

# VerSatile plug-and-play platform enabling remote pREdictive mainteNance

**Grant Agreement No** : 767561  
**Project Acronym** : SERENA  
**Project Start Date** : 1<sup>st</sup> October 2017

**Consortium** : COMAU S.p.A.  
Finn-Power Oyj  
VDL Weweler BV  
WHIRLPOOL EMEA SpA  
Kone Industrial Ltd  
Engineering Ingegneria Informatica S.p.A.  
OCULAVIS GmbH  
SynArea Consultants S.r.l.  
DELL EMC  
Laboratory for Manufacturing Systems & Automation  
Fraunhofer Gesellschaft zur Förderung der angewandten Forschung  
VTT Technical Research Centre of Finland Ltd  
TRIMEK S.A.  
Politecnico Di Torino



**Title** : Deliverable Name  
**Reference** : D2.1  
**Dissemination Level** : PU (public)  
**Date** : 2018-09-30  
**Author/s** : IPT  
**Circulation** : EU/Consortium

## **Summary:**

*Purpose of this document is to summarize activities on the design and development of a framework for condition monitoring of factory equipment, as a result of the SERENA project activities in the first year.*



**Contents**

**List of Abbreviations .....3**

**List of Figures.....3**

**Executive Summary .....4**

**1 Introduction.....5**

**1.1 Motivation.....5**

**1.2 Work package objectives .....5**

**1.3 Requirements.....5**

**2 DataBox 4.0 design .....6**

**2.1 Concept .....6**

**2.2 DataBox 4.0.....6**

        2.2.1 Description.....6

        2.2.2 Design Constraints .....7

        2.2.3 Hardware architecture .....7

        2.2.4 Software architecture.....8

**3 System interfaces .....11**

**3.1 Human – machine interfaces .....11**

        3.1.1 Edge Gateway GUI / Operating system..... 11

**3.2 External interfaces.....11**

        3.2.1 Network interface..... 11

        3.2.2 Additional Interfaces via WAGO interface controller..... 11

**4 Integrity controls .....12**

**5 Operational scenario .....13**

**6 Conclusion .....14**

**References .....15**



**List of Abbreviations**

CEN	European Committee for Standardisation
CWA	CEN Workshop Agreement
ICT	Information and Communications Technology
KPI	Key Performance Indicator
OEM	Original Equipment Manufacturer
PLC	Programmable Logic Controller
SME	Small and Medium Sized Enterprise
TRL	Technology Readiness Level

**List of Figures**

Figure 1: System structure overview .....6

Figure 2: DataBox 4.0 Hardware Architecture and System integration.....8

Figure 3: Docker environment [1] .....9

Figure 4: Basic aspects of software infrastructure design .....9



## Executive Summary

The purpose of this document is to report the results of WP2 in project month 12 and contains the following tasks:

- Task 2.1: Design of versatile framework for factory condition monitoring
- Task 2.2: Machine's 'smart data' collection
- Task 2.3: Multi-level factory sensing and machine data matching

This deliverable provides an overview of the technical structure, software and implementation of the Serena DataBox developed in WP2. This includes the hardware composition of the DataBox, the connection to an existing control device (e.g. COMAU robot controller) and the integration into the existing company network. In terms of Software, the basic principles and components of device/software management, data acquisition, data pre-processing and communication infrastructure are explained.



## 1 Introduction

### 1.1 Motivation

The outcome of this WP deliverable is a basic design of the DataBox prototype. This will be the foundation for the following use case SERENA test systems and implementations that will be realized together with the use case partners.

### 1.2 Work package objectives

Main objective in M12 of WP2 are the definition of a basic system design including its internal structure in terms of hardware and software. This includes the concept of modularity, flexibility and plug and play capability. The versatile data collection framework and the connection to the embedded smart data extraction module of the system enable these main requirements.

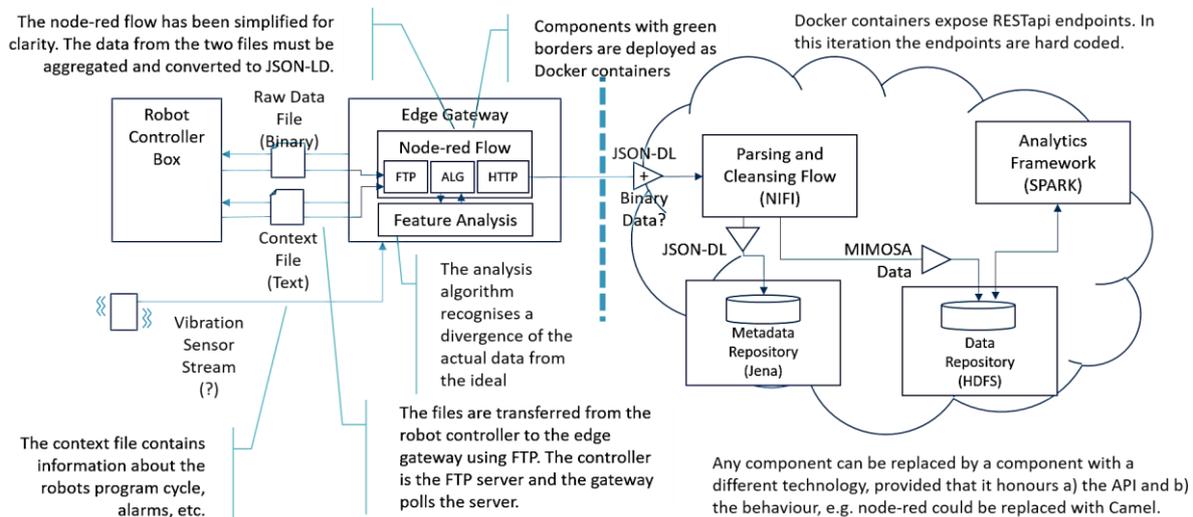
### 1.3 Requirements

The requirements for the DataBox development have been stated in D1.1. They are the main input for all further development including this basic design and the following prototype implementations.

## 2 DataBox 4.0 design

### 2.1 Concept

Figure 1: System structure overview Figure 1 gives an overview of the overall prototype implementations, based on the COMAU use case. This use case is the first test-implementation and basis for the development of the general system infrastructure.



**Figure 1: System structure overview**

In this use case, data is acquired from the robot controller developed by COMAU. The controller creates one binary file for each cycle and provides these files via HTTP REST to external systems. This first system design was developed along the COMAU use case, but is still designed in a general and not specific way to ensure the easy transferability to other use cases using the same design as basis.

### 2.2 DataBox 4.0

The DataBox itself can be any kind of field computing device that is able to run the software and fulfils the hardware requirements. Recommended is a small, ruggedized industrial field PC with a Windows or Linux distribution. The device also needs network connectivity in order to connect to the cloud system via company network and the sensor interface modules by WAGO. For the prototyping within SERENA, DELL Edge gateways of the 3000 series are used.

The functionalities of the DataBox are implemented using the NodeRed framework. Based on a graphical block design, new dataflows and functionalities can easily be implemented by using a large library of predefined functions. The NodeRed standard library includes many kinds of standard interface drivers and data manipulation algorithms. If necessary, self-written code can also be added to custom designed blocks.

#### 2.2.1 Description

The designed NodeRed flow contains all functionalities needed to read and handle the incoming and outgoing data, while feature analysis is done in a separate module. Separating these two main tasks makes it possible to exchange and maintain both separately and therefore makes it easier to combine different flow and processing modules according to the requirements of the individual DataBox. For the COMAU use case, the input comes from the COMAU robot controller which provides new data as binary file via HTTP REST for each cycle (one file per completed rerun of the robot program). The files contain the motor current and the angular position of each robot axis. The ALG-Module extracts the raw data from the received files, transfers it to the analysis module and receives the calculation/analysis results. The analysis module can contain any kind of data analysis algorithm and therefore is the



foundation for Task 2.2 and parts of Task 2.3 such as pre-processing and filtering. Relevant Smart Data for this use case are mostly statistical values of the motor current graph, for example mean, min, max, peak to peak or RMS. Long term analysis of these values allows conclusions on the wear status of the mechanical drive belt used in the gears of the robot arm. The analysis module is deployed separately from all other NodeRed flows and comes in its own Docker container, making it as simple as possible to change the analysis behaviour without interfering with the rest of the data flow. The publication of the pre-processing results and all other communication with the cloud environment is done via http requests through the NodeRed http module. The request is triggered by the ingestion layer of the SERENA cloud system, which includes the received data into the cloud databases and provides it for further long time analysis (Task 2.3) and machine condition forecasting. All communication between cloud and edge is done through the REST application interface endpoint of the cloud ingestion layer. All data transfer through the network is secured by SSL encryption. JSON-LD is the standard message format used in SERENA cloud to edge communication. Each message can contain multiple parameters of sensor data and metadata, which will be divided into two separate repositories in the cloud system storage. If more complex algorithms have to be used to analyse certain sensor signals, it is also possible to transfer the raw sensor signal batch wise to the cloud and therefore outsource the processing load.

### 2.2.2 Design Constraints

The design has to allow flexible exchange of the functionality submodules. Therefore, these functionalities need to be clearly separated. To enable drop-in exchange of modules, each type of software module implements the same interfaces as their previous version.

The same request for universality counts for the transfer of information and data within the SERENA system network. This issue was discussed intensely during the tech-meeting at IPT premises in Aachen during 23rd/24th of May 2018. As data format REST and MQTT were considered by brainstorming the advantages and the suitability regarding the communication requirements of both, see Table 1. Due to its multiple advantages and good suitability for large data files it was decided to use REST in the SERENA communication. However, the design will be held open to integrate MQTT in an effective and easy way if needed in any use case implementation.

**Table 1: Advantage comparison MQTT vs. REST**

MQTT	REST
+ Publish and Subscribe possible	+ Suitable for large data files
+ Common control	+ More general
+ Easier security	+ Decentralized communication
+ Management functions	+ Service discovery
	+ Resilience
	+ running more stable

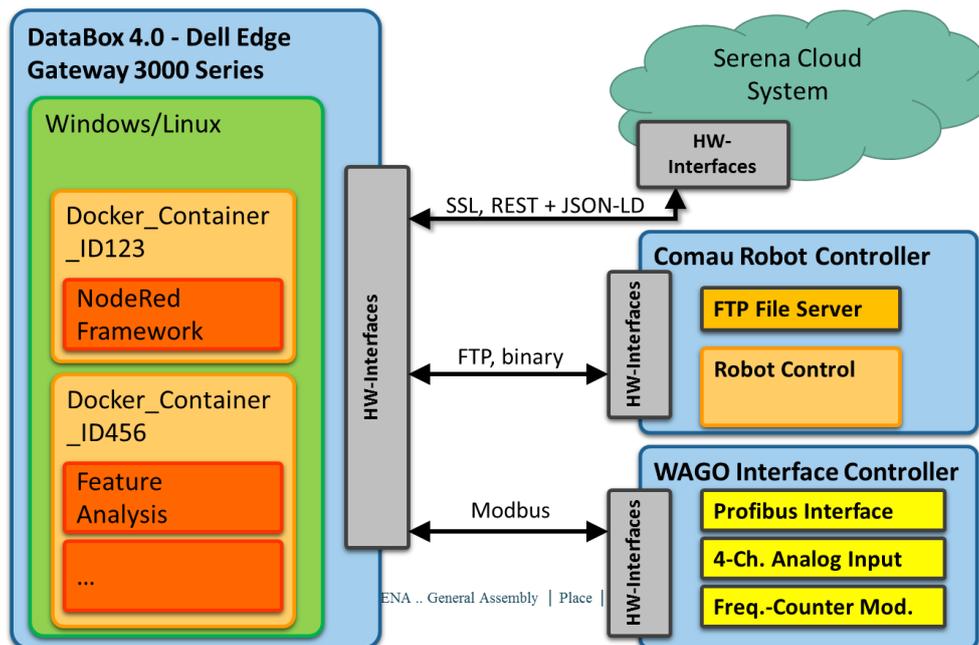
### 2.2.3 Hardware architecture

Figure 2 shows the technical setup of the hardware. This design closely relates to the technical design requirements in the first part of Task 2.1, such as modularity and a large variety of hardware connectivity. The Edge gateway directly accesses and integrates the company network of the use case partner. To ensure full functionality the device must be completely integrated into the company's security systems, firewalls and device management. This allows access to the SERENA cloud running on either company server infrastructure or an external 3<sup>rd</sup> party Cloud of any provider.

Local control infrastructure devices such as PLC's can be connected to the DataBox in two different ways. The first possibility is to connect it directly to a free network port of the DataBox if available and if the device supports a standard network protocol. The second way is to connect it to an interface controller and use this as an adaptor if a direct network connection is not possible for any reason. The WAGO interface controllers used for SERENA, support the concept of flexibility, modularity and extendibility. The controller itself is available for multiple different network protocols and standardized interfaces. Each controller can be equipped with a various number of different I/O

modules, ranging from all kinds of industrial bus systems, to digital/analog inputs/outputs and also direct sensor driver modules. The controller reads out the status of each I/O-module frequently and reports it to the connected DataBox. Due to the large variety of available I/O-modules, these WAGO interface controllers can be used in combination with all kinds of data sources and enable the plug and play concept as stated in Task 2.1 in terms of the used field hardware. This makes it act as universal adapter and ensures the connectivity to each existing use case infrastructure that SERENA solutions are expected to be demonstrated.

To allow network access to all mentioned data sources and devices, the DataBox must be equipped with multiple network interfaces. Depending on the possible network configuration according to the company's network policies, a switch can be used instead.



**Figure 2: DataBox 4.0 Hardware Architecture and System integration**

In the COMAU use case, the DataBox directly accesses the robot controller via network interface to copy the binary file. For the demonstrator of the COMAU use case a Dell edge gateway device of the 3000-series will be used. This can be replaced by and suitable device from other vendors if the hardware requirements are fulfilled. Access to the SERENA cloud system is realised through the company network. This connection is SSL encrypted to enable an easy integration into the existing company network and its security concept. The integration in the company system and local security systems has to be done by the local IT-department.

#### 2.2.4 Software architecture

The use of a Docker swarm enables the required modularity as visualized in Figure 3. The swarm (group of client devices) is managed by one or more swarm managers. These managers update each device as soon as a new version of an already deployed software container or an entirely new container is available and the rollout gets triggered by the system administrator. Using this, new or updated software modules, e.g. for a newer robot controller or sensor interface, can seamlessly be integrated into the system without updating each device manually. Each container holds a special set of Docker labels to specify which edge device it needs to be deployed to by the swarm manager. The software functionalities itself are created using the NodeRed framework. The different NodeRed flows are each embedded into a single Docker container.

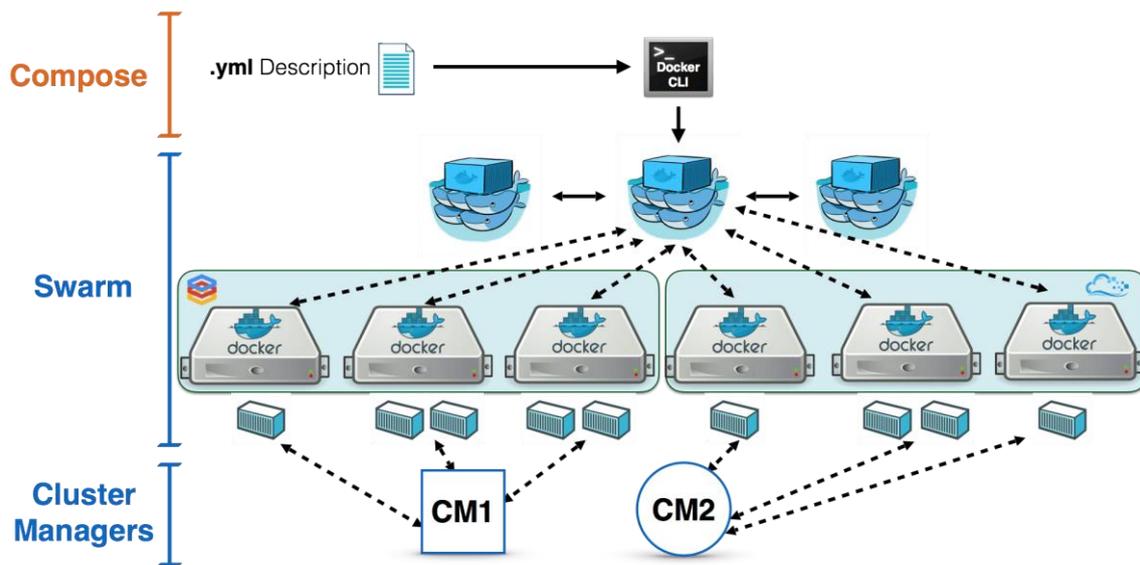


Figure 3: Docker environment [1]

Figure 4 shows the basic software architecture defined during the tech-meeting in Aachen and is the foundation of the versatile data integration software as required according to Task 2.1. Using the microservice-pattern allows flexible handling of new software components and also guarantees a fine granularity of the functionality that each microservice contains. The pattern and its components can easily be distributed to multiple devices connected directly or through a network. Furthermore it does not stipulate any specific protocols or communication mechanisms to be used.

Using REST-compliant (Representational State Transfer) webservices guarantees interoperability between networked devices by offering standardised communication endpoints. REST is commonly known for good performance, reliability and easy extendibility. Especially the extendibility issue makes REST a good choice for non-standard and highly specialized systems like SERENA.

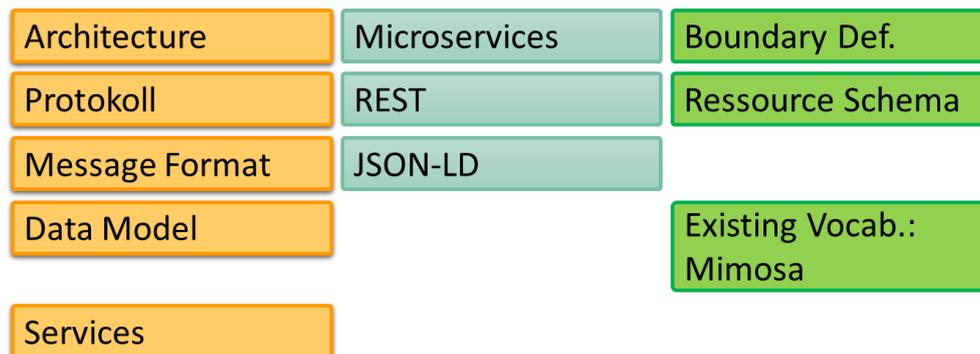


Figure 4: Basic aspects of software infrastructure design

The JSON-LD message format makes it possible to not only transmit data as an attribute-value-combination, but also to bring each dataset into the specific context of a resource. In combination with the MIMOSA data model this gives the opportunity to design and adapt the whole communication set especially for the needs of SERENA.

The described architecture has so be considered when implementing the different services of the SERENA system. A basic list of those services has been created by stepping through the COMAU use case and considering possible solutions, see Table 2.

**Table 2: Basic List of Serena Services**

<b>Service</b>	<b>Possible Solutions / Tools</b>
Logging	
Input: Machines, Sensors, IT-Systems	NodeRed, Custom Code, SPARK
Processing: Analysis, Transformations, Data synchronization	
Human Machine Interface: Inputs, Outputs, Configuration	
Buffer	Database, File
Security Management	Liota, Pulse
Updates	Docker Swarm: Local management Service + Protocol Integration
Process Manager	
Security Management	Handshake (Arrowhead)
Time synchronization	
Service Repository	
Service Discovery	

Using this combination of the aforementioned standards in combination with the Docker environment, completes the plug and play concept (Task 2.1) on the software side. The procedure could be automated entirely as soon as a new sensor or I/O-module gets connected. By detecting the new Sensor or I/O-module with a separate microservice, this will then trigger the update of the concerning DataBox with a newly and automatically composed Docker container. This includes all local code to read the new sensor data, analyse it and forward it to the cloud system. After the automated update the new sensor will be ready for use. The realization of this mechanism is still subject to future discussion within WP2. Especially the automated composition and deployment of the Docker container to deliver the interface and sensor driver could become a challenge due to the large variety of possible new devices that could be connected.



## 3 System interfaces

### 3.1 Human – machine interfaces

#### 3.1.1 Edge Gateway GUI / Operating system

##### 3.1.1.1 Purpose

- Initial setup and configuration of the device
- Maintenance operations that cannot be done remotely

### 3.2 External interfaces

#### 3.2.1 Network interface

##### 3.2.1.1 Purpose

- Connection to cloud system, control hardware in the field and WAGO interface controller
- Configuration access to device

##### 3.2.1.2 Inputs

- Data and metadata from cloud, field control hardware, WAGO interface controller

##### 3.2.1.3 Outputs

- Smart/Meta/Raw-Data to cloud system

#### 3.2.2 Additional Interfaces via WAGO interface controller

##### 3.2.2.1 Purpose

- Connection and power supply of additional sensors, actors and other data sources if needed

##### 3.2.2.2 Inputs

- Multiple possibilities:
  - o Digital/Analog Input and Output
  - o Standard Bus interfaces
  - o Direct driver modules for analog or digital sensors

##### 3.2.2.3 Outputs

- Sensor data as raw data



## 4 Integrity controls

Whenever data of any kind is sampled, acquired, or read from a source, it has to be validated to ensure the quality of the measurement and the derived data. Especially in long term data capturing like done in SERENA, where many different steps of data transformation and analysis are performed subsequently, it is very important to include validation mechanisms into the dataflow.

The most important task for the SERENA DataBox is to ensure the correct timestamping of the data and a proper synchronisation of all component clocks. To realise this in the SERENA system, different protocols and other methods will be discussed to achieve enough accuracy for the timestamping of the collected data.

The validation of each measured/sampled value is another issue. Even in modern measurement systems, outliers occur and have to be detected by a suitable filter algorithm as soon as possible. Which algorithms can be used and included/executed in the DataBox Software is still subject to discussion within WP2.



## 5 Operational scenario

The operation of the DataBox is mostly done by SERENA system maintenance engineers or programmers to adapt or add functionalities that cannot be done remotely. Once running stable, the DataBox does not need any further user interaction. Therefore, the device should be enclosed into a locked housing to prevent it from harmful environmental influences or 3<sup>rd</sup> party manipulation. For the physical installation of new sensors, interface drivers, or connections to other local control infrastructure, the responsible technicians need access to the device.

During normal operation the DataBox collects the data from all connected sources and sensors as defined in the corresponding NodeRed input flow. The pre-processing and analysis is done continuously (continuous data stream) or in batches (irregular availability of new data). This also depends on the processor load caused by the analysis algorithm and the processing power of the edge gateway. The transfer of the results can be triggered by event, by scheduler, or by the ingestion service of the cloud system.



## 6 Conclusion

In SERENA M1-12 a complete concept for the DataBox framework has been designed and partially developed. This includes the design and parametrization of the needed hardware as well as the integration into the existing use case infrastructure and connection to the company IT systems. On the software side, a comprehensive design was created respecting the identified requirements on all layers of communication, data handling, data processing, software deployment and system integration to existing IT systems. This design has been developed along the COMAU use case as decided by the consortium, but as described, is kept open and flexible, not specified, to this use case and enables easy adaption to the other use cases within the SERENA project as well as all future use cases.

This design will be tested in a demo setup with a simplified COMAU robot to evaluate the current status of the development. All further work will be based on the results of this first test while the demonstrator can constantly be updated to test the latest versions of soft- and hardware. The transfer to the other use cases follows as soon as this prototype reaches fulfils the identified requirements concerning portability and functionality.



## References

[1] *devops*. (25. 09 2018). Von <https://devopscube.com/docker-container-clustering-tools/> abgerufen