekly cleaning, monthly check of inside of oven, service of burners and tubes once a year*
Indexes	
Health	
Performance	
Production	
List of Errors	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Fails of conveyor	Fall down parts in oven	
Metrics		
MTTF	*Every failure on the furnace is reparable*	
MTTR	*Burner - 90 minutes (for 2 cases in 2020)*	
MTBF	*Burner - 165 days, 2 cases in 2020*	
Max Production	*2848 pcs of cooktops / day*	
Max Capacity	*4300 pcs of cooktops in two shifts*	
Hours down (planned)	*1500*	
Hours down (unplanned)	*200*	
Parameters		
Temperature	Type of param	
	Frequency of update	*Automatically controlled*
	Amount of data	
Speed of conveyor	Type of param	
	Frequency of update	*Constant speed (electromotor)*
	Amount of data	
Processes		
<Process 1>	Type of process	*Identify the type of process*
	List of parameters affected	*List the parameters from the section above that are affected by this process*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
<Process N>	Type of process	*Identify the type of process*
	List of parameters affected	*List the parameters from the section above that are affected by this process*

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
Data sources		
<Data Source 1>	Type of data	*Identify the type of data source*
	Frequency of update	*Which is the frequency of the update related to the data source*
	Amount of data	*Amount of data related to the data source*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*
<Data Source N>	Type of data	*Identify the type of data source*
	Frequency of update	*Which is the frequency of the update related to the data source*
	Amount of data	*Amount of data related to the data source*
	<Other metadata worth to mention>	*Other metadata worth to mention may be added as needed*

2.4 Fluchos

MACH.FLUCHOS.01.CZ/M	
Machine Identification	
Model	CZ/M
Manufacturer	COMELZ
Activity	Automatic leather cutting machine
General Data	
Age	4 years (constructed at 2016)
Workload	8-hour working day
Usage	This machine is used for cutting small batches of leather and other synthetic materials
Maintenance	Preventive maintenance is carried out
Indexes	
Health	Subjective measurement. Since they only repair the machines once they are broken, so they do not have any other way to know the real health of the different machines.
Performance	Nothing formal is defined here. There is a machine used for a very low number of pairs of shoes. It is a machine they use for small runs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	in the design department, for when they want to see how the new models work before they are put on the market.
Production	90 pairs of shoes, in an 8-hour day, considering that it is used for shoe models that are in the design phase, it is not a machine that is within the production line itself. The number of pairs depends on the parts contained in the model
List of Errors	
To date the machine has not had any major failures, as it is relatively new. The main faults it has had so far have been in the spindles due to wear and tear, and in the cables that communicate the spindle with the electronic boards. The most frequent problem is the failure to calibrate the cutting system	
Metrics	
MTTF	The failures are minimal, given the short age of the machine, so the time between failures is very long. They do not keep track of this data.
MTTR	The few times that failures have occurred, the repair time has varied between 15 and 30 minutes, depending on the severity of the failure. They do not keep track of this data.
MTBF	Since the machine has suffered very few breakdowns in its four years of life, the time between breakdowns is very long, although they do not keep track of this data.
Max Production	This machine does not work within the footwear production line. It is part of the design part of the models that are going to be tested to be put on the market, so it always works at maximum capacity, which would be about 90 pairs throughout the 8-hour day
Max Capacity	
Hours down (planned down)	None
Hours down (unplanned down)	None
<Other Metrics>	
Parameters	
Note: The data provided by the machine's computer systems help to plan maintenance interventions, both electrical and mechanical	

Speed	Type of param	Speed (m/s, double)
	Frequency of update	Full continuous cycles
	Amount of data	Data is not being recorded onto a system nor being written down, so there is no historical data
Temperature	Type of param	Temperature (ºC, double)
	Frequency of update	Full continuous cycles
	Amount of data	Data is not being recorded onto a system nor being written down, so there is no historical data
Pressure	Type of param	Pressure (PSI, double)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

	Frequency of update	Full continuous cycles
	Amount of data	Data is not being recorded onto a system nor being written down, so there is no historical data
Processes		
Conveyor belt movement	Type of process	Conveyor belt movement
	List of parameters affected	Speed, temperature
Cutting	Type of process	Cutting
	List of parameters affected	Temperature
Suction	Type of process	Suction
	List of parameters affected	Pressure
Data sources		
The machines that have data sources (which are not all) are analogical, so the information is displayed continuously. Pressure is measured with pressure gauges and the temperature with thermometers.		

2.5 Podium

The situation of Podium is different compared to the other User partners of RECLAIM. Podium does not have any machine connected to any device and/or software application to retrieve any kind of information from the machines. Every single process is manual, and they do not even have a formal schedule for checking the health of the machines.

To overcome this situation, Podium is working hand-to-hand with SUPSI RECLAIM partner to develop, as part of the activities within T2.5 (Holistic Life-Cycle Machinery Models for facilitating Refurbishment & Remanufacturing), a model to characterise Podium machinery and, thus, help in make it available to the rest of the consortium. In that sense, the information present in the deliverable D2.5 is used as a base for the specification of Podiums machinery.

As part of their normal business activities, SUPSI is developing a tool that will be extended to cover RECLAIM's needs from Podium and the output of this tool is what is going to be transferred into the RECLAIM Repository for the generation of the indexes. The output being produced by SUPSI's tool is as follows:

- MTBF/MTTF of each component composing the selected system/machine [hours]
- MTBF/MTTF standard deviation of each component composing the selected system/machine [hours]
- Reliability of each component composing the selected system/machine [probability of failure given current life ≅ health index]
- Function parameters of the selected function adopted to fit component's failure data
- MTBF/MTTF of the system composed by all the components in series [hours]
- MTBF/MTTF standard deviation composed by all the components in series [hours]

- Reliability of the system/machine composed by all the components in series [probability of failure given current life ≅ health index]

All these items are exported in the form of JSON file, in which an example can be explored just below.

```
{
  "systemName": <str>,
  "systemId": <int>,
  "workingHours": <float>,
  "currentReliability": <float>,
  "meanTimeBetweenFailures": <float>,
  "analysisExecutionTimestamp": <ISO/UNIX timestamp>,
  "components": [
    {
      "componentName": <str>,
      "repairable": <boolean>,
      "componentId": <int>,
      "meanFailureTime": <float>,
      "stdDeviation": <float>,
      "currentReliability": <float>,
    }
    {
      "componentName": <str>,
      "repairable": <boolean>,
      "componentId": <int>,
      "meanFailureTime": <float>,
      "stdDeviation": <float>,
      "currentReliability": <float>,
    }
    {
      "..." : "..."
    }
  ]
}
```

2.6 Zorluteks

MACH.ZORLUTEKS.01.BLEACHING_MACHINE	
Machine Identification	
Model	1997 YEARS KUSTERS CONTINUOUS BLEACHING MACHINE
Manufacturer	KUSTERS CALICO MACHINERY PVT. LTD

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Activity	OPEN WİTDH CONTINUOUS BLEACHING OF FABRIC	
General Data		
Age	23	
Workload	Bleaching machine works 24/7 continuously as long as any planned and unplanned down happen with human-machine relationship.	
Usage	BLEACHING OF COTTON FABRIC	
Maintenance	*Periodic maintenance is performed 1 time in a year which takes 7 days.*	
Indexes		
Health	Not calculated. Only a visual inspection is performed. According to the diagram shown in Figure 1, the current value should be about 5 (Poor)	
Performance	Ration between actual production with maximum production. Current value is %77	
Production	Zorluteks uses SCADA system to get production-related information. This data is also taken from the SCADA system. An approximate production rate for our pilot machine is 100000 m fabric/day	
List of Errors		
Failure at chemical dosing system, low or high temperature at washing baths and steamer, failure at electrical wirings, less and excess time of steaming,	*failure at valves for water supply, failure at bearings, failure at moisture sensor in dryer, low or high drying temperature, failure at engines*	
Metrics		
MTTF	This value is meaningless for our bleaching machine.	
MTTR	MTTR=(total maintenance time)/(total #of repairs). Current value is 20,7 min	
MTBF	MTBF=(total operational time)/(total # of failures). Current value is 1,3 days	
Max Production	Information is taken from Scada System. Current value is 115000 m fabric/day	
Max Capacity	Information is taken from Scada System. Current value is 1296000 m fabric/day	
Hours down (planned)	Data taken from Online platform which takes data from PLCs on the machine. In this platform, we can determine energy consumptions, efficiency, reasons for stops and its duration for each machine in the production plant. Current value is 7 days/year	
Hours down (unplanned)	Data taken from Online platform which takes data from PLCs on the machine. In this platform, we can determine energy consumptions, efficiency, reasons for stops and its duration for each machine in the production plant. Current value is 21 days/year	
Parameters		
Chemical Concentration	Type of param	String.
	Frequency of update	1 time in a month, system is checked.
	Amount of data	unknown

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

Temperature of washing baths	Type of param	85-95 °C
	Frequency of update	1 time in a year, system is checked as long as any failure occurs.
	Amount of data	unknown
Temperature of steamer	Type of param	98 °C
	Frequency of update	1 time in a year, system is checked as long as any failure occurs.
	Amount of data	unknown
Time of steaming	Type of param	18 min
	Frequency of update	1 time in a year, system is checked as long as any failure occurs.
	Amount of data	unknown
Pick-up value	Type of param	130-140 %
	Frequency of update	1 time in a year, system is checked as long as any failure occurs.
	Amount of data	unknown
Processes		
Bleaching	Type of process	Bleaching of cotton fabric
	List of parameters affected	Temperature of steamer, temperature of washing baths, pick-up value, time of steaming, chemical (caustic, hydrogen peroxide, stabilizer, sequestering agent, and wetting agent) concentration NB: The bleaching process is explained in Annex A
Data sources		
SCADA SYSTEM	Type of data	Production-related information such as production amount and process flow for every order which also include time of start and end of the processes, information of operator who is responsible for running the process, production plan for a specific time period, type of product produced etc.
	Frequency of update	transiently
	Amount of data	The volume is estimated to several Megabytes per day based on the production volume.
PLC	Type of data	Production and machine related information such as amount of energy and water usage daily, temperature of the washing baths and steamer, recipe, velocity of the machine, type of product produced, etc.

	Frequency of update	Transiently
	Amount of data	PLCs data is stored at SCADA System and the online platform which includes several Megabytes per day based on the production volume.
Online platform which takes data from PLCs	Type of data	Data related with energy consumptions, efficiency and reasons for stops for each machine in the production plant
	Frequency of update	Transiently
	Amount of data	Several Megabytes per day

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

3 Indexes: Definition

After having a look at how the different Pilot partners from RECLAIM define their own indexes, the following sections provide the global definition for the entire consortium and how they will be calculated when it is time to exploit RECLAIM by offering the results to the market once the project is over.

However, the indexes will be refined once the pilots provide more information about the data they can share within the project and the collection of data task is allowing RECLAIM to reconsider the different formulas.

3.1 Health-Index

As already stated at the beginning of this document, the health is directly related to the reduction (or prevention) of failures of machinery, production lines and/or equipment. As such, and after the review of the input from the different industrial partners, machine "health" can be defined as a measurement based on parameters related to the amount of time that the machinery and/or the equipment has been ready to be working or actively working.

The main problem related to assessing the health of a machine is that, typically, this assessment is based on a subjective inspection of the machine, which besides not being formalised it entails to be the most difficult to measure and assess. As mentioned at the beginning of this document, some authors have provided a scale-based categorisation index based on the information available and/or collected from the different machinery (see Section 1.2 and Figure 1) where the level of health-index for a 'Failing Bearing Degradation Curve' was shown.

The assessment of the variables to calculate the health-index is performed based on several physical tests such as vibration, acoustic emission, oil sampling, valve monitoring, and even the most common of all monitoring methods – tactile senses and intuition – as information sources. Further, the health-index will also be related to the failure metrics such as Mean-Time-to-Failure (MTTF), Mean-Time-to-Repair (MTTR) or Mean-Time-Between-Failures (MTBF).

As such, the formula proposed relates the MTBF with the amount of production hours (see Figure 4). As an example, assuming that the production hours account for 1000h:

- If we have an MTBF = 100 (i.e., one failure every 100h), then the Health is 0,1
- If we have an MTBF = 900 (i.e., one failure every 900h), then the Health is 0,9
- If we have an MTBF = 1 (i.e., one failure every hour), then the Health is 0,001
- Finally, if we have a broken machine (i.e., MTBF = 0), then the Health is 0

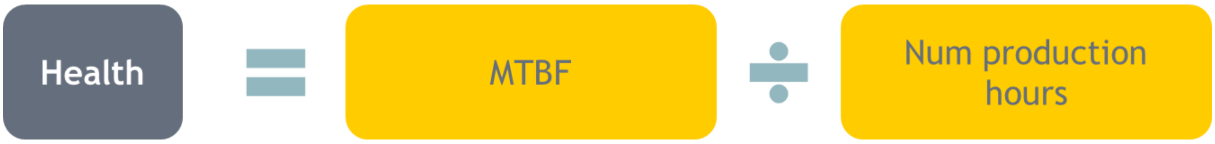


Figure 4: Machine Health Index calculation

3.2 Performance-Index

Since the performance index machine is related to the throughput of the machine (e.g., products per minute, amount of debris produced, etc) depending to a certain extent on the quality of the product produced as well, any form of performing its calculation must involve both actual production and maximum production. In other words, how fast a machine performs a work and how good it is (velocity vs quality).

In fact, all RECLAIM User partners are in favour of this type of formula since they are already calculating this index following similar formulas.

As such, the formula proposed by Sight Machine is the one chosen to calculate the Performance Index. Sight Machine proposed the name of the Manufacturing Performance Index (MPI), which is an easy-to-understand metric that defines factory performance as a ratio of actual production achieved compared to the maximum production the factory is designed for.

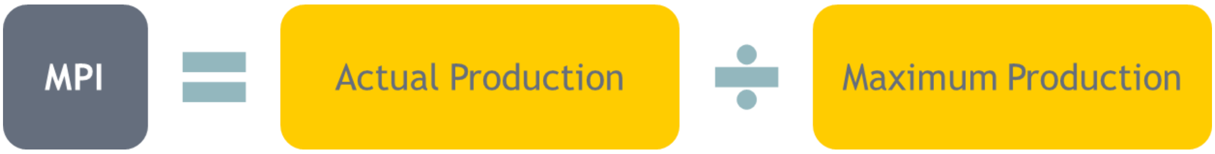


Figure 5: Manufacturing Performance Index (MPI) calculation

3.3 Production-Index

After the analysis of the information provided by the different User partners, the closest definition to the Production index is the one related to the throughput of a machine in which it measures the average number of units being produced on a machine, line, unit or plant over a specified period of time, e.g.: units per minute. In this sense, if the throughput of a given machine suddenly decreases, then it is most likely that such machine is probably having issues while manufacturing.

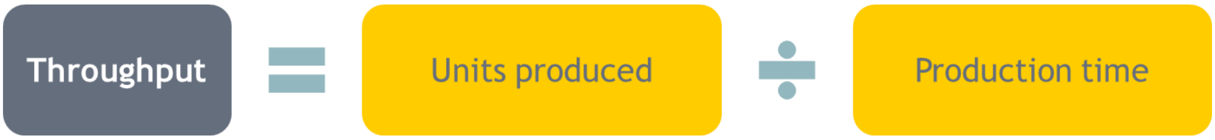


Figure 6: Production Index (Throughput) calculation

4 RECLAIM Repository

4.1 Architecture

Today, terabytes of machine data are being generated in real time from sensors embedded in industrial equipment. The capability to handle such large data streams to draw actionable conclusions is a task that requires artificial intelligence, but to get to that point there are a few steps that need to be achieved.

Initial activities for calculating the RECLAIM indexes should be easily achievable by the RECLAIM partners since most readings can be easily measured by affordable sensors often connected to a PLC to keep track of important parameters. Data logging starts at the edge where the data is collected directly from the equipment, prepared, and transferred to be stored in the RECLAIM repository.

The first thing that needs to be set up is the collection of long-term (historical) data during stable machine operations. The historical data set, with records collected over time (e.g., quarter or year), can be used for advanced ML algorithms that analyse and detect causal correlations in the incoming data records.

This IoT-driven approach makes it possible to assess equipment health, performance, and production (throughput) by monitoring machine parameters such as vibration frequency, rotations, engine temperature and ambient variables (e.g., temperature, humidity, or pressure). Manufacturers from various domains can use IoT devices to monitor machinery and to check the quality of the products and components manufactured on it.

The diagram shown in Figure 7 sketches, in a simplified way, the data flow from the machines at the *edge* (where sensors can easily be placed) to the heart of the RECLAIM DSF, where the three indexes, namely Health, Performance, and Production, will be calculated. Finally, the user will be shown with these values so they can react by proposing the necessary actions.

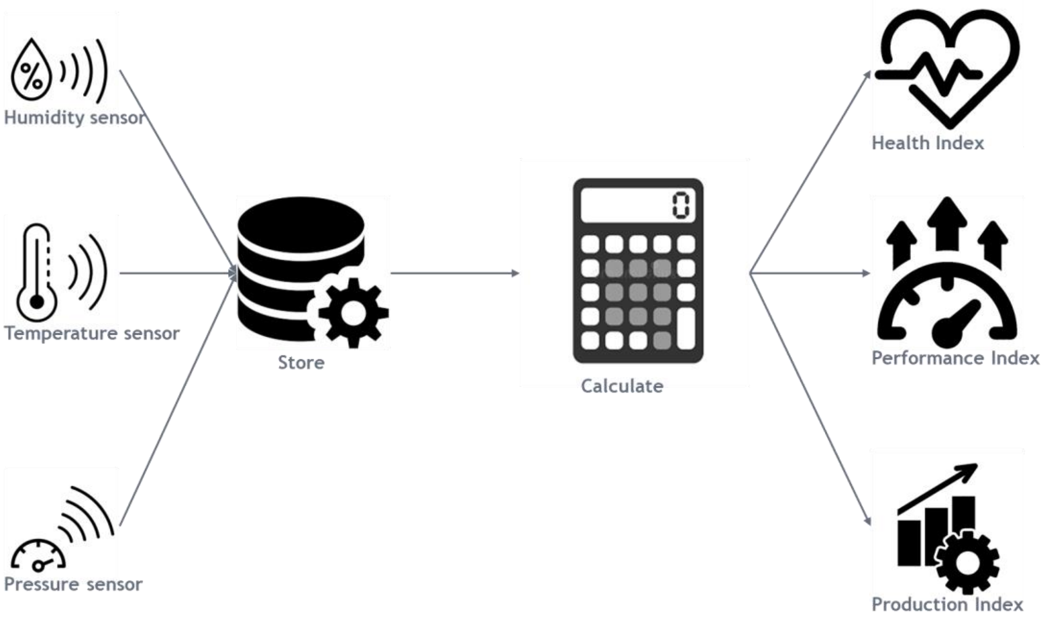


Figure 7: Component high-level data flow

DRyICE is the tool selected for taking care of the storage and the calculation of these indexes in real time. DRyICE is composed of a timeseries data warehouse as main storage system, a set of microservices that perform small operations on the data, a message bus to pass data between microservices, and a user interface for displaying multiple dashboards. DRyICE also

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

comes with an API for setting up ingestion tasks as well as for internal management of microservices from an admin user interface. This API is the main point of data ingestion for RECLAIM. This is explained further in Section 4.2.3. Figure 8 shows DRyICE architecture including its different elements.

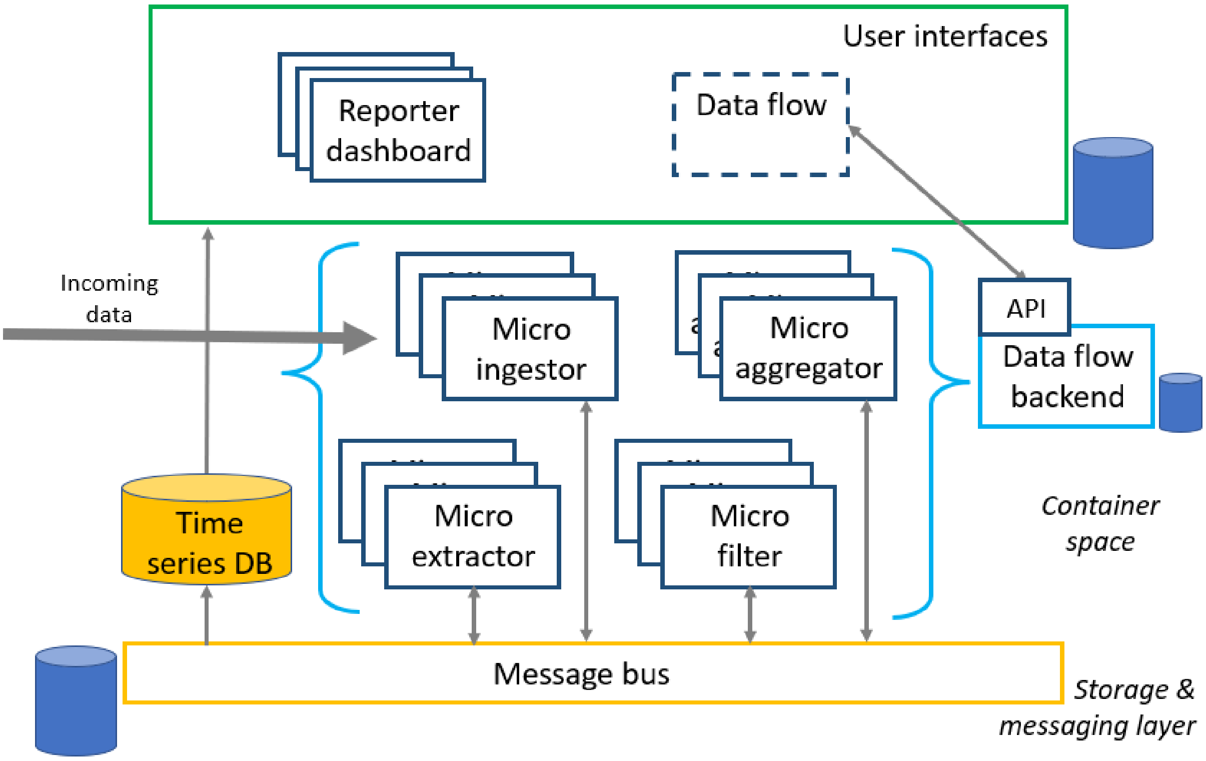


Figure 8: DRyICE architecture

4.2 Accessing the RECLAIM Repository

This section details how the data can be accessed, stored and retrieved by two means:

- Through the embedded UI Dashboard
- Through the REST API

4.2.1 RECLAIM Repository UI Access

The RECLAIM Repository can be accessed through its User Interface for viewing data in a human-readable way by using the following URL to the standalone instance:

- https://iceberg.icelab.cloud/

The credentials to access it are as follows:

- Username: reclaim
- Password: R3claim!

This user will grant access to the following functionalities:

- View dashboards, where access to the published dashboards is granted
- View Charts, where access to the created charts is granted
- SQL Lab, where a SQL Editor can be found to consult all tables created from the data that has been uploaded into the repository.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

4.2.2 RECLAIM Repository REST API Access

This API will allow the RECLAIM users to upload data into the repository by using the different methods available. In order to execute the methods in the REST API, the following the API-KEY=C5CC3F19 has to be used.

- The URL base for all entry points is https://iceberg.icelab.cloud/docbase/
- The documentation (see next section) can be found also at https://iceberg.icelab.cloud/docbase/docs

4.2.3 RECLAIM Repository REST API Documentation

This section describes the different methods available for accessing and storing data in the repository.

4.2.3.1 Machine Operations' Toolkit

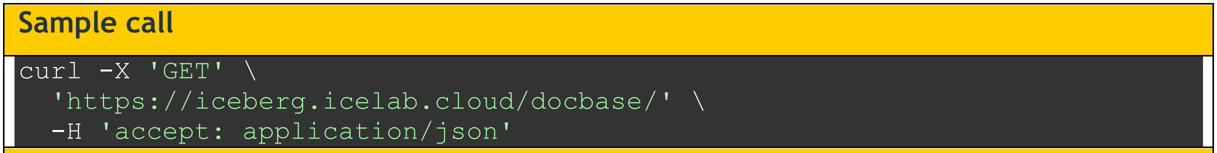
Method		Test API
URL		
/		
Method		
GET		
Description		
This method is used to test that the program is working. It will return a JSON with the "Hello: World" and the timestamp when it is invoked		
URL Params		
Required:		
Optional:		
Data Params		
Required:		
Optional:		
Success response		

200
Content:
`{`
`"Hello": "World",`
`"datetime": "2022-06-01T14:52:15.660873"`
`}`
		Request was successful

Error response		
400		*Bad Request*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Sample call



```
curl -X 'GET' \
  'https://iceberg.icelab.cloud/docbase/' \
  -H 'accept: application/json'
```

Notes

Method		Get Projects
URL		
/rest2/{project}		
Method		
GET		
Description		
This method will list all the available tables in the RECLAIM Repository for the "project" requested		
URL Params		
Required:		
Project	string	*This parameter is used to filter the tables returned. It will return all the tables that have the project parameter as part of its name*
Optional:		
Data Params		
Required:		
Optional:		
Success response		
200 Content: ["hwh_welding_load", "hwh_welding_sensors", "single_welding", "welding", "welding_load_data", "welding_sensors"]		*Request was successful*
Error response		
400		*Bad Request*
403		*Unauthorized*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'GET' \
  'https://iceberg.icelab.cloud/docbase/rest2/welding' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
```

Notes

Method		Get Project Factory Machine Data
URL		
/rest2/{project}/{factory}/{machine}		
Method		
GET		
Description		
This method will get the data stored in the RECLAIM Repository using the project, factory, and machine parameters		
URL Params		
Required:		
Project	string	*Name of the project or pilot where the machine is running*
Factory	string	*Name of the factory where the machine is installed*
machine	string	*Name of the machine that is generating the data*
Optional:		
page_number	integer	*Paging parameter. Page Number to get data based on Page size* *Default value: 1*
page_size	integer	*Paging parameter. Number of records to be returned per page* *Default value: 100*
order_by	string	*Paging parameter. Column used to sort the data* *Default value: timestamp*
loading_date_start	date	*Search parameter. Initial date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.*

		Example: 2021-11-11
loading_date_end	date	*Search parameter. Final date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_start	date	*Search parameter: Initial date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_end	date	*Search parameter: Final date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
custom_search	string	*Search parameter: This is a JSON object that contains search parameters for the data uploaded. All the data that matches the object properties and values will be returned (AND operation)* *Date format: JSON object.* *Default value: null value.* *Example:* *{* *"property": 100,* *"property_2": "value"* *}*
column	array[string]	*A list of names of the columns that will be retrieved in the JSON objects.* *Example: column=timestamp&column=iceTimeIn&column=column_name_1&column=column_name_2*
Data Params		
Required:		
Optional:		

Success response

200 Content:	*Request was successful*

```
[
  {
    "machine_id": 20,
    "component_id": 20,
    "stop_type_id": 20,
    "timestamp": "2022-04-10T00:00:00",
    "iceTimeIn": "2022-04-19T18:07:32.647764",
    "modelId": "project1factory1machine1",
    "project": "project1",
    "factory": "factory1",
    "machine": "machine1",
    "task": "",
    "streamName": "project1factory1machine1",
    "timestamp2": "2022-04-10T00:00:00"
  },
  {
    "machine_id": 55,
    "component_id": 56,
    "stop_type_id": 71,
    "timestamp": "2022-04-18T00:00:00",
    "iceTimeIn": "2022-04-19T17:28:34.370602",
    "modelId": "project1factory1machine1",
    "project": "project1",
    "factory": "factory1",
    "machine": "machine1",
    "task": "",
    "streamName": "project1factory1machine1",
    "timestamp2": "2022-04-18T00:00:00"
  }
]
```

Error response

400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'GET' \
'https://iceberg.icelab.cloud/docbase/rest2/project1/factory1/machine1?page_number=1&page_size=100&order_by=timestamp&loading_date_start=2022-04-10&data_date_end=2022-04-19' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
```

Notes

Method	Add Project Factory Machine Data

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

URL		
/rest2/{project}/{factory}/{machine}		
Method		
POST		
Description		
This method will insert the data into the RECLAIM Repository using the project, factory, and machine parameters		
URL Params		
Required:		
Project	string	*Name of the project or pilot where the machine is running*
Factory	string	*Name of the factory where the machine is installed*
Machine	string	*Name of the machine that is generating the data*
Optional:		
Data Params		
Required:		
request_body	JSON object	*An array of JSON objects with the following format.* • **timestamp:** This property will have the timestamp when the data was created. It must use the standard ISO format. • **data:** This property is a JSON object with the values to be inserted into the RECLAIM Repository. *it does not support inner objects *Example:* *[* *{* *"timestamp": "2021-11-16T21:01:36.123456",* *"data": {* *"name": "My Sensor 22",* *"position": "up",* *"cms": "cms",* *"value": "150"* *},* *{* *"timestamp": "2021-11-16T21:01:40.123456",* *"data": {* *"name": "My Sensor 22",* *"position": "left",* *"cms": "cms",* *"value": "5"* *}*

		]
Optional:		
Success response		

200
Content:
[
{
"name": "My Sensor 22",
"position": "up",
"cms": "cms",
"value": "150",
"timestamp": "2021-11-16T21:01:36.123456",
"iceTimeIn": "2022-09-14T18:28:39.752017",
"modelId": "TestProjectTestFactoryTestMachine",
"project": "TestProject",
"factory": "TestFactory",
"machine": "TestMachine",
"task": "",
"streamName": "TestProjectTestFactoryTestMachine"
},
{
"name": "My Sensor 22",
"position": "down",
"cms": "cms",
"value": "50",
"timestamp": "2021-11-16T21:01:38.123456",
"iceTimeIn": "2022-09-14T18:28:39.752017",
"modelId": "TestProjectTestFactoryTestMachine",
"project": "TestProject",
"factory": "TestFactory",
"machine": "TestMachine",
"task": "",
"streamName": "TestProjectTestFactoryTestMachine"
},
{
"name": "My Sensor 22",
"position": "left",
"cms": "cms",
"value": "5",
"timestamp": "2021-11-16T21:01:40.123456",
"iceTimeIn": "2022-09-14T18:28:39.752017",
"modelId": "TestProjectTestFactoryTestMachine",
"project": "TestProject",
"factory": "TestFactory",
"machine": "TestMachine",
"task": "",
"streamName": "TestProjectTestFactoryTestMachine"
}
]
		Request was successful

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Error response	
400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'POST' \

'https://iceberg.icelab.cloud/docbase/rest2/TestProject/TestFactory/TestMachine' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: application/json' \
  -d '[
  {
    "timestamp": "2021-11-16T21:01:36.123456",
    "data": {
      "name": "My Sensor 22",
      "position": "up",
      "cms": "cms",
      "value": "150"
    }
  },
  {
    "timestamp": "2021-11-16T21:01:38.123456",
    "data": {
      "name": "My Sensor 22",
      "position": "down",
      "cms": "cms",
      "value": "50"
    }
  },
  {
    "timestamp": "2021-11-16T21:01:40.123456",
    "data": {
      "name": "My Sensor 22",
      "position": "left",
      "cms": "cms",
      "value": "5"
    }
  }
]
'
```

Notes

Method	Get Project Factory Machine Task Data
URL	
/rest2/{project}/{factory}/{machine}/{task}	
Method	
GET	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Description		
This method will get the data stored in the RECLAIM Repository using the project, factory, machine, and task parameters		
URL Params		
Required:		
Project	string	*Name of the project or pilot where the machine is running*
Factory	string	*Name of the factory where the machine is installed*
Machine	string	*Name of the machine that is generating the data*
Task	string	*Name of the task or sensor that is producing the data*
Optional:		
page_number	integer	*Paging parameter. Page Number to get data based on Page size* *Default value: 1*
page_size	integer	*Paging parameter. Number of records to be returned per page* *Default value: 100*
order_by	string	*Paging parameter. Column used to sort the data* *Default value: timestamp*
loading_date_start	date	*Search parameter. Initial date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
loading_date_end	date	*Search parameter. Final date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_start	date	*Search parameter: Initial date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

data_date_end	date	*Search parameter: Final date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
custom_search	string	*Search parameter: This is a JSON object that contains search parameters for the data uploaded. All the data that matches the object properties and values will be returned (AND operation)* *Date format: JSON object.* *Default value: null value.* *Example:* *{* *"property": 100,* *"property_2": "value"* *}*
column	array[string]	*A list of names of the columns that will be retrieved in the JSON objects.* *Example: column=timestamp&column=iceTimeIn&column=column_name_1&column=column_name_2*
Data Params		
Required:		
		•
Optional:		
Success response		

200
Content:
`[`
`  {`
`    "name": "My Sensor 30",`
`    "position": "up",`
`    "cms": "cms",`
`    "value": 150,`
`    "timestamp": "2021-11-16T21:01:36.659842",`
`    "iceTimeIn": "2022-09-14T18:34:07.952824",`
`    "modelId": "TestProj2TestFact2TestMach2TestTask2",`
`    "project": "TestProj2",`
`    "factory": "TestFact2",`
`    "machine": "TestMach2",`
		Request was successful

```
    "task": "TestTask2",
    "streamName":
"TestProj2TestFact2TestMach2TestTask2",
    "timestamp2": "2021-11-16T21:01:36"
  }
]
```

Error response	
400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'GET' \
'https://iceberg.icelab.cloud/docbase/rest2/TestProj2/TestFact2/TestMach2
/TestTask2?page_number=1&page_size=100&order_by=timestamp&custom_search=%
7B%22value%22%3A%20150%7D' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
```

Notes

Method		Add Data to specific factory machine
URL		
/rest2/{project}/{factory}/{machine}/{task}		
Method		
POST		
Description		
This method will insert the data into the RECLAIM Repository using the project, factory, machine and task parameters		
URL Params		
Required:		
Project	string	*Name of the project or pilot where the machine is running*
Factory	string	*Name of the factory where the machine is installed*
Machine	string	*Name of the machine that is generating the data*
Task	string	*Name of the task or sensor that is producing the data*
Optional:		
Data Params		
Required:		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

request_body	JSON object	*An array of JSON objects with the following format.* • timestamp: This property will have the timestamp when the data was created. It must use the standard ISO format. • data: This property is a JSON object with the values to be inserted into the RECLAIM Repository. *it does not support inner objects
Optional:		
Success response		

200
Content:
[
  {
    "name": "My Sensor 22",
    "position": "up",
    "cms": "cms",
    "value": "60",
    "timestamp": "2021-11-
16T21:01:43.123456",
    "iceTimeIn": "2022-09-
14T18:34:07.952824",
    "modelId":
"TestProj2TestFact2TestMach2TestTask2",
    "project": "TestProj2",
    "factory": "TestFact2",
    "machine": "TestMach2",
    "task": "TestTask2",
    "streamName":
"TestProj2TestFact2TestMach2TestTask2"
  },
  {
    "name": "My Sensor 30",
    "position": "up",
    "cms": "cms",
    "value": "150",
    "timestamp": "2021-11-
16T21:01:36.659842",
    "iceTimeIn": "2022-09-
14T18:34:07.952824",
    "modelId":
"TestProj2TestFact2TestMach2TestTask2",
    "project": "TestProj2",
    "factory": "TestFact2",
    "machine": "TestMach2",
    "task": "TestTask2",
    "streamName":
"TestProj2TestFact2TestMach2TestTask2"
  },
  {
    "name": "My Sensor 30",
    "position": "down",
    "cms": "cms",
    "value": "45",
		Request was successful

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

```
    "timestamp": "2021-11-16T21:01:37.512698",
    "iceTimeIn": "2022-09-14T18:34:07.952824",
    "modelId": "TestProj2TestFact2TestMach2TestTask2",
    "project": "TestProj2",
    "factory": "TestFact2",
    "machine": "TestMach2",
    "task": "TestTask2",
    "streamName": "TestProj2TestFact2TestMach2TestTask2"
  }
]
```

Error response	
400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'POST' \

'https://iceberg.icelab.cloud/docbase/rest2/TestProj2/TestFact2/TestMach2/TestTask2' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: application/json' \
  -d '[
  {
    "timestamp": "2021-11-16T21:01:43.123456",
    "data": {
      "name": "My Sensor 22",
      "position": "up",
      "cms": "cms",
      "value": "60"
    }
  },
  {
    "timestamp": "2021-11-16T21:01:36.659842",
    "data": {
      "name": "My Sensor 30",
      "position": "up",
      "cms": "cms",
      "value": "150"
    }
  },
  {
    "timestamp": "2021-11-16T21:01:37.512698",
    "data": {
      "name": "My Sensor 30",
      "position": "down",
      "cms": "cms",
      "value": "45"
    }
  }
]'
```

Notes

4.2.3.2 Fluchos Pilot – REST API to upload data files to the RECLAIM Repository

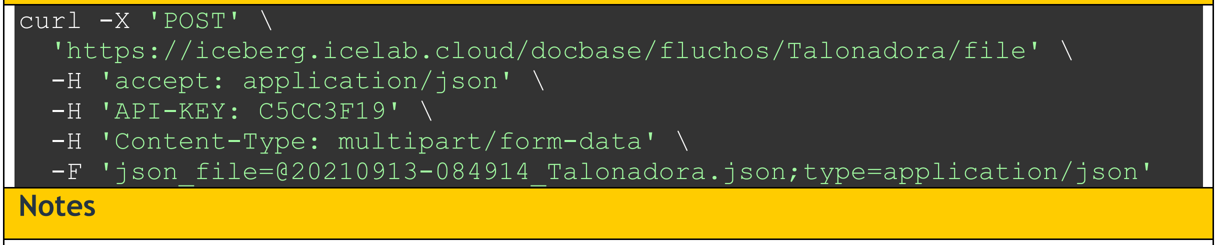
Method		Upload Fluchos File
URL		
/fluchos/{model}/file		
Method		
POST		
Description		
This method is used to upload Fluchos data. It receives a file with the format for Talonadora or Rotostir machines and inserts the data in the RECLAIM Repository		
URL Params		
Required:		
Model	string	*This is the Fluchos data type to upload* *Values: "Talonadora" or "Rotostir"*
Optional:		
Data Params		
Required:		
json_file	string ($binary)	
Optional:		
Success response		

200
Content:
`[`
`  {`
`    "timestamp": "2021-09-13T08:49:05.000",`
`    "batchNo": "20220915-1",`
`    "iceTimeIn": "2022-09-15T01:18:54.295149",`
`    "streamName": "Talonadora",`
`    "modelId": "Talonadora",`
`    "SN": "98356",`
`    "name": "fluchos",`
`    "Foco_Caliente_IZQ": null,`
`    "Foco_Frio_IZQ": null,`
`    "Din1": 0,`
`    "Din2": null,`
`    "Pulsadorfococaliente": null,`
`    "Pulsadorfocofrio": null`
`  }`
`]`
		Request was successful

Error response		

400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call



```
curl -X 'POST' \
  'https://iceberg.icelab.cloud/docbase/fluchos/Talonadora/file' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: multipart/form-data' \
  -F 'json_file=@20210913-084914_Talonadora.json;type=application/json'
```

Notes

Sample file:

20210913-084914_Talonadora.json

Method		Upload Fluchos JSON
URL		
/fluchos/{model}/string		
Method		
POST		
Description		
This method is used to upload Fluchos data. It receives a string text with the format for Talonadora or Rotostir machines and inserts the data in the RECLAIM Repository		
URL Params		
Required:		
model	string	*This is the Fluchos data type to upload* *Values: “Talonadora” or “Rotostir”*
Optional:		
Data Params		
Required:		
request_body	string	*This is the actual data to be uploaded in the RECLAIM Repository.* *See the “Fluchos Talonadora Example object” below for reference*
Optional:		
Success response		

<table>
<tr>
<td>

200

Content:

```
[
  {
    "timestamp": "2021-09-13T09:04:15.000",
    "batchNo": "20220915-1",
    "iceTimeIn":                   "2022-09-
15T01:41:17.093383",
    "streamName": "Talonadora",
    "modelId": "Talonadora",
    "SN": "98356",
    "name": "fluchos",
    "Foco_Caliente_IZQ": 121,
    "Foco_Frio_IZQ": -4.800000000000001,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null
  },
  {
    "timestamp": "2021-09-13T09:04:19.000",
    "batchNo": "20220915-1",
    "iceTimeIn":                   "2022-09-
15T01:41:17.093744",
    "streamName": "Talonadora",
    "modelId": "Talonadora",
    "SN": "98356",
    "name": "fluchos",
    "Foco_Caliente_IZQ": 122,
    "Foco_Frio_IZQ": -4.9,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null
  },
  {
    "timestamp": "2021-09-13T09:04:21.000",
    "batchNo": "20220915-1",
    "iceTimeIn":                   "2022-09-
15T01:41:17.093962",
    "streamName": "Talonadora",
    "modelId": "Talonadora",
    "SN": "98356",
    "name": "fluchos",
    "Foco_Caliente_IZQ": 122,
    "Foco_Frio_IZQ": -5,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null
  },
]
```

</td>
<td>

Request was successful

</td>
</tr>
<tr><td colspan="2">

Error response

</td></tr>
<tr><td>400</td><td><em>Bad Request</em></td></tr>
<tr><td>403</td><td><em>Not authorized</em></td></tr>
<tr><td>422</td><td><em>Validation Error</em></td></tr>
<tr><td>500</td><td><em>Generic Error</em></td></tr>
<tr><td colspan="2">

Sample call

</td></tr>
<tr><td colspan="2">

```
curl -X 'POST' \
```

</td></tr>
</table>

```
  'https://iceberg.icelab.cloud/docbase/fluchos/Talonadora/string' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: application/json' \
  -d '{"SN": "98356", "name": "fluchos", "header": {"SamplingPeriod":
"1", "Timestamp": "2021-09-13T09:09:13.898", "recordCount": 103,
"columns": {"0": {"id": "0", "name": "Foco_Caliente_IZQ", "dataType":
"NUMBER", "format": "double"}, "1": {"id": "1", "name": "Foco_Frio_IZQ",
"dataType": "NUMBER", "format": "double"}, "2": {"id": "2", "name":
"Din1", "dataType": "NUMBER", "format": "double"}, "3": {"id": "3",
"name": "Din2", "dataType": "NUMBER", "format": "double"}, "4": {"id":
"4", "name": "Pulsador foco caliente", "dataType": "NUMBER", "format":
"double"}, "5": {"id": "5", "name": "Pulsador foco frío", "dataType":
"NUMBER", "format": "double"}}}, "data": [{…}]'
```

Notes

Example file:

20210913-090914_Talonadora.json

Method		Get Fluchos data
URL		
/fluchos/{model}		
Method		
GET		
Description		
This method is used to get Fluchos data present in the RECLAIM repository. It receives a string text with the data to retrieve: Talonadora or Rotostir		
URL Params		
Required:		
Model	string	*This is the Fluchos data type* *Values: “Talonadora” or “Rotostir”*
Optional:		
page_number	integer	*Paging parameter. Page Number to get data based on Page size* *Default value: 1*
page_size	integer	*Paging parameter. Number of records to be returned per page* *Default value: 100*
order_by	string	*Paging parameter. Column used to sort the data* *Default value: timestamp*

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

loading_date_start	date	*Search parameter. Initial date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
loading_date_end	date	*Search parameter. Final date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_start	date	*Search parameter: Initial date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_end	date	*Search parameter: Final date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
custom_search	string	*Search parameter: This is a JSON object that contains search parameters for the data uploaded. All the data that matches the object properties and values will be returned (AND operation)* *Date format: JSON object.* *Default value: null vale.* *Example:* *{* *"property": 100,* *"property_2": "value"* *}*
column	array[string]	*A list of names of the columns that will be retrieved in the JSON objects.* *Example: column=timestamp&column=iceTimeIn&column=column_name_1&column=column_name_2*

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884

Data Params		
Required:		
Optional:		
Success response		
200		*Request was successful*

Content:

```
[
  {
    "timestamp": "2022-04-22T05:13:00.000",
    "batchNo": "20220422-1",
    "iceTimeIn":                   "2022-04-
22T05:15:39.615766",
    "streamName": "Talonadora",
    "modelId": "Talonadora",
    "SN": 98356,
    "name": "fluchos",
    "Foco_Caliente_IZQ": 17,
    "Foco_Frio_IZQ": 152,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null,
    "timestamp2": "2022-04-22T05:13:00"
  },
  {
    "timestamp": "2022-04-22T05:13:03.000",
    "batchNo": "20220422-1",
    "iceTimeIn":                   "2022-04-
22T05:15:39.616180",
    "streamName": "Talonadora",
    "modelId": "Talonadora",
    "SN": 98356,
    "name": "fluchos",
    "Foco_Caliente_IZQ": 17,
    "Foco_Frio_IZQ": 151,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null,
    "timestamp2": "2022-04-22T05:13:03"
  },
  {
    "timestamp": "2022-04-22T05:13:06.000",
    "batchNo": "20220422-1",
    "iceTimeIn":                   "2022-04-
22T05:15:39.616411",
    "streamName": "Talonadora",
    "modelId": "Talonadora",
    "SN": 98356,
    "name": "fluchos",
    "Foco_Caliente_IZQ": 17,
    "Foco_Frio_IZQ": 149,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null,
    "timestamp2": "2022-04-22T05:13:06"
  }
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

]	
Error response	
400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*
Sample call	

```
curl -X 'GET' \
'https://iceberg.icelab.cloud/docbase/fluchos/Talonadora?page_number=1&page_size=3&order_by=timestamp&data_date_start=2022-04-01' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
```

Notes	

4.2.3.3 HWH Pilot - REST API to upload data files to the RECLAIM Repository

Method	Upload welding sensors data file	
URL		
/hwh/file		
Method		
POST		
Description		
This method is used to upload HWH sensors data. It receives a file with the format for “welding sensors” and inserts the data in the RECLAIM Repository		
URL Params		
Required:		
Optional:		
Data Params		
Required:		
model	string	*A name to identify the data that is being uploaded to the RECLAIM Repository* *Values: “welding_sensors”*
json_file	string ($binary)	*A JSON file with the HWH welding sensors file format.*
Optional:		
Success response		
200	*Request was successful*	

Content:

```
{
  "result": "File Successfully uploaded.",
  "item_list": [
    {
      "timeStart": 1652573607.9901192,
      "timeEnd": 1652573622.4722629,
      "environmentT": 330.65838512216897,
      "motorBearingT": 334.0230672720759,
      "spindleBearingT":
359.60936095481463,
      "counter": 15207,
      "sdIntensity": 1.390561662782471,
      "total_time": 14.482143640518188,
      "production": 1449,
      "timestamp":                 "2022-05-
15T00:13:27.990119",
      "iceTimeIn":                 "2022-09-
15T16:57:29.393943",
      "modelId": "welding_sensors"
    }
  ]
}
```

Error response	
400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'POST' \
  'https://iceberg.icelab.cloud/docbase/hwh/file' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: multipart/form-data' \
  -F 'model=welding_sensors' \
  -F 'json_file=@welding_sensors_20220512-
073449.json;type=application/json'
```

Notes

File used for testing proposes.

welding_sensors_2
0220512-073449.jso

Method	Upload welding sensors data JSON
URL	
/hwh/string	
Method	
POST	
Description	

This method is used to upload HWH sensors data. It receives a JSON text with the format for "welding sensors" data and inserts the data in the RECLAIM Repository		
URL Params		
Required:		
Optional:		
Data Params		
Required:		
model	string	*A name to identify the data that is being uploaded to the RECLAIM Repository* *Values: "welding_sensors"*
json_dict	string	*A JSON object with the HWH welding sensors file format.*
Optional:		
Success response		
200 Content:		*Request was successful*

```
{
  "result": "JSON Successfully uploaded.",
  "item_list": [
    {
      "timeStart": 1652573625.393203,
      "timeEnd": 1652573639.69794,
      "environmentT": 334.18325810710456,
      "motorBearingT": 347.8666092427973,
      "spindleBearingT": 362.7043930483701,
      "counter": 15208,
      "sdIntensity": 1.3719549948562872,
      "total_time": 14.304737091064453,
      "production": 1431,
      "timestamp":                 "2022-05-
15T00:13:45.393203",
      "iceTimeIn":                 "2022-09-
15T17:13:35.804843",
      "modelId": "welding_sensors"
    }
  ]
}
```

Error response		
400		*Bad Request*
403		*Not authorized*
422		*Validation Error*
500		*Generic Error*
Sample call		

```
curl -X 'POST' \
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

```
  'https://iceberg.icelab.cloud/docbase/hwh/string' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: application/json' \
  -d '{
  "model": "welding_sensors",
  "json_dict": [{"timeStart": 1652573625.393203, "timeEnd":
1652573639.69794, "environmentT": 334.18325810710456, "motorBearingT":
347.8666092427973, "spindleBearingT": 362.7043930483701, "counter":
15208, "sdIntensity": 1.3719549948562872, "times": [1652573625.393203,
1652573625.403203, 1652573625.413203, …], "angularVelocity": [0.0, 0.0,
0.0, …], "force": [0.0, 0.0, 0.0, …], "displacement": [0.0, 0.0, 0.0,
…]}]
}'
```

Notes

Sample file:

welding_sensors_20220512-073459.json

Method		Get welding data
URL		
/hwh/{model}		
Method		
GET		
Description		
This method is used to get the HWH data present in the RECLAIM repository. It receives a string text with the data to retrieve: “single_welding”, “welding_load_data” or “welding_sensors”		
URL Params		
Required:		
Model	string	*This is the HWH data type* *Values: “single_welding”, “welding_load_data” or “welding_sensors”*
Optional:		
page_number	integer	*Paging parameter. Page Number to get data based on Page size* *Default value: 1*
page_size	integer	*Paging parameter. Number of records to be returned per page* *Default value: 100*
order_by	string	*Paging parameter. Column used to sort the data*

		Default value: timestamp
loading_date_start	date	*Search parameter. Initial date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
loading_date_end	date	*Search parameter. Final date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_start	date	*Search parameter: Initial date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_end	date	*Search parameter: Final date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
custom_search	string	*Search parameter: This is a JSON object that contains search parameters for the data uploaded. All the data that matches the object properties and values will be returned (AND operation)* *Date format: JSON object.* *Default value: null value.* *Example:* *{* *"property": 100,* *"property_2": "value"* *}*
column	array[string]	*A list of names of the columns that will be retrieved in the JSON objects.* *Example: column=timestamp&column=iceTimeIn&col*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

<table>
<tr><td></td><td></td><td><i>umn=column_name_1&column=column_name_2</i></td></tr>
<tr><td colspan="3"><b>Data Params</b></td></tr>
<tr><td colspan="3">Required:</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Optional:</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3"><b>Success response</b></td></tr>
<tr><td colspan="2">200
Content:<pre>
[
  {
    "timeStart": 1653004929.1665635,
    "timeEnd": 1653004935.7082438,
    "environmentT": 317.8480899620895,
    "motorBearingT": 345.3456021892771,
    "spindleBearingT": 371.3022337234572,
    "counter": 22160,
    "sdIntensity": 1.3118183183461287,
    "total_time": 6.541680335998535,
    "production": 655,
    "timestamp":   "2022-05-
20T00:02:09.166564",
    "iceTimeIn":   "2022-05-
16T03:50:17.627852",
    "modelId": "welding_sensors",
    "times":   "[1653004929.1665635,
1653004929.1765635, 1653004929.1865635,…]",
    "angularVelocity": "[0.0, 0.0, 0.0, …]",
    "force": "[0.0, 0.0, 0.0, …]",
    "displacement": "[0.0, 0.0, 0.0, …]",
    "timestamp2": "2022-05-20T00:02:09"
  }
]
</pre></td><td><i>Request was successful</i></td></tr>
<tr><td colspan="3"><b>Error response</b></td></tr>
<tr><td colspan="2">400</td><td><i>Bad Request</i></td></tr>
<tr><td colspan="2">403</td><td><i>Not authorized</i></td></tr>
<tr><td colspan="2">422</td><td><i>Validation Error</i></td></tr>
<tr><td colspan="2">500</td><td><i>Generic Error</i></td></tr>
<tr><td colspan="3"><b>Sample call</b></td></tr>
<tr><td colspan="3"><pre>
curl -X 'GET' \
'https://iceberg.icelab.cloud/docbase/hwh/welding_sensors?page_number=1&p
age_size=1&order_by=timestamp&data_date_start=2022-05-20' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
</pre></td></tr>
<tr><td colspan="3"><b>Notes</b></td></tr>
<tr><td colspan="3"></td></tr>
</table>

4.2.3.4 Generic Access - REST API to upload and get data from RECLAIM Repository

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Method		Get data from DryIce
URL		
/dryice/data/{dryicekey}		
Method		
GET		
Description		
This method will get the data from the RECLAIM Repository. This is a method that implements paging and searching		
URL Params		
Required:		
dryicekey	string	*This is the name of the tables from where the data is going to be obtained*
Optional:		
page_number	integer	*Paging parameter. Page Number to get data based on Page size* *Default value: 1*
page_size	integer	*Paging parameter. Number of records to be returned per page* *Default value: 100*
order_by	string	*Paging parameter. Column used to sort the data* *Default value: timestamp*
loading_date_start	date	*Search parameter. Initial date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
loading_date_end	date	*Search parameter. Final date to search when the data was uploaded to the RECLAIM Repository (Column iceTimeIn)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*
data_date_start	date	*Search parameter: Initial date to search when the data was created. Actual data timestamp. (Column timestamp)* *Date format: YYYY-MM-DD.* *Default value: null value.* *Example: 2021-11-11*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

<table>
<tr><td>data_date_end</td><td>date</td><td><i>Search parameter: Final date to search when the data was created. Actual data timestamp. (Column timestamp)</i>
<i>Date format: YYYY-MM-DD.</i>
<i>Default value: null value.</i>
<i>Example: 2021-11-11</i></td></tr>
<tr><td>custom_search</td><td>string</td><td><i>Search parameter: This is a JSON object that contains search parameters for the data uploaded. All the data that matches the object properties and values will be returned (AND operation)</i>
<i>Date format: JSON object.</i>
<i>Default value: null value.</i>
<i>Example:</i>
<i>{</i>
<i>"property": 100,</i>
<i>"property_2": "value"</i>
<i>}</i></td></tr>
<tr><td>column</td><td>array[string]</td><td><i>A list of names of the columns that will be retrieved in the JSON objects.</i>
<i>Example: column=timestamp&column=iceTimeIn&column=column_name_1&column=column_name_2</i></td></tr>
<tr><td colspan="3"><b>Data Params</b></td></tr>
<tr><td colspan="3">Required:</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Optional:</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3"><b>Success response</b></td></tr>
<tr><td colspan="2">200
Content:<pre>
[
  {
    "SN": 98356,
    "name": "fluchos",
    "Foco_Caliente_IZQ": null,
    "Foco_Frio_IZQ": null,
    "Din1": 0,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null,
    "timestamp": "2021-09-13T08:49:05.000"
  },
  {
    "SN": 98356,
    "name": "fluchos",
</pre></td><td><i>Request was successful</i></td></tr>
</table>

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

```
    "Foco_Caliente_IZQ": 122,
    "Foco_Frio_IZQ": -5,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null,
    "timestamp": "2021-09-13T09:04:21.000"
  },
  {
    "SN": 98356,
    "name": "fluchos",
    "Foco_Caliente_IZQ": 122,
    "Foco_Frio_IZQ": -6,
    "Din1": null,
    "Din2": null,
    "Pulsadorfococaliente": null,
    "Pulsadorfocofrio": null,
    "timestamp": "2021-09-13T09:04:42.000"
  }
]
```

Error response	
400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'GET' \

'https://iceberg.icelab.cloud/docbase/dryice/data/Talonadora?page_number=1&page_size=3&order_by=timestamp&column=SN&column=name&column=Foco_Caliente_IZQ&column=Foco_Frio_IZQ&column=Din1&column=Din2&column=Pulsadorfococaliente&column=Pulsadorfocofrio&column=timestamp' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
```

Notes

Method		Add Data to DryIce
URL		
/dryice/data/{dryicekey}		
Method		
POST		
Description		
This method will insert the data sent into the RECLAIM Repository		
URL Params		
Required:		
dryicekey	string	*This is the name used to store the data. This name will be used to create a new table where all the data will be stored*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

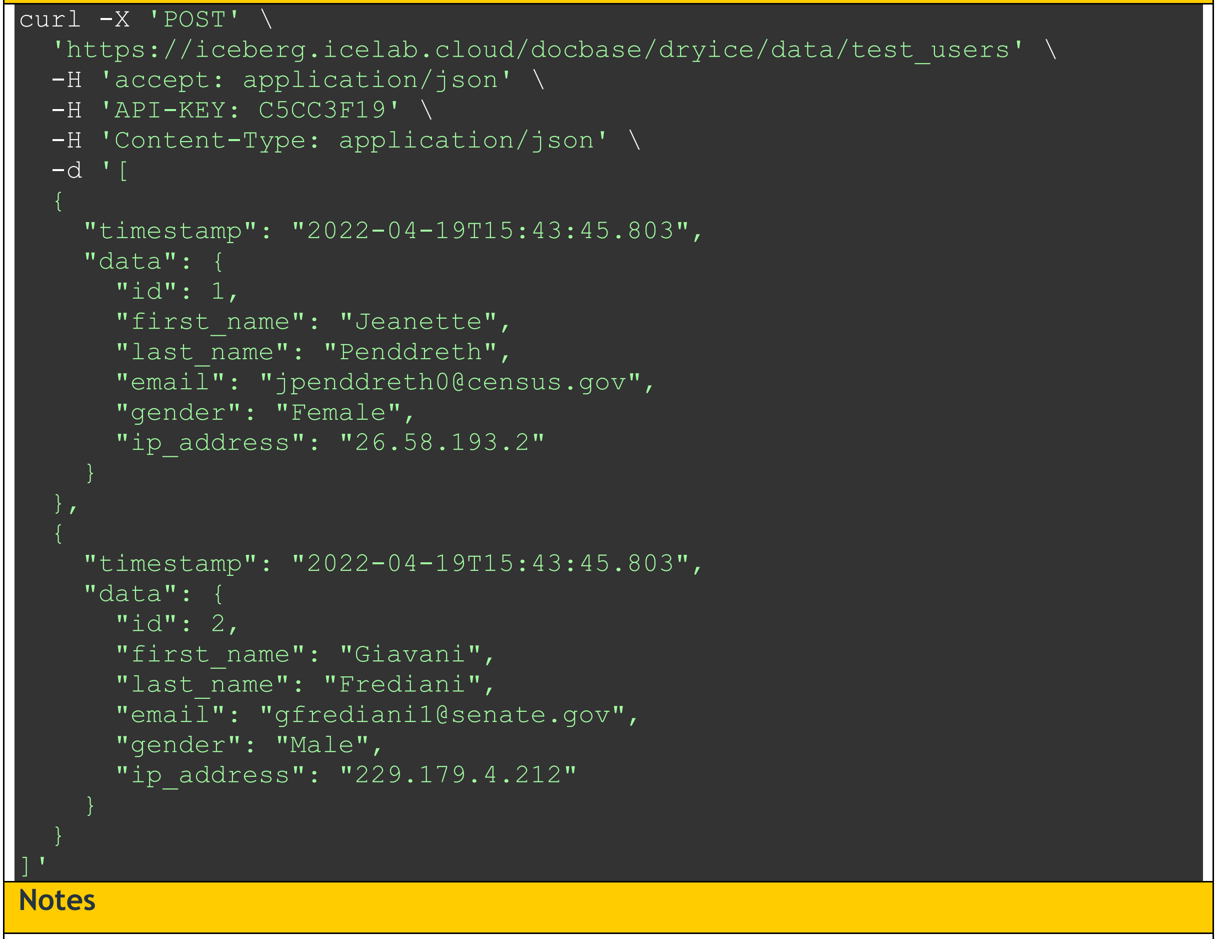
Optional:		
Data Params		
Required:		
request_body	string	*An array of JSON objects with the following format.* • timestamp: This property will have the timestamp when the data was created. It must use the standard ISO format. • data: This property is a JSON object with the values to be inserted into the RECLAIM Repository. *it does not support inner objects
Optional:		
Success response		

200
Content:
`[`
`{`
`"id": 1,`
`"first_name": "Jeanette",`
`"last_name": "Penddreth",`
`"email": "jpenddreth0@census.gov",`
`"gender": "Female",`
`"ip_address": "26.58.193.2",`
`"timestamp": "2022-04-19T15:43:45.803000",`
`"iceTimeIn": "2022-09-15T15:48:17.195778",`
`"modelId": "test_users",`
`"streamName": "test_users"`
`},`
`{`
`"id": 2,`
`"first_name": "Giavani",`
`"last_name": "Frediani",`
`"email": "gfrediani1@senate.gov",`
`"gender": "Male",`
`"ip_address": "229.179.4.212",`
`"timestamp": "2022-04-19T15:43:45.803000",`
`"iceTimeIn": "2022-09-15T15:48:17.195778",`
`"modelId": "test_users",`
`"streamName": "test_users"`
`}`
`]`
		Request was successful

Error response		
400		*Bad Request*
403		*Not authorized*

422	*Validation Error*
500	*Generic Error*

Sample call



```
curl -X 'POST' \
  'https://iceberg.icelab.cloud/docbase/dryice/data/test_users' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: application/json' \
  -d '[
  {
    "timestamp": "2022-04-19T15:43:45.803",
    "data": {
      "id": 1,
      "first_name": "Jeanette",
      "last_name": "Penddreth",
      "email": "jpenddreth0@census.gov",
      "gender": "Female",
      "ip_address": "26.58.193.2"
    }
  },
  {
    "timestamp": "2022-04-19T15:43:45.803",
    "data": {
      "id": 2,
      "first_name": "Giavani",
      "last_name": "Frediani",
      "email": "gfrediani1@senate.gov",
      "gender": "Male",
      "ip_address": "229.179.4.212"
    }
  }
]'
```

Notes

Method		Get latest data from DryIce
URL		
dryice/latest/{dryicekey}		
Method		
GET		
Description		
This method will get the latest data from the RECLAIM Repository.		
URL Params		
Required:		
dryicekey	string	*This is the name of the tables from where the data is going to be obtained*
Optional:		
page_number	integer	*Paging parameter. Page Number to get data based on Page size*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

		Default value: 1
page_size	integer	*Paging parameter. Number of records to be returned per page* *Default value: 100*
time_column	string	*Name of the column for the Timestamp of the data.* *Default value: timestamp2*
search_data	string	*Search parameter: This is a JSON object that contains search parameters for the data uploaded. All the data that matches the object properties and values will be returned (AND operation)* *Date format: JSON object.* *Default value: null value.* *Example:* *{* *"property": 100,* *"property_2": "value"* *}*
column	array[string]	*A list of names of the columns that will be retrieved in the JSON objects.* *Example: column=timestamp&column=iceTimeIn&column=column_name_1&column=column_name_2*
Data Params		
Required:		
Optional:		
Success response		

200
Content:
`[`
`{`
`"mtbf": 57600,`
`"missionFailureProb": 0.743,`
`"failureProb": 0.6,`
`"missionduration": 86400,`
`"kind": "SYSTEM",`
`"systemId": 2,`
`"timestamp": "2022-07-14T13:32:25.936000",`
`"iceTimeIn": "2022-07-14T13:36:38.777175",`
`"modelId": "supsirat",`
		Request was successful

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

```
    "streamName": "supsirat",
    "timestamp2": "2022-07-14T13:32:25"
  },
  {
    "mtbf": 0,
    "missionFailureProb": 0,
    "failureProb": 0,
    "missionduration": 0,
    "kind": "",
    "systemId": 2,
    "timestamp": "2020-10-22T08:03:59",
    "iceTimeIn":                  "2022-07-
20T05:52:25.862265",
    "modelId": "supsirat",
    "streamName": "supsirat",
    "timestamp2": "2020-10-22T08:03:59"
  }
]
```

Error response	
400	*Bad Request*
403	*Not authorized*
422	*Validation Error*
500	*Generic Error*

Sample call

```
curl -X 'GET' \
'https://iceberg.icelab.cloud/docbase/dryice/latest/supsirat?page_number=
1&page_size=100&time_column=timestamp2&search_data=%7B%22systemId%22%3A%2
02%7D' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
```

Notes

Method		Get table name from DryIce
URL		
/dryice/tables?dryicekey		
Method		
GET		
Description		
This method will return the list of available tables in the RECLAIM repository		
URL Params		
Required:		
dryicekey	string	*This parameter is used to filter the tables returned. It will return all the tables that have the key specified* *Default value: empty string*
Optional:		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Data Params		
Required:		
Optional:		
Success response		
200 Content: ["hwh_welding_load", "hwh_welding_sensors", "single_welding", "welding", "welding_load_data", "welding_sensors"]		*Request was successful*
Error response		
400		*Bad Request*
403		*Not authorized*
422		*Validation Error*
500		*Generic Error*
Sample call		

```
curl -X 'GET' \
  'https://iceberg.icelab.cloud/docbase/dryice/tables?drydicekey=welding'
\
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19'
```

Notes

4.2.3.5 cost-model - REST API to upload Cost Model excel file to the RECLAIM Repository.

Method	Upload cost-model Excel file
URL	
/cost-model/upload	
Method	
POST	
Description	
This method is used to upload an excel file with the Cost Model format in the RECLAIM Repository.	
URL Params	
Required:	

Optional:		
Data Params		
Required:		
output_dsf_key	String	*The table name where the dsf output data will be stored.* *Default value: output_for_dsf*
output_user_key	String	*The table name where the user output data will be stored.* *Default value: output_for_user*
excel_file	string($binary)	*The Cost Modelling file that we want to upload into DryIce.*
Optional:		
Success response		

200
Content:
{
  "result": "File Successfully uploaded.",
  "LoadId":   "99cbc4a9-ada6-441e-8a11-ac10c7b32432"
}
		Request was successful

Error response		
400		*Bad Request*
403		*Not authorized*
422		*Validation Error*
500		*Generic Error*
Sample call		

```
curl -X 'POST' \
  'https://iceberg.icelab.cloud/docbase/cost-model/upload' \
  -H 'accept: application/json' \
  -H 'API-KEY: C5CC3F19' \
  -H 'Content-Type: multipart/form-data' \
  -F 'output_dsf_key=cost_modeling_dsf' \
  -F 'output_user_key=cost_modeling_user' \
  -F 'excel_file=@HWH RECLAIM T4.3 Cost Model
v1.7.xlsx;type=application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet'
```

Notes
File used for testing propuses.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

HWH RECLAIM T4.3
Cost Model v1.7.xlsx

5 Conclusion

Machine data is being produced in greater quantities and in real time. This data not only tells an operator when something is wrong with the process in the now, it also can be used to predict failure before it occurs, protecting the asset (the machine) and the end product (the part). Traditional methods of preventative maintenance can often lead to unnecessary machine inspections. With the approach proposed by RECLAIM, it is possible to assess and monitor the operation of critical systems in real time and identify potential wear or faults in moving components as they develop. This allows for a more convenient scheduling of service or repair intervals.

Malfunctioning machinery causes a huge amount of lost productivity and revenue for manufacturing companies worldwide. This is of course a major concern, but also something that in many cases is avoidable.

A well-developed monitoring and assessment of machines allows the detection of machinery failure signs well before malfunction, helping the prioritization of maintenance tasks and addressing issues – before they lead to breakdown. This proactive approach of course reduces maintenance costs and maximizes production output.

The assessment of the machinery leads to a machine profiling activity which ends up with the formalisation and calculation of the RECLAIM indexes, namely health, performance, and production. These three indexes will then be used within the RECLAIM DSF so that the manufacturing industry will have a one-stop point where to see the status of the machinery and take the appropriate informed decision.

The main goals of optimization of these indices are the reduction of costs, the increase of safety (e.g. safety of batteries in electric cars), or the unleash of new capabilities and development of business models. These new business models are related to build the cost models for the replacement or refurbishment of machine parts as well as the acquisition of new knowledge regarding the machine lifecycle, e.g. machine wear/reliability. In the former case, the business models are oriented on spare/efficient usage of equipment or in keeping health resources (reuse of physical equipment), while in the latter case the business models are related to the diagnosis and prevention of failures within production lines.

6 References

[1] Martin K.F. (1994). “A review by discussion of condition monitoring and fault diagnosis in machine tools”, Int. Journal of Tools Manufacturing, Vol.34, No.4, pp.527-551.

[2] Mahantesh, Nadakatti & Aditya, Parida & Kumar, Uday. (2013). “Integrated machine health monitoring: A knowledge based approach”, International Journal of System Assurance Engineering and Management. 5. 10.1007/s13198-013-0178-1.

[3] Tajiri M., Gotoh F. (1992). “TPM Implementation: A Japanese Approach”, McGraw-Hill, New York.

[4] Mirko S., Jelena J., Zdravko K., Aleksandar V. (2009). “Basic Quality Tools in Continuous Improvement Process”, Journal of Mechanical Engineering, Volume 55, Issue 5, pp 1-4.

[5] Bhadury B. (2000). “Management of Productivity Through TPM”, Volume 41, Issue 2, pp. 240-51.

[6] Heba A., Qiu X. (2007). “A Model For Assessing Cost Effectiveness of Applying Lean Tools”, School of Technology and Design, Växjö University, Volume 46, Issue 73, pp.

[7] Sondalini, M. “Plant and Equipment Health and Reliability Measurement using the Plant Health Index”. On-line (https://www.lifetime-reliability.com/cms/machinery-health-measurement), last accessed: July 2020.

[8] Industry Forum. “What is Autonomous Maintenance?” On-line (https://www.industryforum.co.uk/resources/articles/autonomous-maintenance), last accessed: July 2020.

[9] UNI CEN/TS 17385:2019 - Method for condition assessment of immobile constructed assets.

[10] UNI EN 15341:2019 - Maintenance - Maintenance Key Performance Indicators.

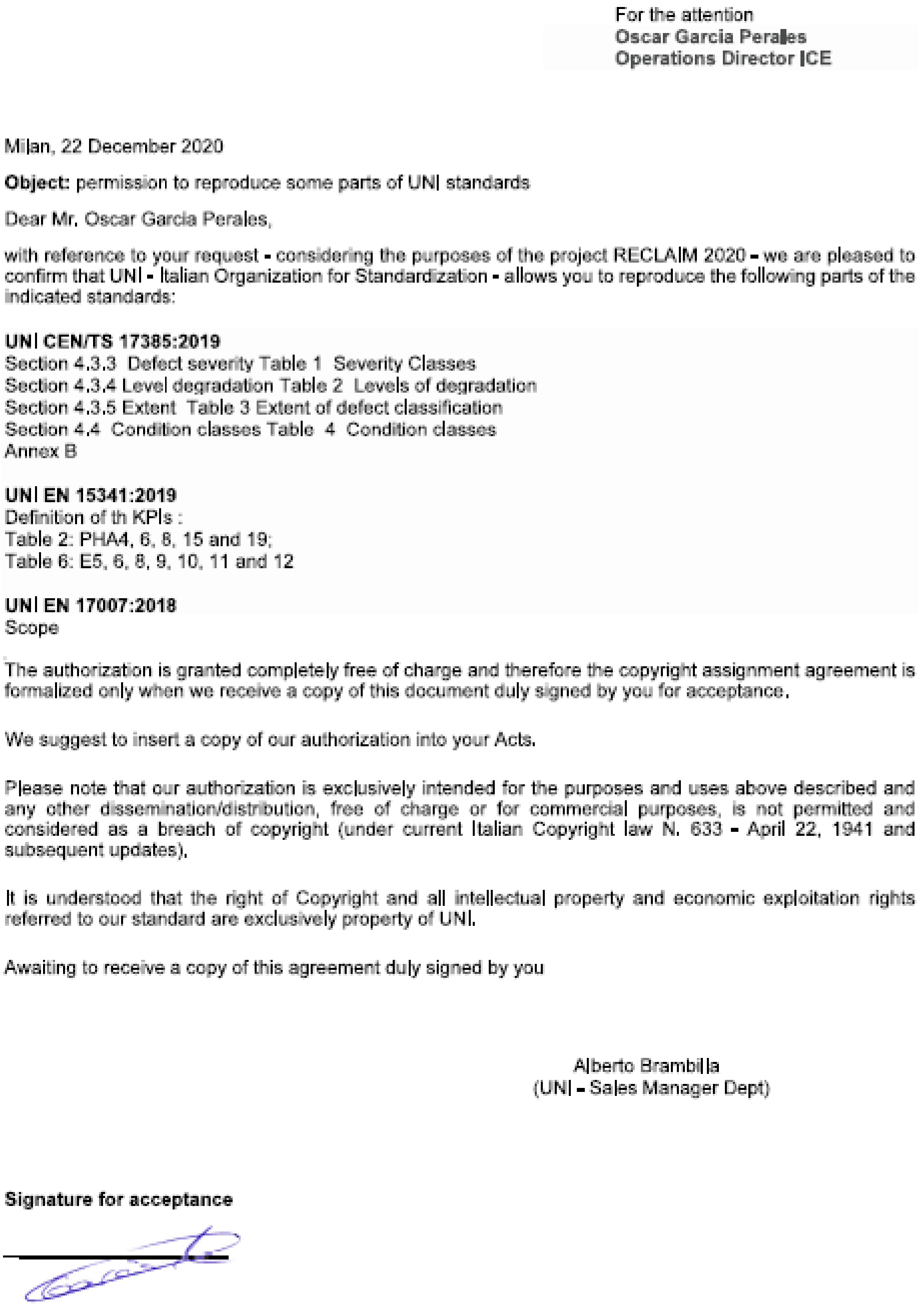
[11] UNI EN 17007:2018 - Maintenance process and associated indicators.

Annex A. Authorization from UNI

The document below grants access to the deliverable lead to cite and reproduce partial content of the standards indicated below:

- UNI CEN/TS 17385:2019 - Method for condition assessment of immobile constructed assets.
- UNI EN 15341:2019 - Maintenance - Maintenance Key Performance Indicators.
- UNI EN 17007:2018 - Maintenance process and associated indicators.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884



For the attention
Oscar Garcia Perales
Operations Director ICE

Milan, 22 December 2020

Object: permission to reproduce some parts of UNI standards

Dear Mr. Oscar Garcia Perales,

with reference to your request - considering the purposes of the project RECLAIM 2020 - we are pleased to confirm that UNI - Italian Organization for Standardization - allows you to reproduce the following parts of the indicated standards:

UNI CEN/TS 17385:2019
Section 4.3.3 Defect severity Table 1 Severity Classes
Section 4.3.4 Level degradation Table 2 Levels of degradation
Section 4.3.5 Extent Table 3 Extent of defect classification
Section 4.4 Condition classes Table 4 Condition classes
Annex B

UNI EN 15341:2019
Definition of th KPIs :
Table 2: PHA4, 6, 8, 15 and 19;
Table 6: E5, 6, 8, 9, 10, 11 and 12

UNI EN 17007:2018
Scope

The authorization is granted completely free of charge and therefore the copyright assignment agreement is formalized only when we receive a copy of this document duly signed by you for acceptance.

We suggest to insert a copy of our authorization into your Acts.

Please note that our authorization is exclusively intended for the purposes and uses above described and any other dissemination/distribution, free of charge or for commercial purposes, is not permitted and considered as a breach of copyright (under current Italian Copyright law N. 633 - April 22, 1941 and subsequent updates).

It is understood that the right of Copyright and all intellectual property and economic exploitation rights referred to our standard are exclusively property of UNI.

Awaiting to receive a copy of this agreement duly signed by you

Alberto Brambilla
(UNI - Sales Manager Dept)

Signature for acceptance

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Annex B. Zorluteks Bleaching Operation

Raw cotton, like all-natural fibres, has some natural colouring matter, which confers a yellowish-brown colour to the fibre. The purpose of bleaching is to remove this colouring material and to confer a white appearance to the fibre.

The following are the objectives of bleaching:

- The main objectives of bleaching are to get a sufficiently high and uniform degree of whiteness in the textile materials.
- To get a high and uniform absorptivity in the textile materials.
- Bleaching agent occur some damage to the textile materials. So, bleaching must be accompanied with minimum fibre damage.
- To preserve a good user and technological properties of the textile materials.
- The process must be ecologically and financially sensible.
- To accelerate the next dyeing process.

In Zorluteks' production plant, hydrogen peroxide (H_2O_2) is used as a bleaching agent. The bleaching bath is composed of hydrogen peroxide as a bleaching agent, an alkali activator (caustic soda), stabilizer, sequestering agent, and wetting agent.

Though hydrogen peroxide is stable in acidic medium, but bleaching occurs by the addition of alkali or by increasing temperature. Hydrogen peroxide liberates per hydroxyl ion (HO_2^-) in aqueous medium and chemically behaves like a weak dibasic acid. The per-hydroxyl is highly unstable and in the presence of oxidizable substance (coloured impurities in cotton), it is decomposed and thus bleaching action takes place as in the following equation.

$$H_2O_2 + OH^- \leftrightarrow H_2O + HO_2^-$$

Parameters of Bleaching Process

The bleaching of textile fabric with hydrogen peroxide is dependent on many aspects such as pH, temperature, time, etc.

Effect of chemical concentrations: Caustic soda, hydrogen peroxide, stabilizer, sequestering agent, and wetting agent are used for bleaching operation in Zorluteks. Chemical concentration affects pH of the bath which provides to reach appropriate pH value for bleaching reaction. The separation of hydrogen peroxide depends on the amount of alkali in the bath. Hydrogen peroxide is a weak acid. The activation of peroxide occurs at pH 10-12, so hydrogen peroxide is not an active bleaching agent for cellulose fibres in acidic and neutral environments.

Effect of temperature of steamer: Hydrogen peroxide solution at low temperature is very stable and reacts very slowly; as the temperature rises, the stability of hydrogen peroxide decreases and reaction takes place. The bleaching process usually takes place at around 100˚C in steamer.

Effect of temperature of washing baths: At the inlet and outlet of the bleaching machine washing baths located. Temperature of these baths should be 85-95 ˚C.

Effect of steaming time: Steaming time is one of the important parameters of bleaching process. The time required for bleaching with hydrogen peroxide depends on the temperature, the type of fibre and the equipment used. In general, the duration of bleaching is inversely proportional to the temperature of steamer. In Zorluteks, steaming time is set at 18 minutes.

Pick-up value: The amount of finishing solution or emulsion applied is referred to as the *% wet pick-up* which is expressed as:

$$\%pick\_up = \frac{(wt\_of\_padded\_fabric - wt\_of\_dry\_fabric)\ x\ 100}{wt\_of\_dry\_fabric}$$

Pick up value can be arranged by changing pressure of squeeze roller. In our process, pick up value is %130-140.

Bleaching Machine

Following Figure 9 demonstrates continuous bleaching machine (Küsters-1997) and the recipe that is used during the bleaching operation in Zorluteks' production plant.

BLEACHING RECIPE

1. Caustic
2. Hydrogen peroxide
3. Stabilizer
4. Sequestering agent
5. Wetting agent

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Roller pressure before bleaching chemical trough: Left: 4.2 bar, Mid: 2.9bar, Right:4.2bar

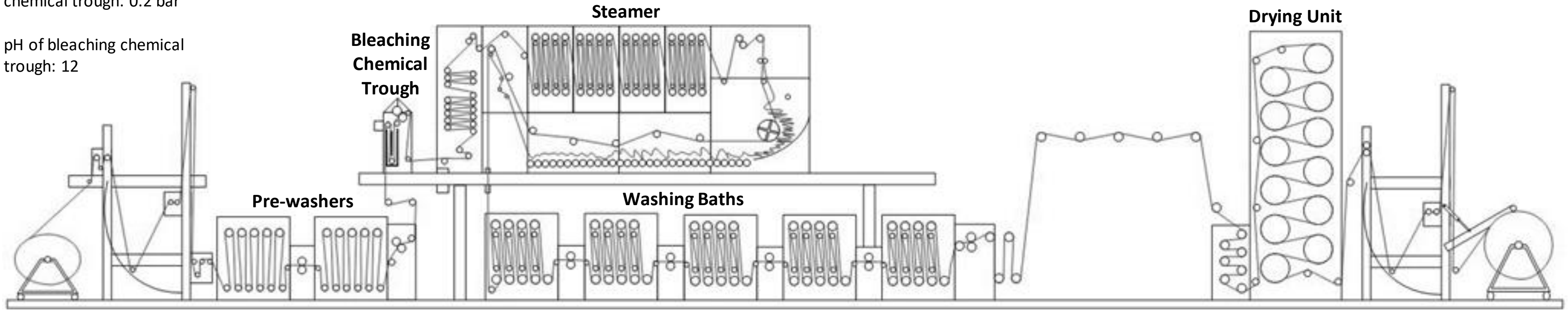
Roller pressure after chemical trough: 0.2 bar

pH of bleaching chemical trough: 12

Temperature of Steamer : 98 °C

Temperature of Steam : 102 °C

Steaming Time: 18 min



Temperature of washing baths:	95 °C	85 °C	95 °C	95 °C	95 °C	85 °C	25-30 °C	Temprature drying unit:	110-130 °C
pH of washing baths:	8	8	11-12	8	8	8	5	Moisture in outlet fabric:	5-8 %

Figure 9: Continuous Bleaching Machine (Küsters-1997)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N˚869884

After finishing desizing process, fabric comes to the bleaching machine. First step is pre-washing to wash off the starch and other impurities from desized fabric by using hot water. There are 2 pre-washers at the inlet of the machine. Then fabric moves to bleaching chemical through which is the place that bleaching chemicals based on the recipe are applied uniformly on the fabric. After that, bleaching reaction takes place in steamer at 102°C and in 18 minutes. The next step is to remove bleaching chemicals on the fabric in washing baths. Although the first four washing bats contain hot water, the last one is filled with water at room temperature. The reason behind is that neutralization is occurred with acetic acid. Finally, fabric is dried and ready for the following processes. All related parameters are given in Figure 9.

Whiteness Index (WI)

Whiteness is defined as a measure of how closely a surface matches the properties of a perfect reflecting diffuser, i.e. an ideal reflecting surface that neither absorbs nor transmits light but reflects it at equal intensities in all directions. For the purposes of this standard, the colour of such a surface is known as preferred white.

Whiteness Index is a measure which correlates the visual ratings of whiteness for certain white and near-white surfaces. There is a number of different indices available. In Zorluteks, Berger Whiteness index is used. At the end of the bleaching process, whiteness of cotton fabric is checked.

Berger: This whiteness index is specified for illuminant C and 2-degree observer functions only. However, the equation is commonly used with other illuminants and observer functions, therefore the value shown will depend on the primary illuminant and the observer function you have chosen. The formula is:

$$WI = Y + 0{,}3018Z - 3.831X$$

In this formula X, Y, Z values represent 3 different colours (red/green/blue) on a 3D Space. 100 Berger stands for perfect whiteness which is located centre of the XYZ tristimulus given in the Figure 10. In Zorluteks, the aim is that produce fabric with 70-75 Berger at the end of the bleaching process. Measurements for Berger index is carried out with the help of “Macbeth Color-Eye 7000” spectrophotometry (Figure 11). Data taken from spectrophotometry are analysed by an application named “Protect TexSilver” to get a whiteness degree.

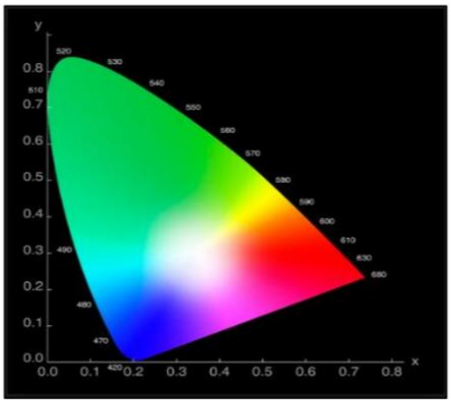


Figure 10: CIE XYZ colour space

Figure 11: “Macbeth Color-Eye 7000” Spectrophotometry

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°869884