ZDMP: Zero Defects Manufacturing Platform

WP5: ZDMP Core Services and Middleware

EU ID: D068: Orchestration, Monitoring and Alerting (M18) - Vs: 1.0.0A

ZDMP ID: D5.4a

Deliverable Lead and Editor: Jose Luis Alfonso, ICE

Contributing Partners: ICE, ASC

Date: 2020-06

Dissemination: Public

Status: EU Approved

Abstract

The deliverables for this task, and all WP5-8 tasks, are software and are of EU type “OTHER”. The software and accompanying material (eg description, instructions) is available on the ZDMP software repository which is updated dynamically. However, for EU formal reporting purposes, this brief cover document provides a formalised pointer to the downloadable software and related content. This deliverable should read in conjunction with the D006-D020 deliverables which document the software process/status for each WP/Task. This deliverable represents the status as at M18 with further living editions at M18 and M48

Grant Agreement: 825631

Document Status

Deliverable Lead	Jose Luis Alfonso, ICE
Internal Reviewer 1	Sandra Vilaplana, CET
Internal Reviewer 2	Petru Demian, CONT
Internal Reviewer 3	Stuart Campbell, ICE
Type	Deliverable
Work Package	WP5: ZDMP Core Services and Middleware
ID	D068: Orchestration, Monitoring and Alerting (M18)
Due Date	2020-06
Delivery Date	2020-06
Status	EU Approved

Status

This deliverable is subject to final acceptance by the European Commission.

Further Information

www.zdmp.eu and mailto:info@zdmp.eu

Disclaimer

Project Partners:

UNINOVA Instituto de Desenvolvimento de Novas Tecnologias

ICE Information Catalyst Services Data Software

Ford Go Further

MONDRAGON ASSEMBLY

HSD MECHATRONICS

FIDIA

FORMPLAST SVOBODA

CONSULGAL CONSULTORES DE ENGENHARIA E GESTÃO, S.A.

PT MILLS

FLEXEFELINA STEEL TUBES

cei by ZIPOR

Continental

AlfaTest Your automation and test partner

DIN

software AG

SIMAVI Software Imagination & Vision

ceteck

VIDEOSYSTEMS Technologies for a new world

ASCORA Consulting - Development - Research

PROFACTOR

algowATT GREEN TECH SOLUTIONS

Etxetar

AL ANTÓNIO LONGARITO

rooter

ITI INVESTIGATE TO INNOVATE

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ikerlan MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE

iT innovation

MARTINREA

Tampere University

Executive Summary

The main objective of “WP5: Core Services and Middleware” is to deliver the core Industrial IoT/Network support of data acquisition, interoperability, and AI/analytics supported by orchestration, monitoring and autonomous computing.

The deliverables of this work package and the WP1 Management work package are divided into software packages and document/reports. In terms of reporting:

- **Process/Status:** This report corresponds to D009 Technical Management: WP5 Report of WP1 Management: Procedures, Metrics, Coordination, and Reporting and, as identified in the DOA, focuses on the process/status of the work accomplished in WP5
- **Software:** All WP5 software deliverables of T5.1-T5.5 (type “OTHER”) are available in the ZDMP public repository with access details and install instructions further described in this report which is a ‘current’ extract of the repository

“WP5: ZDMP Core Services and Middleware” consists of the following main parts: Data Acquisition, Network Support, Data Harmonisation, Orchestration and Monitoring, Distributed and Autonomous Computing, and AI and Analytics. The tasks of WP5 are the following:

- T5.1: Data Acquisition and IIoT
- T5.2: Robust Industrial Network Support
- T5.3: Data Harmonisation and Interoperability
- T5.4: Orchestration, Monitoring, and Alerting
- T5.5: Distributed and Autonomous Computing
- T5.6: AI and Analytics

This deliverable represents Task T5.4 Orchestration, Monitoring and Alerting which in turn is components of the following components:

- Orchestration Designer and Runtime
- Monitoring and Alerting

As reported in the architecture deliverable the purpose of these components is: “The Orchestration Designer is responsible for allowing users to model multiple manufacturing workflows so orchestrating the various assets available within a collaborative framework. The Orchestration Run-time interprets and uses process flows designed by Orchestration Designer component. The Monitoring and Alerting component is responsible for allowing users to collect data, eg KPIs and other data points from machines, infrastructure, and zApps.”.

Each of the components is structured into the following sections:

- General Description
- Architecture Diagram
- Features
- Requirements
- Installation
- How to Use
- Functional Requirements Implementation Status (M18)

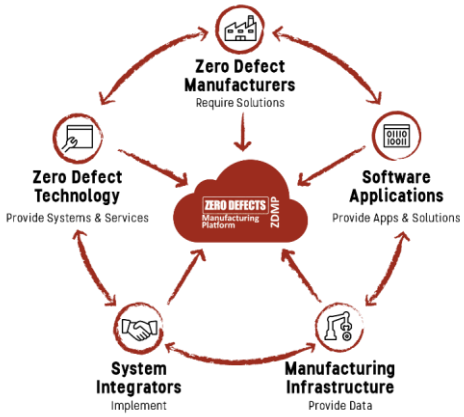
This report covers the period from the project start until M18 with most activity in the M13-M18 period. Further formal deliverables are due M30 and M48 as well as an informal iteration at 24.

Table of Contents

0 Introduction

Due to the cover nature of this deliverable; this introduction is presented in short-form only. For further information please consults D006 - Technical Management: Overview Report.

0.1 ZDMP Project Overview



ZDMP – Zero Defects Manufacturing Platform – is a project funded by the H2020 Framework Programme of the European Commission under Grant Agreement 825631 and conducted from January 2019 until December 2022. It engages 30 partners (Users, Technology Providers, Consultants and Research Institutes) from 11 countries with a total budget of circa 16.2M€. Further information can be found at www.zdmp.eu.

ZDMP aims at providing such an extendable platform for supporting factories with a high interoperability level, to cope with the concept of connected factories to reach the goal of zero-defect production. For this, the platform provides the tools to allow following each step of production, using data acquisition to automatically determine the functioning of each step regarding the quality of the process and product.

0.2 Deliverable Purpose and Scope

The deliverables for this task, and all WP5-8 tasks, are software and are of EU type “OTHER”. The software and accompanying material (eg description, instructions) is available on the ZDMP software repository which is updated dynamically. However, for EU formal reporting purposes, this brief cover document provides a formalised pointer to the downloadable software and related content. This deliverable should read in conjunction with the D006-D020 deliverables which document the software process/status for each WP/Task. This deliverable represents the status as at M18 with further living editions at M18 and M48. Specifically, the DOA states the following regarding this Deliverable:

T5.4	Orchestration, Monitoring, and Alerting			ICE	Six Monthly
D068 D069 D070	Orchestration, Monitoring, and Alerting	OTHER (Prototype)	PU	18, (24), 30, 48, Reporting via T1.4.x Series	RDI3-6, 8
Process orchestration will be responsible for executing manufacturing Applications and affect the processes which are currently running. The process execution engine will be based on existing commercial-grade but open source tools as used in the vf-OS project. Delivery will be the repackaging and production of the Processing Engine as well as an accompanying design studio. Coupled to this will be the monitoring and alerting system which will be triggered, typically, via process steps (themselves interacting with data providers) and then provided targeting delivery to those who may wish to monitor the stems – this alerting could be via email, text etc (as necessary).					

0.3 Target Audience

The primary target audience for this document are the partners and WPs of the project, as well as the EU and reviewers.

0.4 Deliverable Context

The deliverable context is as per Section 0.2:

Primary Preceding documents:

- **D006: Technical Management Overview Report:** Represents the general software status of the project including information on commits and WP5-8 Risks and mitigations
- **D009: Technical Management: WP5 Report:** Represents the process/status and future actions of this work package, including this task. It also includes related KPIs and their status
- **D055: Technical Specification and Update:** Describes the different APIs of the components

0.5 Document Structure

This deliverable is broken down into the following sections:

- **Section 1:** Component: Orchestration Designer and Runtime
- **Section 2:** Component: Monitoring and Alerting

0.6 Document Status

This document is listed in the Description of Action as “public” since it represents the open nature of the project’s software deliverables.

0.7 Document Dependencies

- None

0.8 Glossary and Abbreviations

A definition of common terms related to ZDMP, as well as a list of abbreviations, is available at http://www.zdmp.eu/glossary.

0.9 External Annexes and Supporting Documents

- See the ‘Resources’ grid within the General Description Section of each component

0.10 Reading Notes

- None

0.11 Document Updates

- This is the first version of this document

1 Component: Orchestration Designer and Runtime

1.1 General Description

The Orchestration Designer and Runtime component is responsible for allowing users to model multiple manufacturing workflows to orchestrate the various assets available within a collaborative framework.

To support the objectives of ZDMP a tool needs to be created and then utilised that allows the orchestration of various ZDMP assets (components, sub-components or zApps) and other services.

The Orchestration Designer is a visual online reactive canvas allowing a business process designer to pull in existing models from a library representing the virtualised manufacturing assets. Each asset may support additional properties that can be defined. The defined workflow can consist of sub workflows and be saved and versioned within the storage as (eg BPMN 2.0) model definitions.

The Orchestration Runtime is based on open source BPMN engine Camunda, and provides a “super layer” on top of it to manage process instances, user tasks, service to service communication, and the NodeJS Code Generator upgraded from the vf-OS project.

Resource	Location
Source Code	Link
Latest Release (v1.0.0)	N/A Download source and use docker compose
X Open API Spec	Link
Video	Coming soon

The date of generation of this component content is: 2020-06-25

1.2 Architecture Diagram

The following diagram shows the position of this component in the ZDMP architecture

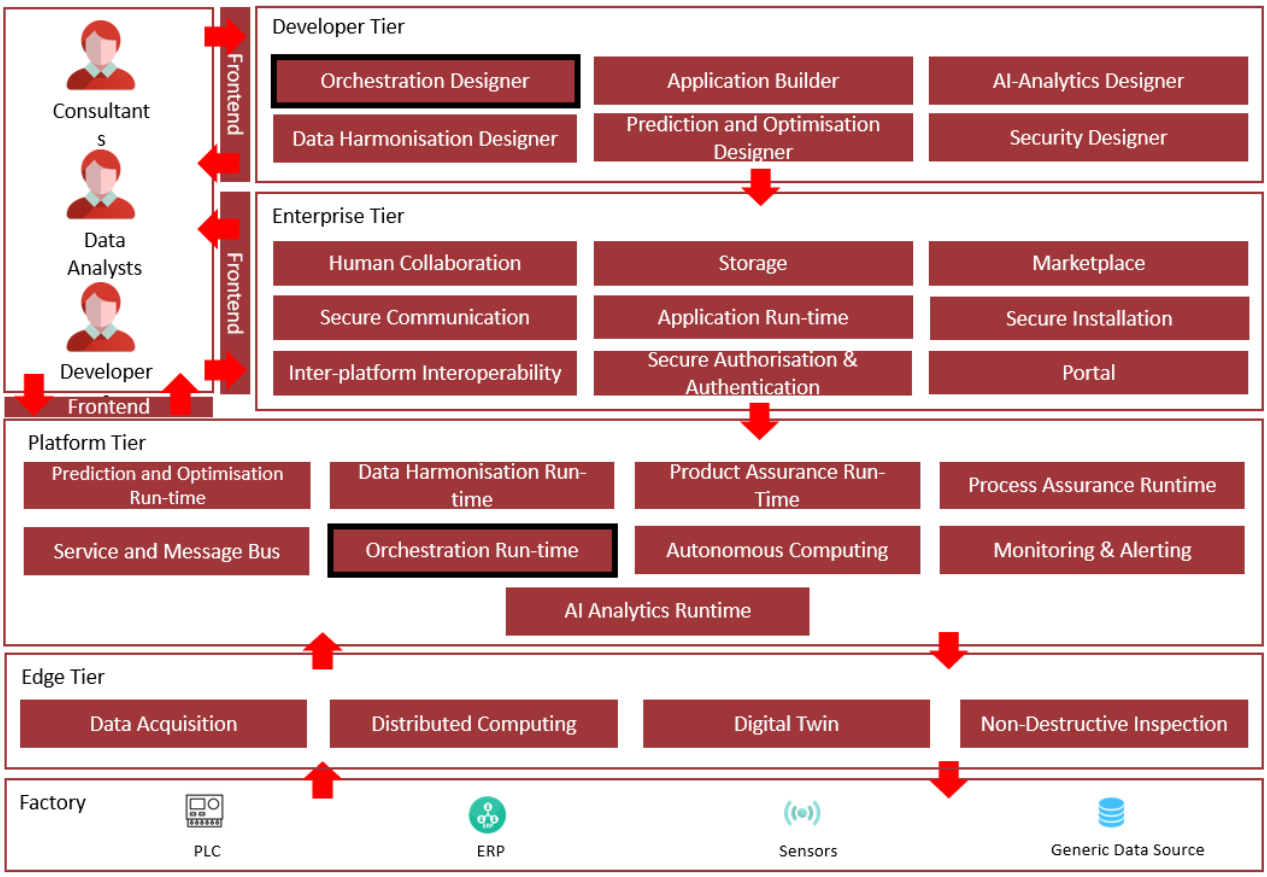


Figure 1: Position of Component in ZDMP Architecture

1.3 Features

The Orchestration Designer and Runtime offers the following features:

- **BPMN browser-based designer**: Full BPMN designer that provides the user all tools needed to design a business process:
  - User tasks
  - Exclusive, inclusive, parallel gateways
  - Service tasks
  - Script tasks using JavaScript
- **NodeJS Code Generator**: Used to convert the BPMN diagrams to fully performant NodeJS code. This gives superior performance for service to service communication processes
- **Integrated with Camunda Engine**: Powered by Camunda, the open source BPMN Engine, tweaked to get the most out of business process execution:
  - Start/Resume/Stop processes API
  - Process instance dashboard
- **Integrated with ICE Service Repository**: User can drag and drop a service from the repository, and include its functionality inside the business diagram

- **User tasks management**: User task management to support human input in business processes. Users can complete, assign. and/or claim tasks. The process engine automatically resumes its operation with the new user inputs, once the task has been completed
- **User tasks form editor**: This is a web-based user task forms designer, so users can customize the appearance of forms at runtime, adding/removing fields and choosing validations

1.4 System requirements

The Orchestration Designer and Runtime has the following system requirements:

- Docker

1.5 Installation

The ZDMP Orchestration Designer and Runtime can be installed using the following steps:

1. Download the latest source code from ZDMP repository
2. Under command line, while being in the same folder than the docker-compose.yml file, run: docker-compose up
3. Open browser and point to: http://localhost:8080

1.6 How to use

- **Manage processes**
  - The first-time the user opens the application, the welcome screen lets the user choose whether to create a new process or open an existing one. It is also possible to import a .BPMN file

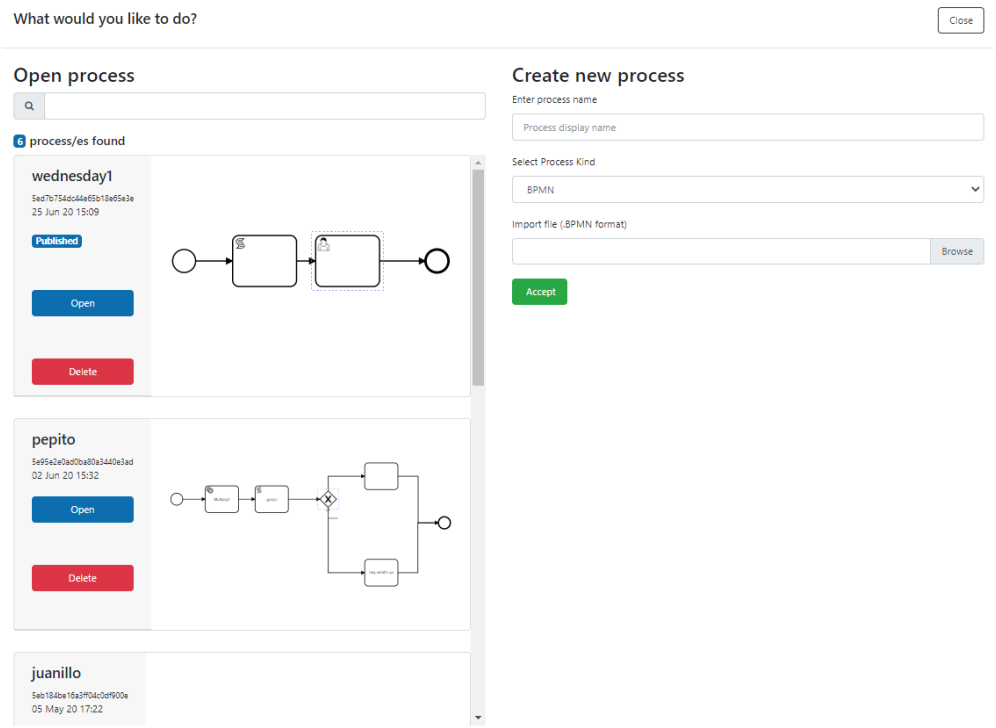


Figure 2: Welcome screen

- The process is created/opened and the main window opens, with the palette at the left, the canvas at the centre, and the properties panel at the right

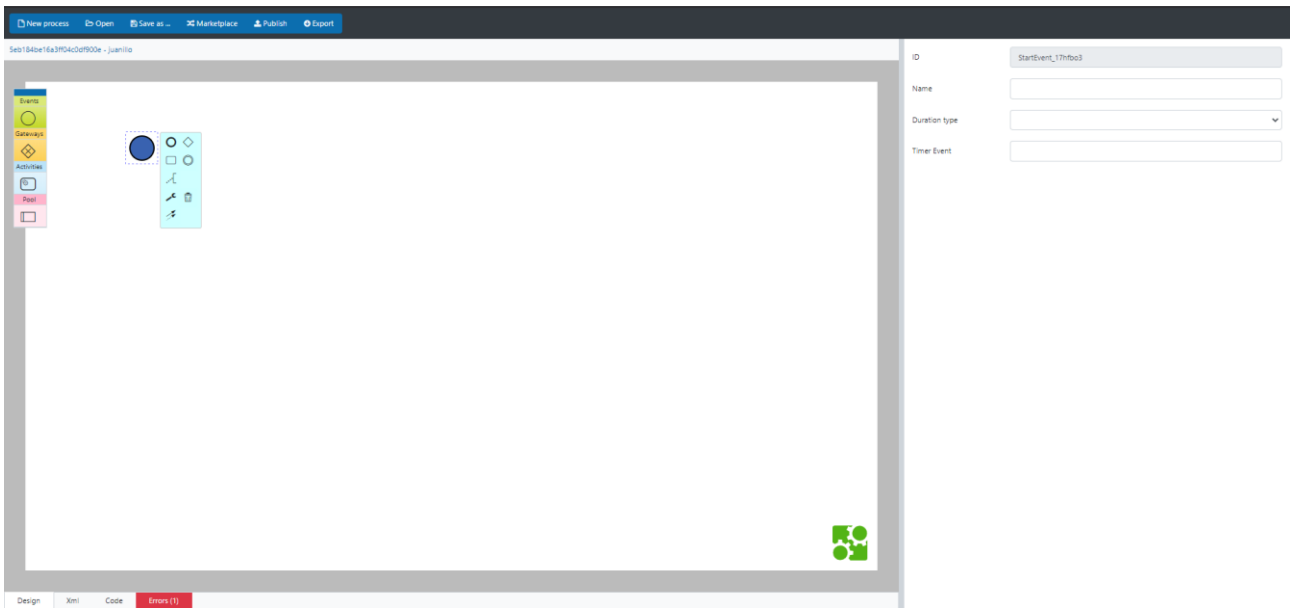


Figure 3: Main screen

- **Design a process**
  - When selecting the start element (the empty circle), the user can then select from the popup window the type of element to attach, whether it is a task (square), an event (circle) or a gateway (diamond)
  - It is also possible to attach elements by dragging and dropping from the palette
  - After selecting an element, it is possible to move it to another place by simply dragging it
  - The zoom can be increased/decreased by pressing Ctrl while moving the mouse wheel
  - The auto save feature takes care of persisting the changes while the user creates the diagram

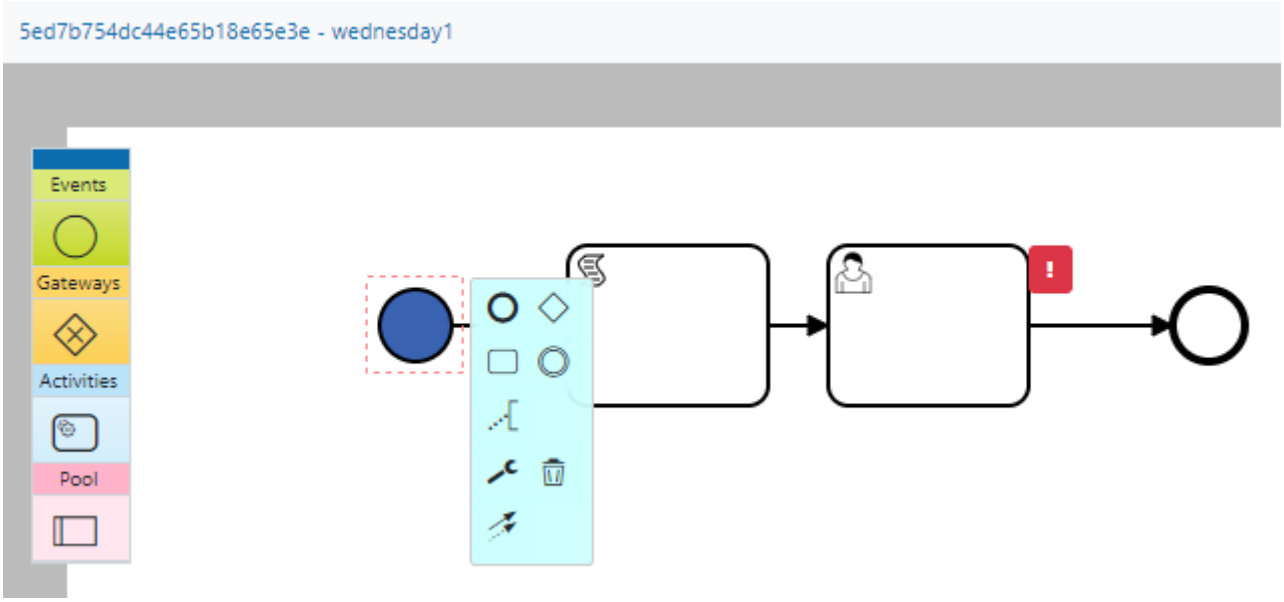


Figure 4: Designing a process

- **Automatic validation**
  - Whilst the user designs a process, there will appear red markers around the elements, indicating that there are missing required parameters
  - The elements must be properly configured, otherwise the list of errors (at the bottom) will appear in red, showing the number of errors
  - It is also possible to open the error list, by clicking on it, so it provides a better description of each error
  - Once the problem is solved, the error automatically disappears

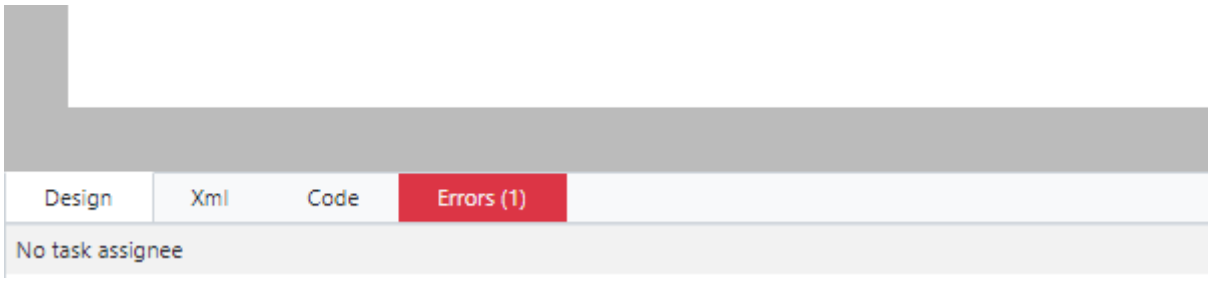


Figure 5: List of errors at the bottom

- **Connecting services from the marketplace**
  - When adding a service task, the Marketplace Browser automatically opens, letting the user to choose between any available service
  - In the Marketplace Browser, each service describes its behaviour, so the user can choose the one that adjust to the user's needs
  - After selecting the service, it is added to the diagram, and validation errors will appear indicating any required input or parameter to configure the service

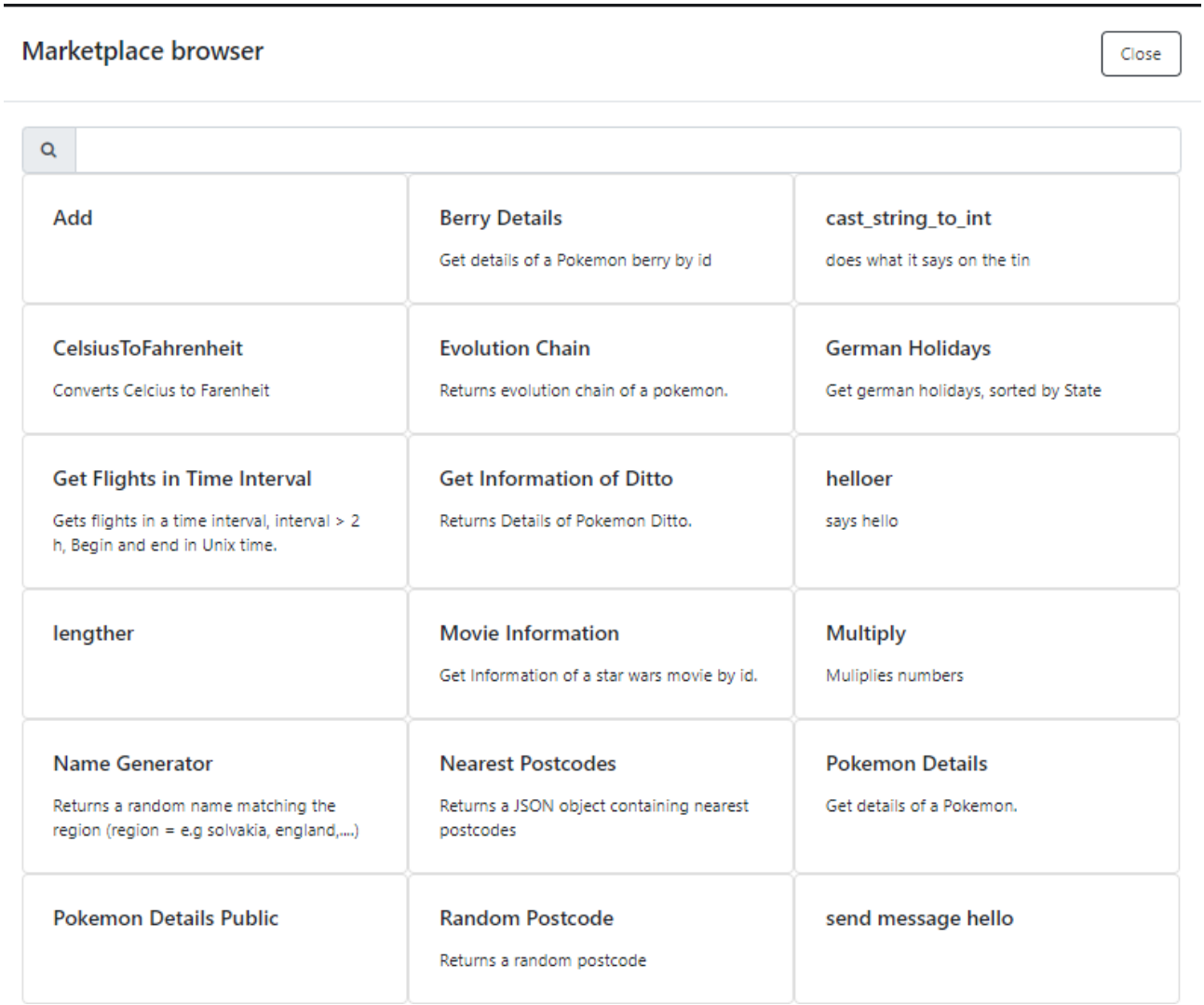


Figure 6: Marketplace browser

- **Code Generator / XML View**
  - At any time, pressing the Code tab, the user can see the auto generated NodeJS code that represents the process flow
  - User tasks and long timers are skipped, since these elements would stop the calling process when running the generated code
  - Typescript blocks written in the Script Editor, are automatically converted to JavaScript
  - The XML view shows the full XML structure of the process

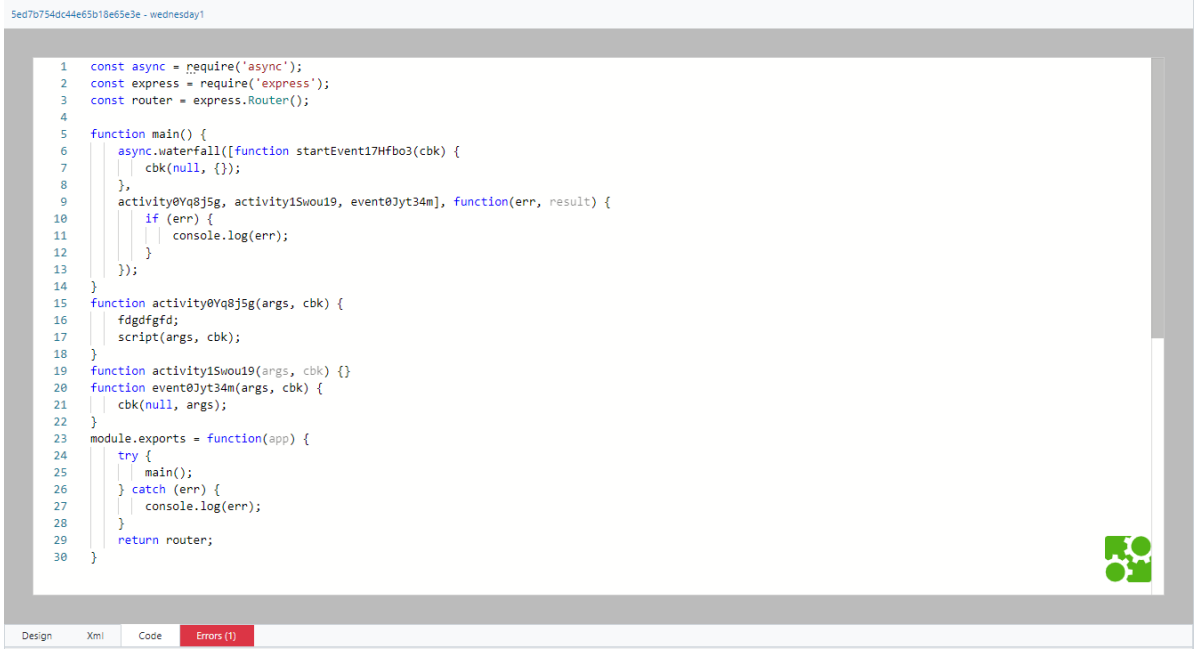


Figure 7: Code generated

- **User tasks management**
  - When adding a service task, the properties panel shows the additional properties supported by this type of element
  - Among others, the user can select the assignee of the task, so when in runtime, the task will be automatically assigned to that user, when the process instance reaches that element
  - The user can also select an existing form, previously created using the Forms Editor
  - The User Tasks main panel can be used in runtime to claim and complete tasks

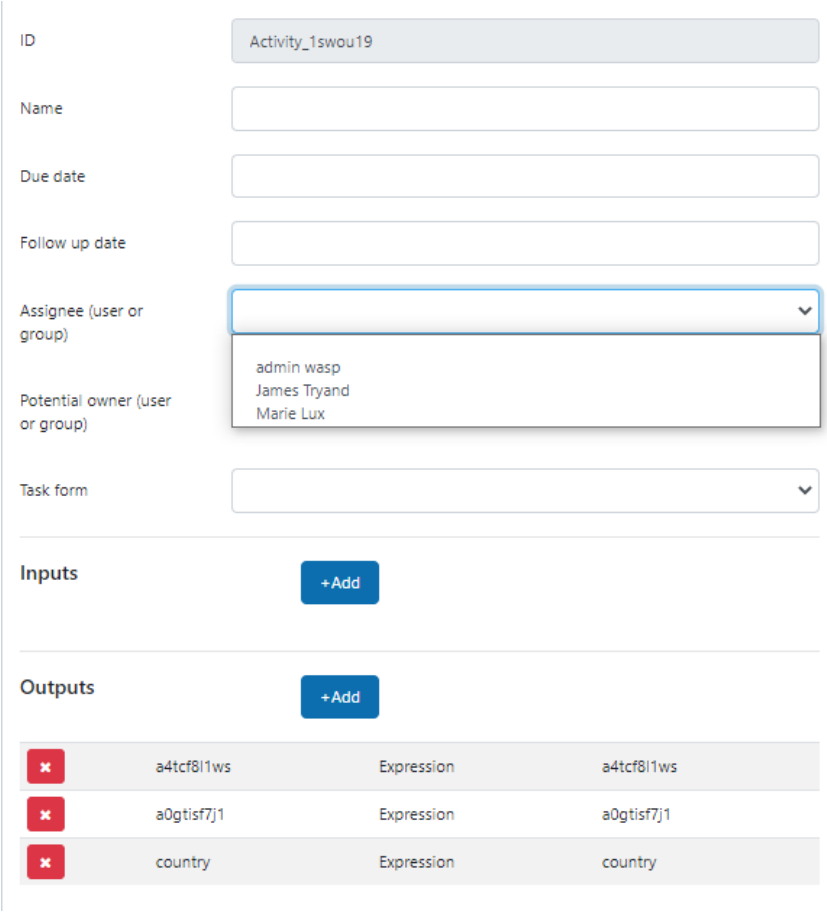


Figure 8: Properties panel for user task

- **Publishing a process for Runtime**
  - Once the process has been designed, the publish action makes it available for the process engine
  - Only processes without errors (totally validated) can be published
  - After they have been published, the Process Control Panel can be used to manage process instances

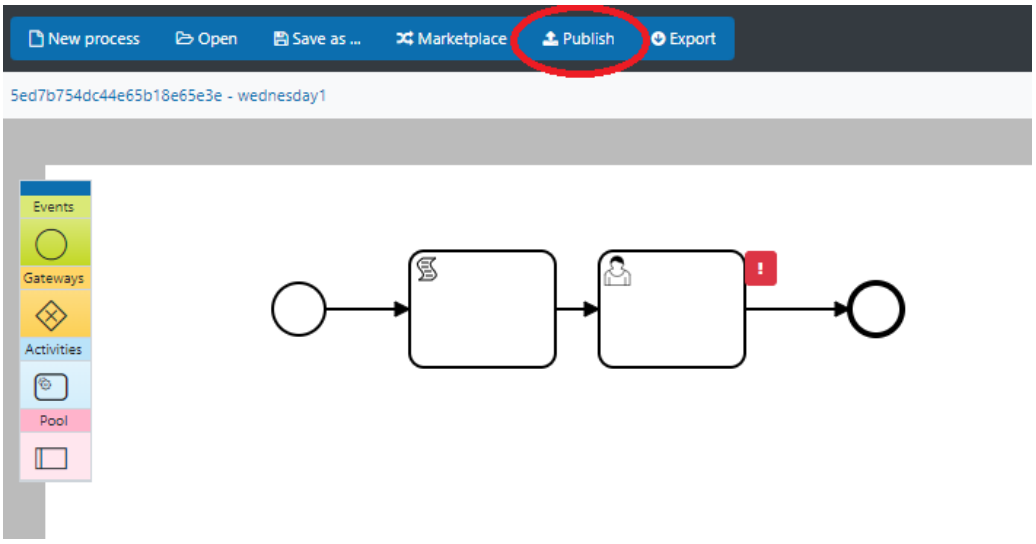


Figure 9: Publish action

- **Process Control Panel**
  - Used for managing the start, resume, and stop process instances
  - All processes published from the Process Designer will appear here
  - The user can get metrics about each process, number of instances and possible problem

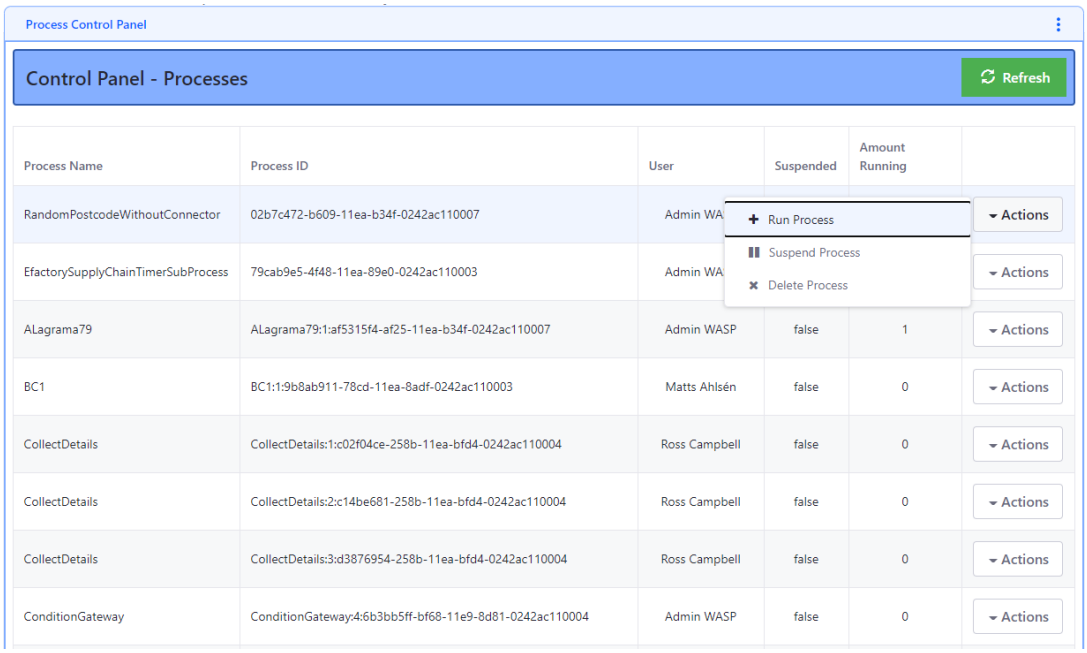


Figure 10: Process Control Panel

1.7 Functional Requirements Implementation Status (M18)

The actual implementation status vis-à-vis the functional requirements implementation at M18 is provided in the annex of the D006 Technical Management Overview Report. This represents the general software status of the project and this WP/Task including information on commits and WP5-8 Risks and mitigations. Below is shown a dummy example for a security component.

Functional requirement	Description	Status	Progress	Comments
T52A013 - Issue New certificates	New client certificates are created. These certificates include the details that permit the identification of the subject (physical device, gateway or server).	Working	90%	Beta version, requires integration API with security command centre for credentials tokenization

2 Component: Monitoring and Alerting

2.1 General Description

The Monitoring and Alerting component is responsible for allowing users to collect data, eg KPIs and other data points from machines, infrastructure, and zApps. The component is also responsible for alerting users and other ZDMP components in case a KPI get out of defined limits, reducing the impact of crises and losses to smart factories.

The different KPIs delivered via the platforms message bus can be configured to be stored in the Storage Component to collect historic data. If historic data is collected, different choices of histograms are presented for the user to choose how the data should be presented.

In order to be able to notify about possible problems, users can define limits for data points, as well as qualifiers (eg energy consumption is ‘larger than’ & ‘100 kWh’) to trigger alerts, ie SMS, emails, push notifications as well as calls to HTTP endpoints when these limits have been crossed for the first time. These limits are used as goals (for example by Autonomous Computing) where a process can be started if this limit is not reached.

Additional alerts can be sent if the component has not sent a response after a defined time has passed. The receivers should be able to check that the problem has been recognized so the system knows it is already being acted on. If this has not happened after passing a critical value, the system notifies other receivers. A reset timeframe can be created, to indicate a duration in which the data point must be back within the regular defined value to be able to trigger the alert again.

Resource	Location
Source Code	Link
Latest Release (v0.3.0)	Download
X Open API Spec	Link
Video	Coming soon

The date of generation of this component content is: 2020-06-25

2.2 Architecture Diagram

The following diagram shows the position of this component in the ZDMP architecture

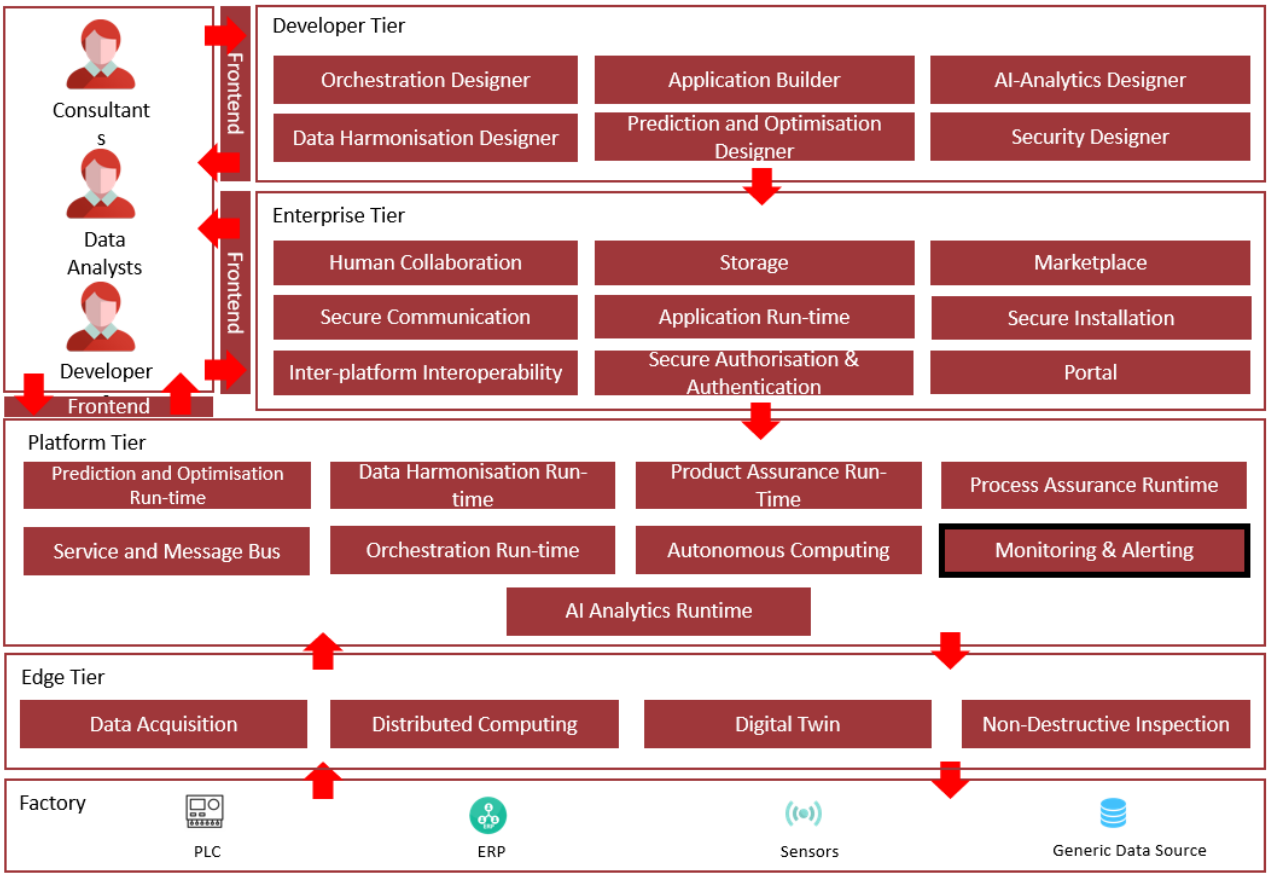


Figure 11: Position of Component in ZDMP Architecture

2.3 Features

This component offers the following features:

- **Create KPIs**: Allow the user to create KPIs to extract important values from the data transmitted through the Message Bus. The user needs to specify through which topics the data will be extracted from
- **View KPI's Historic Data**: Allow the user to see the KPI data and the changes that occurred through time
- **Create Alert**: Allow the user to create Alerts to notify users when the KPI's values are not within the expected by quality standards
- **Create Message Template**: Allow the user to create message templates to be re-used in the Alerts, providing a better standard for the notifications
- **Send Notification / Send Alert**: Allow the user to send an email directly to a user, without the need of an alert to do so. This feature is only available in the API and not in the UI

2.4 System requirements

Hardware Requirements:

- CPUs
- 8GB RAM
- 64GB disk space

Software Requirements:

- Docker

2.5 Installation

The Monitoring and Alerting component can be installed via docker-compose, for that is also needed to have a server for the email credentials:

1. Download the latest docker-compose file from ZDMP's GitLab
   $ wget https://zdmp-gitlab.ascora.eu/zdmp_code/platform-tier/t5.4-monitoring-and-alerting/-/blob/master/orchestration/docker-compose.yml
2. Add the environment variable values. Choose the way to do it following the instructions from docker: https://docs.docker.com/compose/environment-variables/. As an example, create a file named '.env' in the same folder of the docker-compose file, with the following information:
3. EMAIL_SERVER_USER=emailServer@provider.com
4. EMAIL_SERVER_PASSWORD=password
5. Install and start the component by executing the following command:
   $ docker-compose up -d

2.1 How to use

- API
  Please refer to http://localhost:3000/api for the Swagger instructions on how to use the API. There are all the possible requests the component accepts, and its expected parameters or body content. The API can be accessed in http://localhost:3000/
- User Interface (UI)
  Access http://localhost:80 to access the user interface.

2.1.1 KPI

A KPI references a data value that holds a significant meaning for the user, as an example, the length of pencil produced by an automatic machine. As the length of the pencil is one of the keys to measure the quality of the production, we can create an KPI of the length of the Pencil.

To create a KPI, the following is necessary:

- Description to identify the KPI
- Message Bus topics that should be used to extract the KPI value
- Data format expected and the query used to extract the data. The possible data types are JSON and xml.

Follow an example, that extracts the length of the pencils produced:

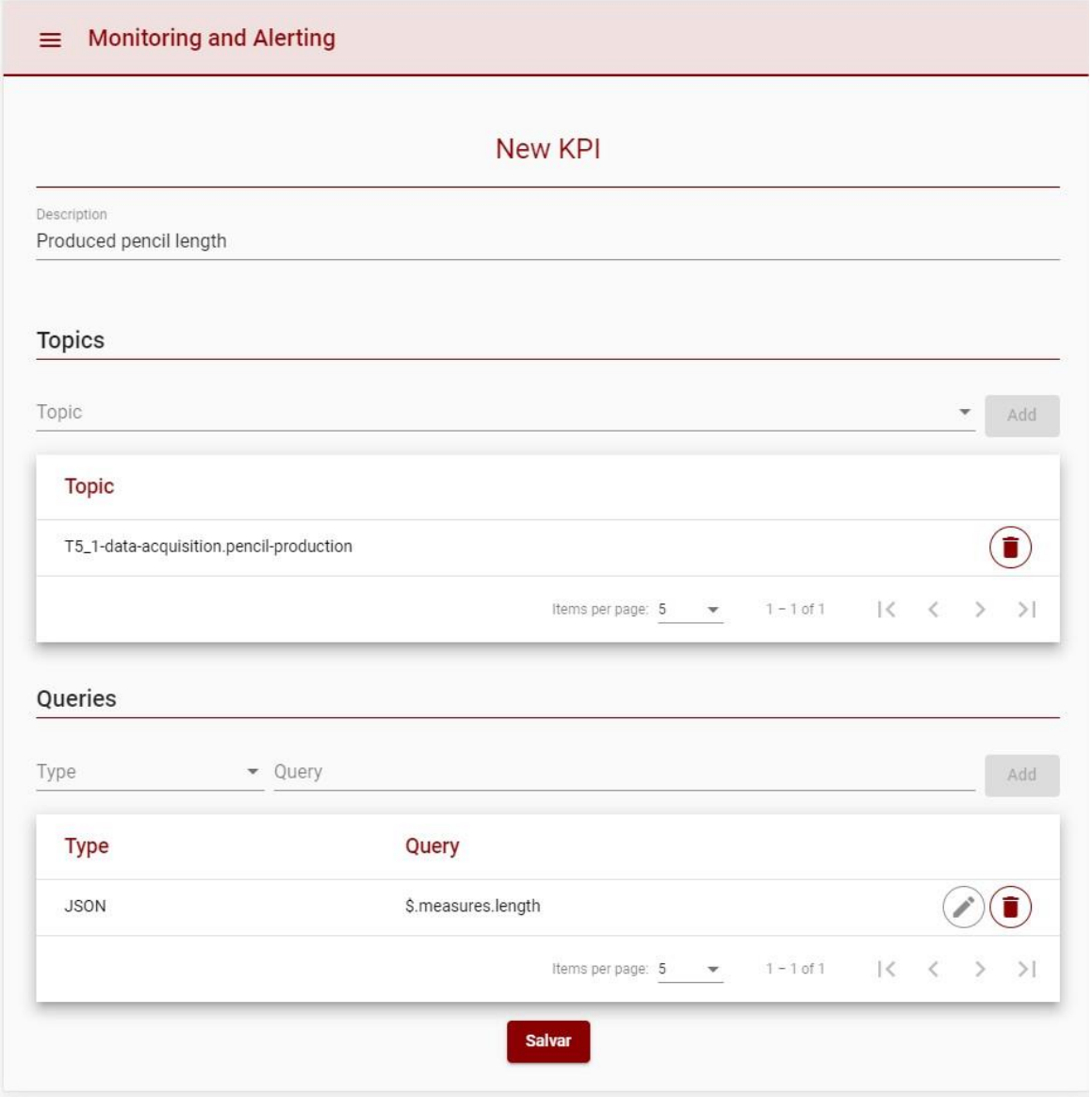


Figure 12 – Create new KPI

After the KPI is created, the item will appear in the KPI List View, and will be available to be used to create Alerts, or in other components that uses the list of KPI's saved:

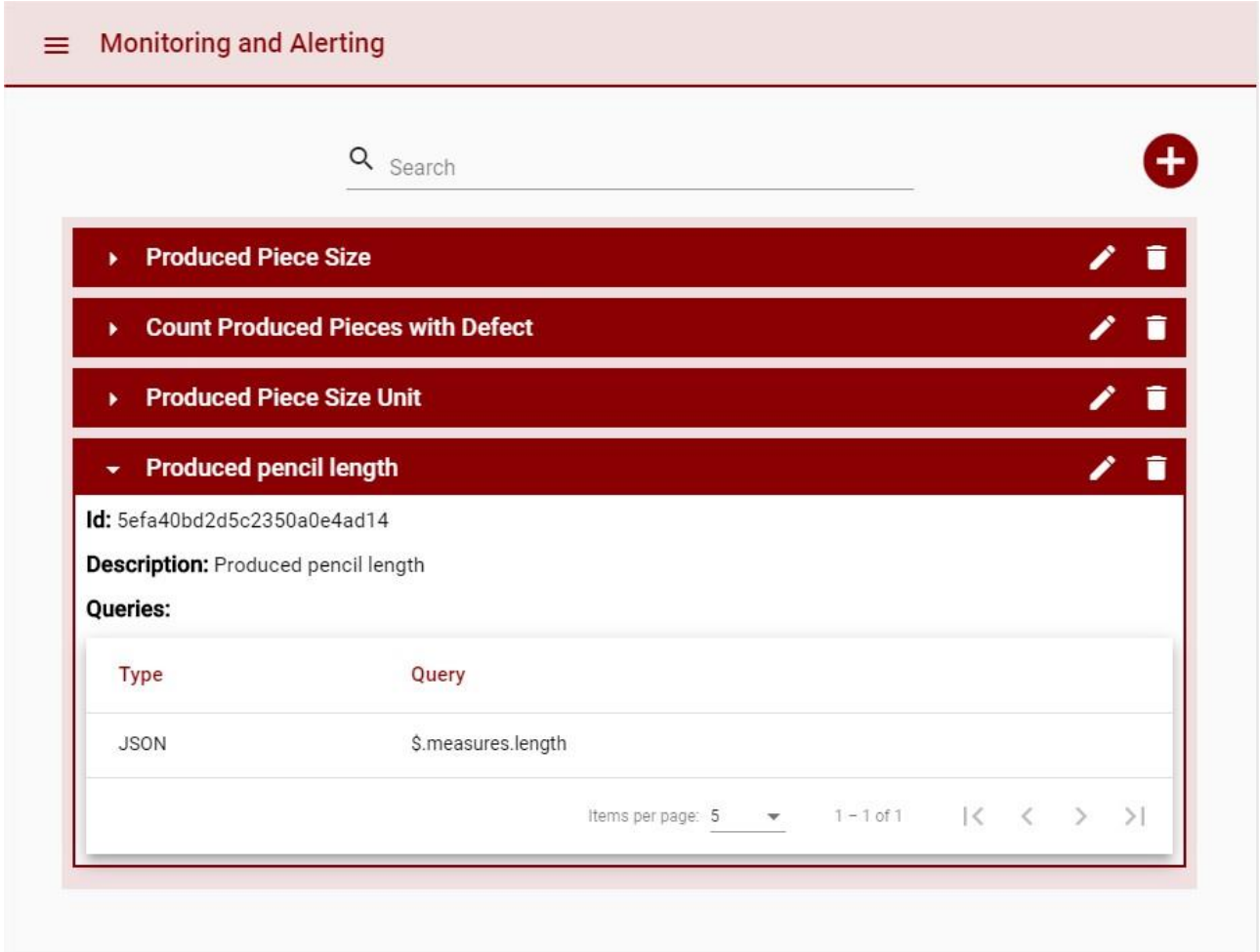


Figure 13 – KPI List View

2.1.2 Alerts

One or more conditions can be applied to KPI's to ensure the quality of the products, and in case a KPI value indicates a quality failure an Alert can be sent to one or more users. Following the pencil example, an alert can be created when the length of the pencil is outside the range delimited by quality standards.

To create an Alert, the following is necessary:

- Description to identify the Alert
- Condition that compares the values of one KPI
- When more than one condition is provided, a logic query identifying the relation between the conditions needs to be provided. (See example bellow)
- Message to be sent when the conditions are matched
- One or more users to receive the message when the conditions are matched

The following example alerts a user when the length of the produced pencil is out of the range delimited by quality standards:

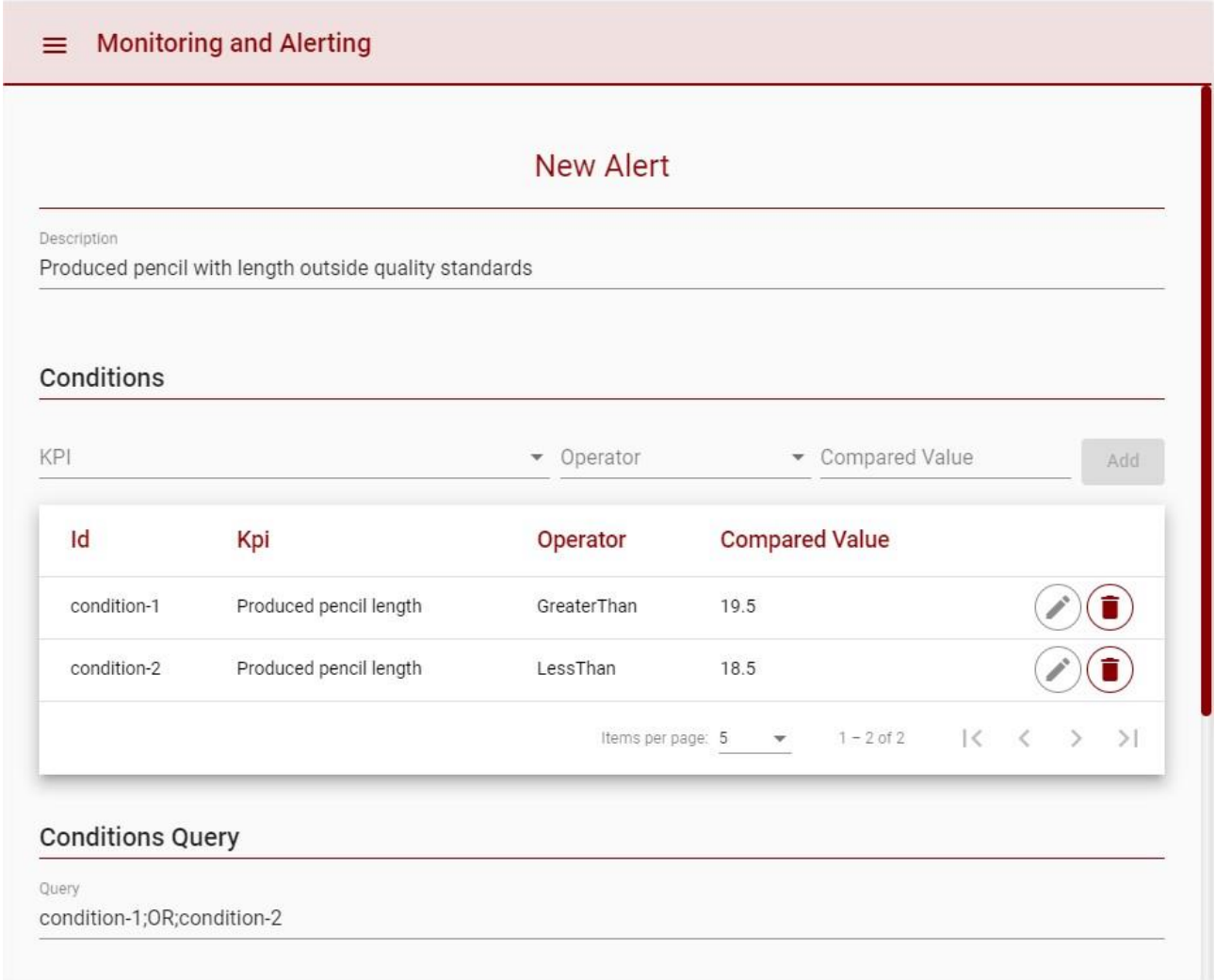


Figure 14 – Create new Alert – Part 1

The alert message is:

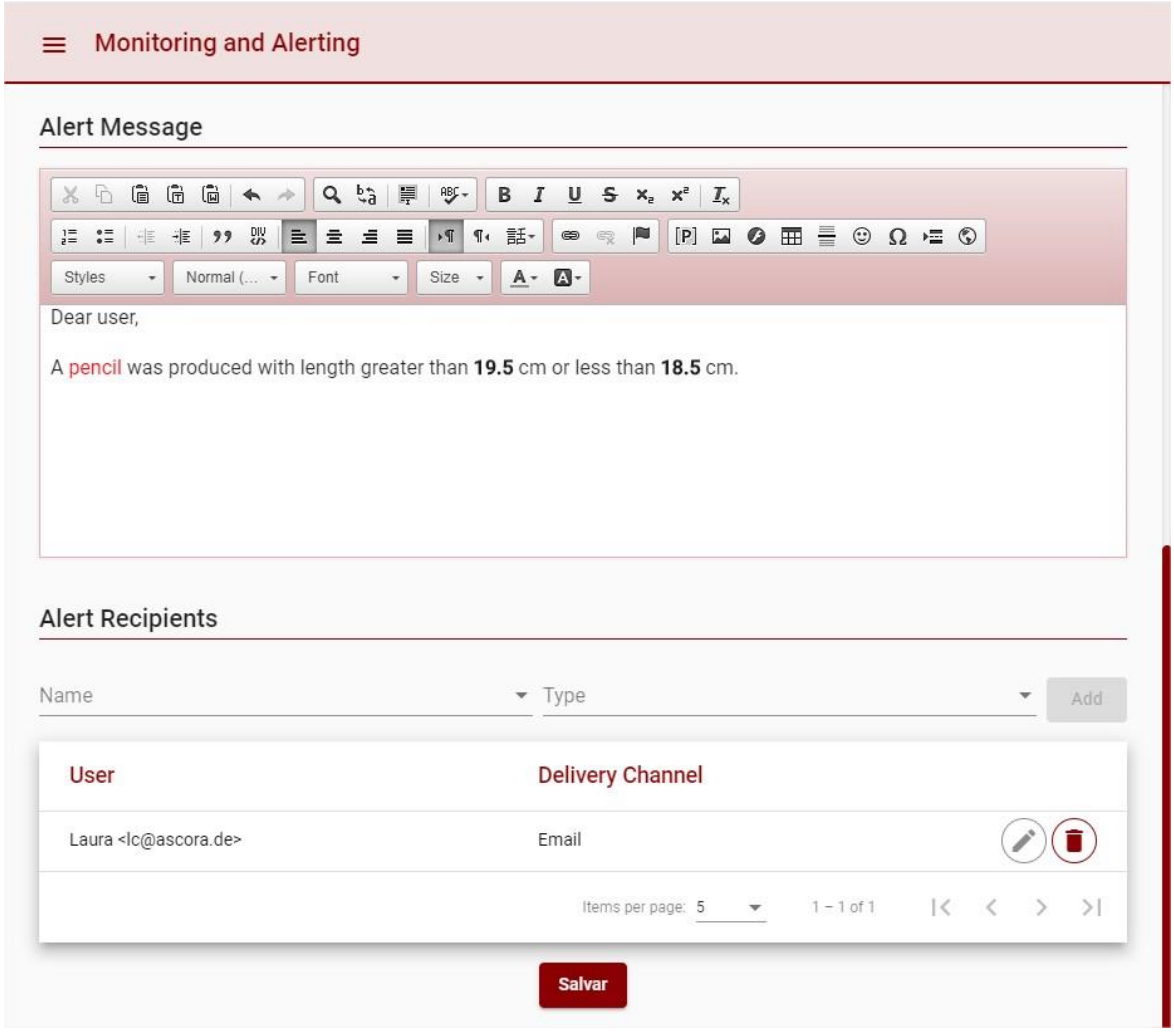


Figure 15 – Create new Alert – Part 2

The Conditions query must be formulated by using the conditions identifiers, a semicolon (;) between items, and logical operators (AND, OR, NOT), to create an logical expression. As example:

Considering the following conditions:

- condition-1: produced pencil length > 18
- condition-2: produced pencil type = ‘long-edition’

To send an alert when the produced pencil length is greater than 18 but its not a “long-edition pencil”, the following condition query must me defined:

- condition1;AND;NOT;condition-2

After the Alert is created, the item will appear in the Alert List View, and the conditions will start to be monitored by the component:

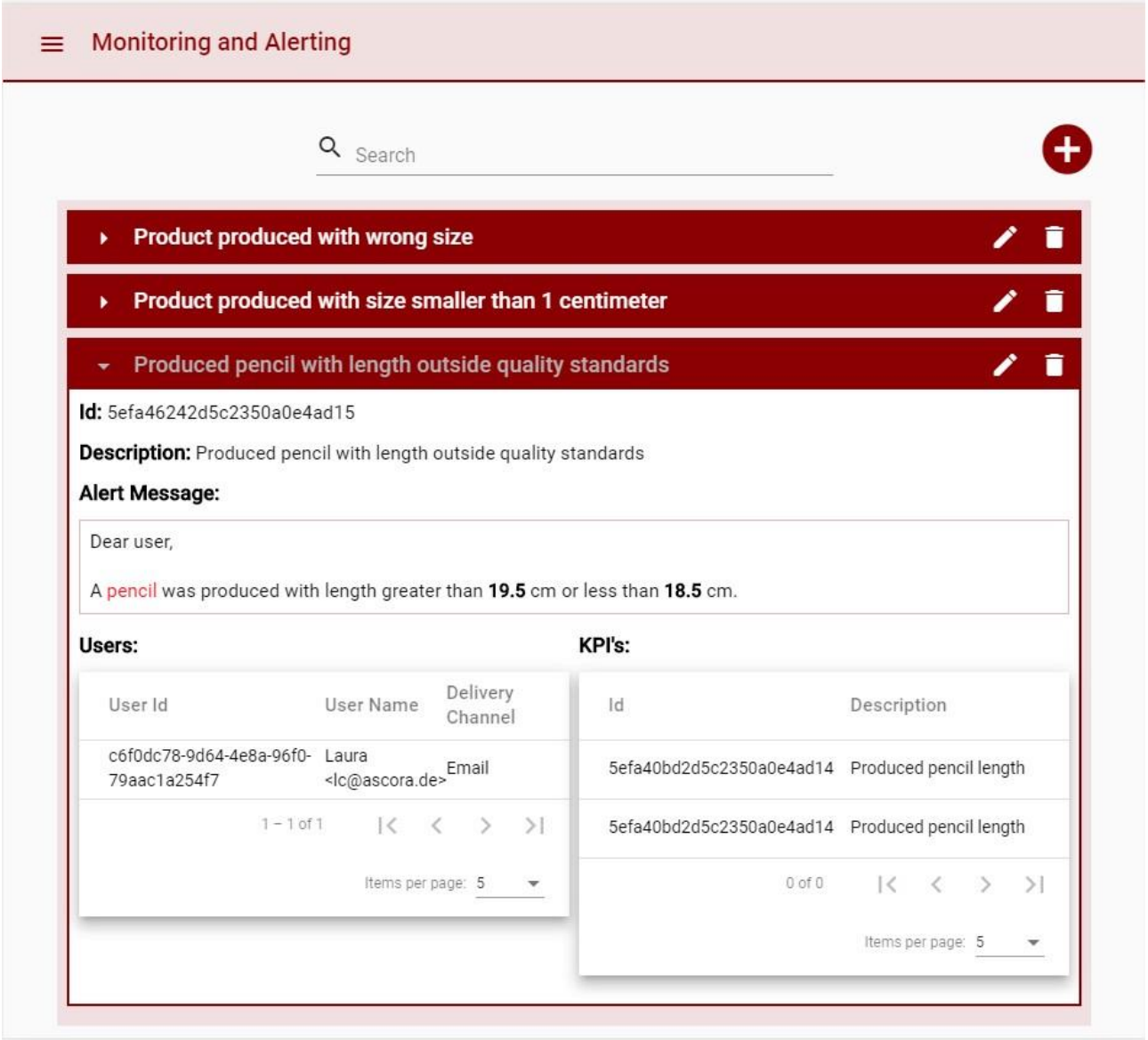


Figure 16 – Alert List View

When the conditions are matched, an e-mail will be sent to the users:

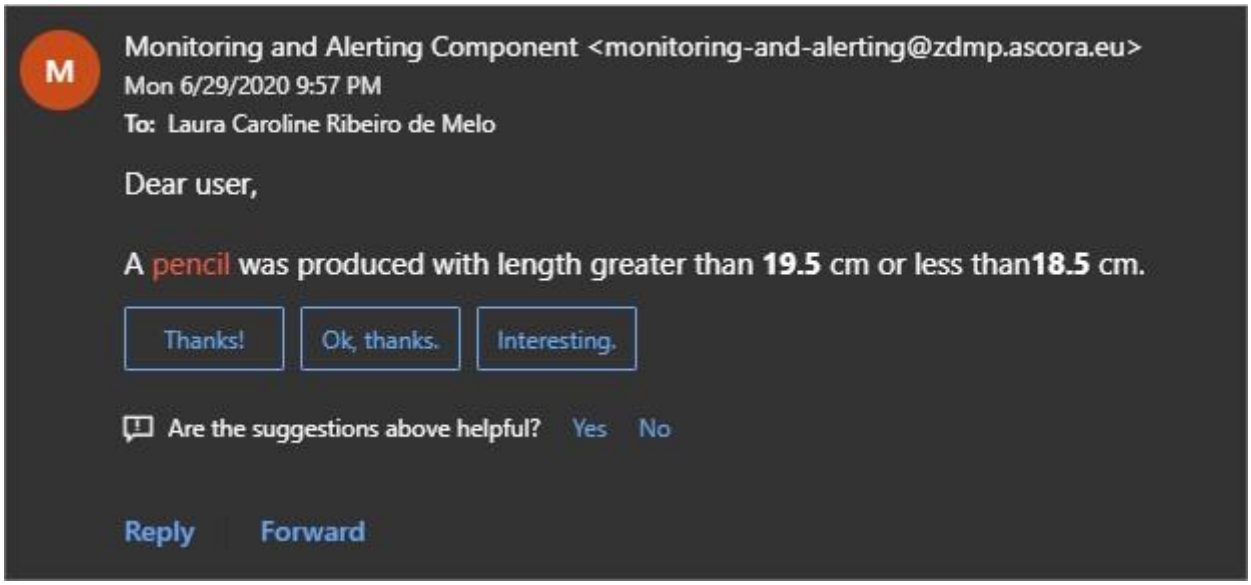


Figure 17 – E-mail sent by Alert

2.2 Functional Requirements Implementation Status (M18)

The actual implementation status vis-à-vis the functional requirements implementation at M18 is provided in the annex of the D006 Technical Management Overview Report. This represents the general software status of the project and this WP/Task including information on commits and WP5-8 Risks and mitigations. Below is shown a dummy example for a security component.

Functional requirement	Description	Status	Progress	Comments
T52A013 - Issue New certificates	New client certificates are created. These certificates include the details that permit the identification of the subject (physical device, gateway or server).	Working	90%	Beta version, requires integration API with security command centre for credentials tokenization

3 Conclusions

This deliverable is the first deliverable in the reporting series for T5.4 Orchestration, Monitoring and Alerting. The deliverables for this task, and all WP5-8 tasks, are software and are of EU type “OTHER”. The software and accompanying material (eg description, instructions) is available on the ZDMP software repository which is updated dynamically. However, for EU formal reporting purposes, this brief cover document provides a formalised pointer to the downloadable software and related content.

This deliverable should read in conjunction with the D006-D020 deliverables which document the software process/status for each WP/Task vs its content. This deliverable represents the status as at M18 with further living editions at M18 and M48 and an informal iteration at M24.

ZERO DEFECTS
Manufacturing
Platform

ZDMP

www.zdmp.eu