PROPHESY

Platform for rapid deployment of self-configuring and optimized predictive maintenance services

DELIVERABLE

D4.1 – Automatic data collection v1

Horizon 2020
European Union funding
for Research & Innovation

The PROPHESY initiative has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 766994.

Project Acronym:	PROPHESY
Grant Agreement number:	766994 (H2020-IND-CE-2016-17/H2020-FOF-2017)
Project Full Title:	Platform for rapid deployment of self-configuring and optimised predictive maintenance services
Project Coordinator:	INTRASOFT International SA

This project is co-funded by the European Union

DELIVERABLE

D4.1 – Automatic Data Collection v1

Dissemination level	PUBLIC
Type of Document	(DEM) DEMONSTRATOR
Contractual date of delivery	M15, 31/12/2018
Deliverable Leader	ICARE
Status - version, date	V1.0, 28/12/2018
WP / Task responsible	TUE/ICARE
Keywords:	Data collection; Automated; Data Producer; Data Consumer

Executive Summary

PROPHESY's WP4 is devoted to the implementation of a framework for automated collection of data from various maintenance related data sources and systems, including equipment sensors, production systems, production quality systems, enterprise system, shopfloor devices and more. This workpackage will deliver the data services that will comprise the PROPHESY-ML toolkit.

To this end, the task will devise a service-oriented architecture based on technologies used in industrial engineering (such as REST and Sharepoint), which will enable access to the various systems and subsequently storage of data in appropriate datastores.

The data collection process will be aligned to the PROPHESY-CPS data models and interoperability mechanisms specified in WP3. The data collection framework will provide readily available drivers for industrial systems and production systems such as OPC-UA clients for different types of systems. The collected data in the datastores will provide the basis for implementing the KPIs calculation and visualization tasks in subsequent workpackages. Data persistence in the datastores will be based on the data models (and related schemas) specified and implemented in WP3.

Deliverable Leader:	ICARE
Contributors:	ICARE
Reviewers:	UNPARALLEL & MONDRAGON
Approved by:	INTRA

Document History			
Version	**Date**	**Contributor(s)**	**Description**
0.1	13/12/2018	ICARE	Initial Structure of the Document and contents
0.2	21/12/2018	UNP, MON, ICARE	Corrected comments and additional contributions described
0.3	21/12/2018	PHI	Corrected comments and additional contributions described
1.0	28/12/2018	ICARE	Final version

Table of Contents

Table of Figures

List of Tables

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
AR	Augmented Reality
BOM	Bill of Materials
CPS	Cyber-Physical System
CTQ	Critical To Quality
DMC	Data Matrix Code – Quick Response
DTA	Digital Torque Analyser
ERP	Enterprise Resource Planning
FIS	Factory Information System
FMECA	Failure modes, effects and criticality analysis
I-DNA	Intelligent Decimation Numerical Algorithm
IoT	Internet of Things
MES	Manufacturing Execution System
MTTR	Mean Time To Repair
OEE	Overall Equipment Effectiveness
OPR	Offline Process Recorder
PdM	Predictive Maintenance
pk-pk	peak-to-peak
PLC	Programmable Logic Controller
PROPHESY System	It is the combination of the PROPHESY-CPS and PROPHESY-PdM platform
PROPHESY-AR	PROPHESY-Augmented Reality
PROPHESY-CPS	PROPHESY-Cyber Physical System
PROPHESY-ML	PROPHESY-Machine Learning
PROPHESY-PdM	PROPHESY-Predictive Maintenance
PROPHESY-SOE	PROPHESY-Service Optimisation Engine
RPN	Risk Priority Number
TTR	Time to Repair
UC	Use Case

1 Introduction

1.1 The PROPHESY Vision

Despite the proclaimed benefits of predictive maintenance (PdM), the majority of manufacturers are still disposing with preventive and condition-based maintenance approaches, which result in suboptimal OEE (Overall Equipment Effectiveness). This is mainly due to the challenges of predictive maintenance deployments, including the fragmentation of the various maintenance related datasets (i.e. data "silos"), the lack of solutions that combine multiple sensing modalities for maintenance based on advanced predictive analytics, the fact that early predictive maintenance solutions do not close the loop to the production as part of an integrated approach, the limited exploitation of advanced training and visualisation modalities for predictive maintenance (such as the use of Augmented Reality (AR) technologies), as well as the lack of validated business models for the deployment of predictive maintenance solutions to the benefit of all stakeholders. The main goal of PROPHESY is to lower the deployment barriers for advanced and intelligence predictive maintenance solutions, through developing and validating (in factories) novel technologies that address the above-listed challenges.

In order *to alleviate the fragmentation of datasets and to close the loop to the field,* PROPHESY has specified a novel CPS (Cyber Physical System) platform for predictive maintenance [See Deliverable D2.1], which shall provide the means for *diverse data collection, consolidation and interoperability*, while at the same time supporting digital automation functions that will *close the loop to the field and will enable "autonomous" maintenance functionalities*. The project's CPS platform is conveniently called PROPHESY-CPS and is developed in the scope of WP3 of the project.

In order *to exploit multiple sensing modalities for timely and accurate predictions of maintenance parameters (e.g., RUL (Remaining Useful Life))*, PROPHESY will employ advanced predictive analytics which shall operate over data collected from multiple sensors, machines, devices, enterprise systems and maintenance-related databases (e.g., asset management databases). Moreover, PROPHESY will provide tools that will facilitate the development and deployment of its library of advanced analytics algorithms. The analytics tools and techniques of the project will be bundled together in a toolbox that is coined *PROPHESY-ML* and is developed in WP4 of the project.

In order *to leverage the benefits of advanced training and visualisation for maintenance,* including *increased efficiency and safety of human-in-the-loop processes* the project will take advantage of an Augmented Reality (AR) platform. The AR platform will be customised for use in maintenance scenarios with particular emphasis on remote maintenance. It will also be combined with a number of visualisation technologies such as ergonomic dashboards, as a means of enhancing workers support and safety. The project AR platform is conveniently called *PROPHESY-AR.*

In order *to develop and validate viable business models for predictive maintenance deployments,* the project will explore optimal deployment of configurations of turnkey solutions, notably solutions that comprise multiple components and technologies of the PROPHESY project (e.g., data collection, data analytics, data visualisation and AR components in an integrated solution). The project will *provide the means for evaluating such configurations against various businesses and maintenance criteria,* based on corresponding, relevant KPIs (Key Performance Indicators). PROPHESY's tools for developing and evaluating alternative deployment configurations form the project service optimisation engine, which we call *PROPHESY-SOE.*

1.2 PROPHESY WP4 Overview

PROPHESY's WP4 is devoted to the implementation of a framework for automated collection of data from various maintenance related data sources and systems, including equipment sensors, production systems, production quality systems, enterprise system, shopfloor devices and more. This work package will deliver the data services that will comprise the PROPHESY-ML toolkit.

Its main objectives include:

- To implement a framework for integrating data from multiple (fragmented) data sources, in-line with PROPHESY-CPS data sharing and interoperability techniques.
- To specify physical models of machines and tools, as an invaluable input for specifying and implementing effective predictive analytics techniques for PdM.
- To devise and implement effective data analytics techniques that can be integrated in industrial practice.
- To bundle data collection and data analytics assets in an integrated and reusable toolkit, which will be used in conjunction with PROPHESY-SOE and PROPHESY-CPS.

1.3 PROPHESY Task 4.1 Overview

To this end, the task will devise a service-oriented architecture based on technologies used in industrial engineering (such as REST and Sharepoint), which will enable access to the various systems and subsequently storage of data in appropriate datastores.

The data collection process will be aligned to the PROPHESY-CPS (top view of Figure 3) data models and interoperability mechanisms specified in WP3. The data collection framework will provide readily available drivers for industrial systems and production systems, for different types of systems. The collected data in the datastores will provide the basis for implementing the KPIs calculation and visualization tasks in subsequent work packages.

Data persistence in the datastores will be based on the data models (and related schemas) specified and implemented in WP3.

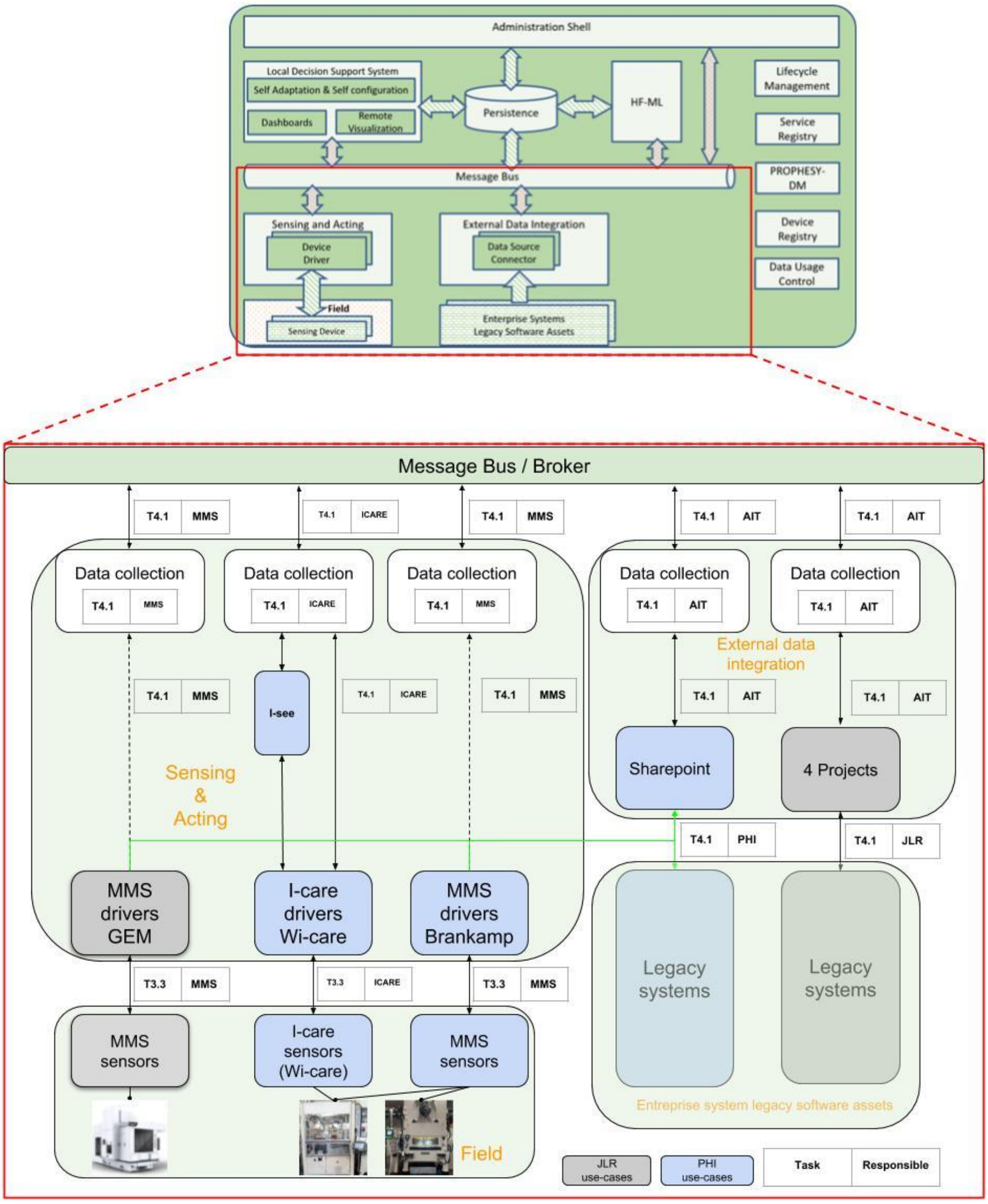


Figure 1 CPS and task 4.1 overview

The data collected from the MMS sensors in the PHI demonstrator (e.g., Brankamp system) will not be fed directly to the Message Bus. The data will instead be collected by the legacy MES system first and then pushed to Sharepoint (green line in Figure 1). In case an HF-ML algorithm is running on an edge device, it will receive sensor data from the Brankamp monitoring system directly.

Each component's functions of Figure 1 are described in Table 1.

Table 1 Component's functions

Components	Function(s) within Prophesy	Related task	Responsible partners
Field - Sensors	The sensors are used to collect physical measurements that will be used to predict remaining useful life of machine wear parts.	3.3	ICARE/MMS
Sensing & Acting - Drivers	The drivers collect raw data from field sensors and convert them into a structured digital data format	3.3	ICARE/MMS
Sensing & Acting - Data collectors	The data collectors act as data producers for the message bus. They collect data automatically from the drivers at a given rate and convert the data according to the Prophesy data model (PROPHESY-DM)	4.1	ICARE/MMS
External data integration - Sharepoint & 4Projects	Collaborative platforms are used to collect and centralise data that comes from various legacy systems (ERP, CMMS, Process data, material data, product data...)	4.1	PHI/JLR
External data integration - Data collectors	Data collectors act as data producers for the message bus. They collect data automatically from the collaborative platforms at a certain rate and convert the data according to the Prophesy data model (PROPHESY-DM)	4.1	AIT
Message Bus	The message bus act as a message delivery tool that provides means to diffuse information from the producers (data collectors) to the CPS consumers.	3.2	AIT /FHG

1.4 Document scope and structure

The current document is structured as follows:

Section 1. **Introduction**: details the document context and purpose.

Section 2. **Sensors data sources:** details the different data sources that have to be collected for each use case.

Section 3. **Legacy systems:** details legacy systems that the PROPHESY-CPS has to connect to in order to retrieve sensors and contextual data.

Section 4. **Automatic data collection demonstration:** provides demonstrations about collecting data and transforming it into a PROPHESY compliant format.

Section 5. **Conclusions**: provides the conclusion of this document and pointed out the next steps.

2 Sensors Data Sources

2.1 Introduction

The purpose of this section is to describe the sensor data sources that should be connected to the automatic data collection framework. By using the “sensor data source” name, it is intended to describe sensors linked to “external” condition monitoring systems, such as MMS and ICARE systems, as opposed to legacy systems. These will be described in the next section of this document.

Even if a brief overview of use cases will be given, the detailed link between the required physical measurements and the corresponding data collection systems has already been made in D3.5.

A mapping between data sources, collectors, use cases and systems (see Figure 2) has been made in order to develop a general framework. This mapping is supposed to describe every data input source, every collection system and every final destination for the whole project. It is a table that will be shared between stakeholders and modified and corrected over the course of the project.

Data input source	Type	Comments	Use-case (UC1 or UC2 or UC3)	Specific Use for (e.g., RUL, Dashboard x, AR glasses,...)	Data Consumers @ CPS Level: HF-ML and/or Dashboards and/or Self-Adaptation and/or Persistence and/or admin Shell	Data Consumers @ PdM Level: to be further defined if PdM is split into smaller service bricks. We don't know	Data Production frequency on the message bus (to be filled by the PROPHESY-ML developers and PROPHESY-AR users)	Data System Source	Data Format (e.g., csv, json)	Data Type	Data Dimension (units) or sample	Sensing System in place i.e., is the sensing system on the production line or at the Use Case level?	Data Acquisition system in place? i.e., there is something that allows to get data from this system to another information system	Data available on Prophesy Sharepoint + date when expected (if relevant)	Count for PHI-2-PROPHESY status calculation	Status PHI-2-PROPHESY
PROGRESS																37%
Material code	Material	Linked to the material being used (strip reference)	UC1, UC2	RUL calculation	?	?	?	Manual input	?	?	?	Done	Done	Done		1
Eddy Current 1-n	Material	Magnetic properties of strip (Eddy current measurement)	UC1, UC2	RUL calculation	?	?	?	MD01	?	?	?	Done	wk1904 - TEG	wk1905 - M-IT		0
Temperature	Material	Is this measurement relevant for RUL? Because it's not appearing in the FMECA	UC1, UC2	RUL calculation	?	?	?	MD01	?	?	?	Done	Done	wk1848 - M-IT		0
Thickness	Material	material thickness	UC1, UC2	RUL calculation	?	?	?	MD01	?	?	?	Done	Done	wk1848 - M-IT		0
Force 1-n	Process	Cutting force measurement (piezoelectric pressure sensor on top of punch head)	UC1	RUL calculation	?	?	?	BKX7.1	?	?	?	wk1848	wk1849 - Brankamp	wk1850 - M-IT		0
Acoustic	Process	Acoustic signal (sensor on guide plate)	UC1	RUL calculation	?	?	?	BKX7.1?	?	?	?	wk1848	wk1849 - Brankamp	wk1850 - M-IT		0
Status	Machine	A Automatic /B Technical stops/C Operator stops/D Organization stops/E Maintenance stops/F No orders/G Starvation/Blocking (No input, No output)	UC1	RUL calculation	?	?	?	MP12	?	?	https://share.philips.com/:w:/s/STS020180523133503/ET62PSb_gUFg62MvELQkSQBhepU0e_GZb9gPikcEeU35A	Done	Done	Done		1
Speed	Machine	RPM of the press motor = strokes per minute	UC1	RUL calculation	?	?	?	MP12	?	?	?	Done	wk1849- Bruderer	wk1904 - M-IT		0
Depth adjustment	Machine	micro-adjustment of the bottom dead center of the ram	UC1	RUL calculation	?	?	?	MP12	?	?	?	Done	wk1849- Bruderer	wk1904 - M-IT		0
Diameter	Product	Diameter of punched hole (in-line CMM-IP measurement)	UC1	RUL calculation	?	?	?	MA80	?	?	?	wk1847 - FMI	wk1848	wk1849 - M-IT		0
Shape accuracy (2D)	Product	Profile accuracy of punched hole (in-line CMM-IP measurement)	UC1	RUL calculation	?	?	?	MA80	?	?	?	wk1847 - FMI	wk1848	wk1849 - M-IT		0
Torque of servo feeding system (electric energy consumption of feeding stroke)	Machine	from the failure modes list sent by PHI --> not sure if we can this out of the machine	UC1	RUL calculation	?	?	?	?	?	?	?	Haije: Check with Bruderer	?	?		0
~~Wear part shape (3D)~~	~~Tool~~	~~offline measurement~~	~~UC1, UC2, UC3~~	~~?~~	~~?~~	~~?~~	~~?~~	~~3D-CMM~~	~~n-a~~	~~n-a~~	~~n-a~~	~~Done~~	~~n-a~~	~~n-a~~		0
Thickness calibrated (off-line measurement)	Product	offline measurement	UC2	RUL calculation	?	?	?	MA67	?	?	?	Done	Done	wk1848 - M-IT		0
Force 1-n	Process	force measurement in the cutout tool UC2	UC2	RUL calculation	?	?	?	BKX7.2	?	?	?	wk1848	wk1849 - Brankamp	wk1902		0
Acoustic	Process	acoustic measurement in the cutout tool UC2	UC2	RUL calculation	?	?	?	BKX7.2	?	?	?	wk1848	wk1849 - Brankamp	wk1902		0

Figure 2 Screenshot of Prophesy Data Sources and Consumers Table

Please note that this figure does not show the entirety of the mapping but only the structure of it.

2.2 PHI Use Cases - UC1

2.2.1 Description and data sources

As previously described in D3.5, use case 1 relates to a cold forming press, shown on Figure 3.

The main purpose is to predict the RUL (Remaining Useful Life) of specific wear parts within a tool assembly.

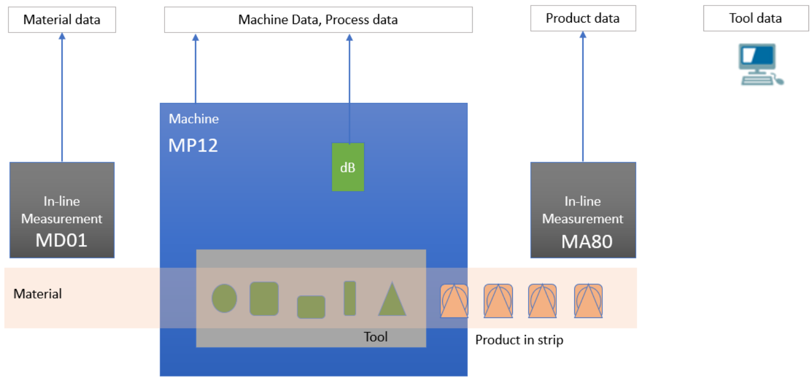


Figure 3 Use Case 1: Cold Forming Press

To date, the following sensors have been installed:

- Pilot hole cutting (module 1)
  - 1x UE10 sensor in top plate
- Calibration unit (module 10)
  - 1x UE10 sensor in top plate
- Existing sensors (data will be fed to Prophesy)
  - 1x Acoustic sensor (die housing)
  - 1x UE13 sensor in ram
- Existing, non-Prophesy sensors (data is not applicable to Prophesy-ML PdM)
  - 2x UE10 sensor in cold forming step 5 (module 5)
  - 2x UE10 sensor in cold forming step 6 (module 6)

UE10 sensors are shear and tension force sensors that measure deviations in resonance frequency. The sensor resonates at the specified frequency. The frequency response changes when more or less force is applied on the body on which the sensor is installed.

Discussion is ongoing about the installation of 5 units of UE10 sensors. It is expected to have 13 sensors installed at Philips UC 1, with 9 sensors being used in PROPHESY.

2.2.2 Data Collectors and Analytics

Sensors for this machine are connected to the Brankamp X5 system and Brankamp X7 system. The Brankamp systems collect data from the sensors and the data are transferred from the Brankamp systems to a local PC via Ethernet connection. At the local PC, a script program is installed, which periodically scans every connected Brankamp system for new process files and start the download and conversion into a txt file. From the local PC, a service is able to export the txt-file data to the PHI Sharepoint platform.

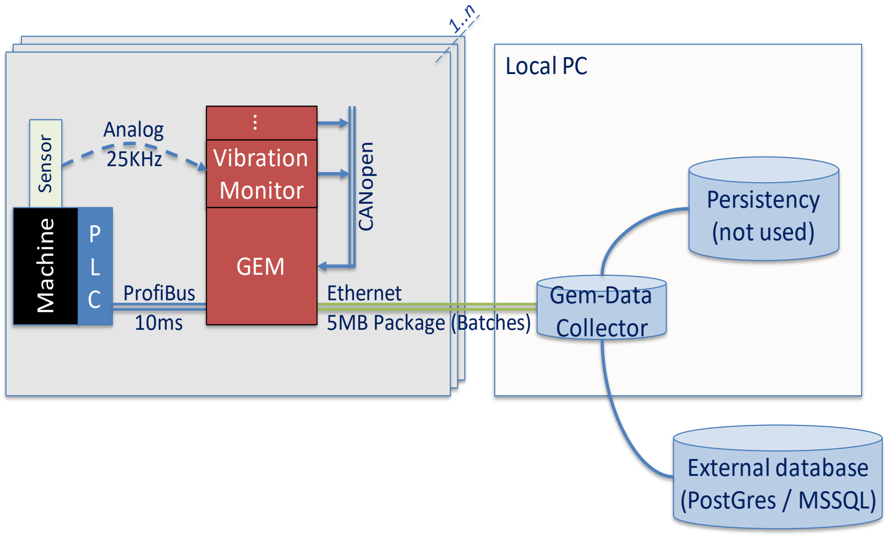


Figure 4 MMS Data collection for UC 1 & UC 2

The next part of the data collection process is to access the Sharepoint where these data are stored in order to retrieve and process it. Analytics methods concerning that machine shall also be implemented.

The description of these next steps will be given in the legacy systems section.

2.3 PHI Use Cases – UC2

2.3.1 Description and data sources

The use case 2 relates to machine UK02 (cutting out machine, Figure 5). The main purpose is to predict the RUL (remaining useful life) of specific wear parts within a tool assembly. A total of 10 wear parts are considered: 5 identical punches (on the lower die half) and 5 identical cutting plates (on the upper die half).

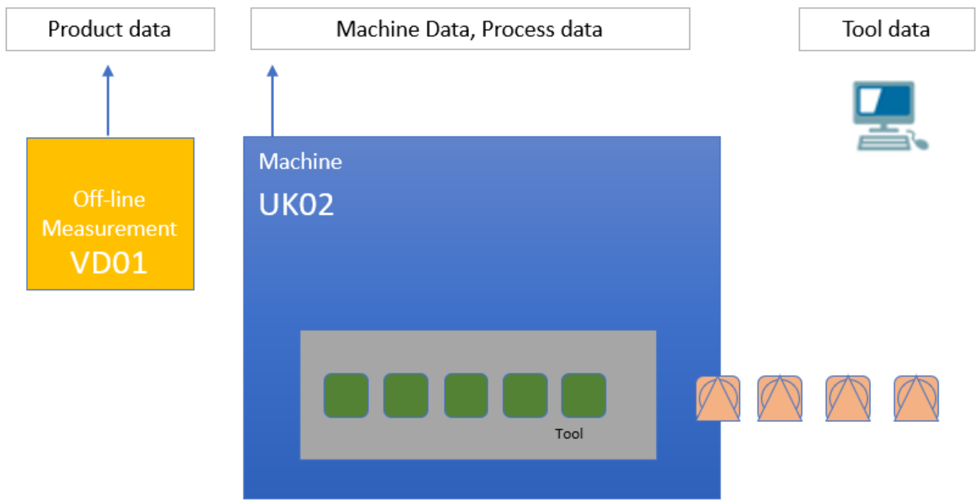


Figure 5 Cutting out machine

To date, the following sensors have been installed or are agreed to be installed on the tool:

- Upper die half
    - 2x UE10 sensor in top plate
    - 1x Acoustic emission sensor under top plate
- Lower die half
    - 5x UE10 sensor in bottom plate
- Press frame
    - 3x UE10 sensor in holding strip right-hand side
    - 3x UE10 sensor in holding strip left-hand side

Finally, a Wi-care wireless system from ICARE has also been set up at the servo drive level, outside of the tool.

In addition to these sensors, 1 extra acoustic emission sensor will be set under the top plate of the lower die half.

Data from each one of these sensors will be collected and fed into PROPHESY.

2.3.2 Data collectors and analytics

Except for the Wi-care sensors, every measurement point shall be collected by MMS using a Brankamp X7.

At this moment, the Brankamp X7 system is installed and sensors are being integrated in the tooling modules. This Brankamp X7 system will be, in a near future, directly connected to PHI legacy systems and thus feed data through the Sharepoint server. The description of these next steps will be given in the legacy systems section.

On the other hand, in order to retrieve data coming from the Wi-care, ICARE developed an API-client able to connect to the I-see platform. This API uses a RESTful architecture and provides an easy-to-use access to Wi-care data.

A script, containing a simple connection to this API, has been developed in order to retrieve information through that API. Every step, regarding how this API works, will be described later in this document (see section 4).

2.4 JLR Use Cases – (UC4/UC5)

The considered JLR production line contains 5 steps (OP30, OP40, OP50, OP70, OP80, OP90, and OP100).

The 'cubed' (pre-machined parts at castings vendor to locate datum points) cylinder head castings enter the EMC machining process at OP30 for valve seats and guide rough machining, which are subsequently cleaned at OP40 and assembled with seats and guides at OP50. Various CNC machining processes including hole generation and milling begin at OP70 on the MAG Specht 600 machines through OP90, followed by cleaning at OP100.

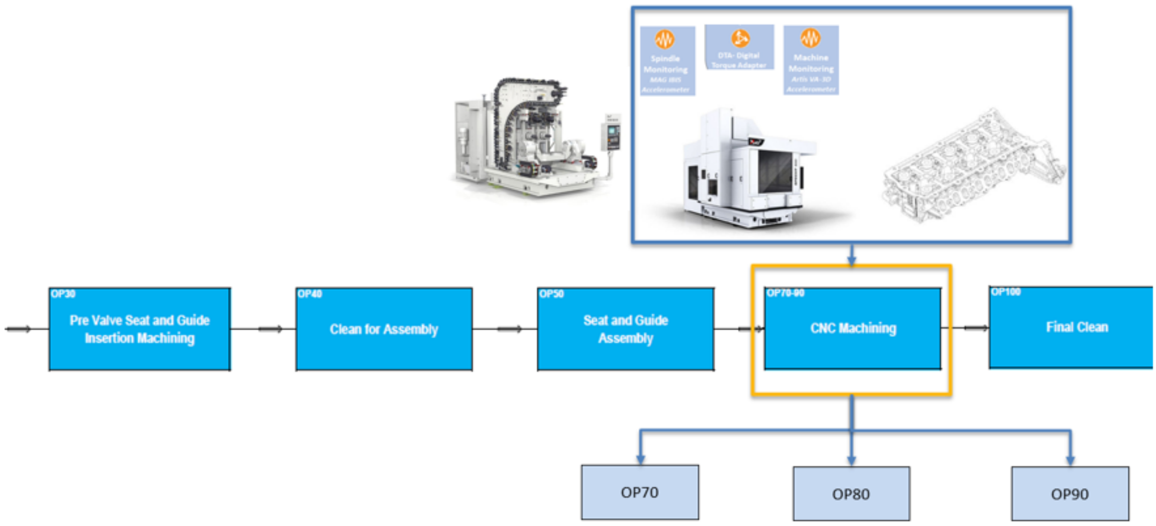


Figure 6 JLR use case production line

2.4.1 Description and data sources

Both JLR use cases relate to OP90 (MAG Specht 600). This is why the description of both use cases will be combined into one section.

The use case 4 focus on the ball screw RUL prediction, while use case 5 focus on the tools RUL prediction.

Figure 7 Example of worn ball screw compared to new ball screw

Figure 8 UC5 JLR tools examples

The various systems linked to the OP90 are represented on the figure below.

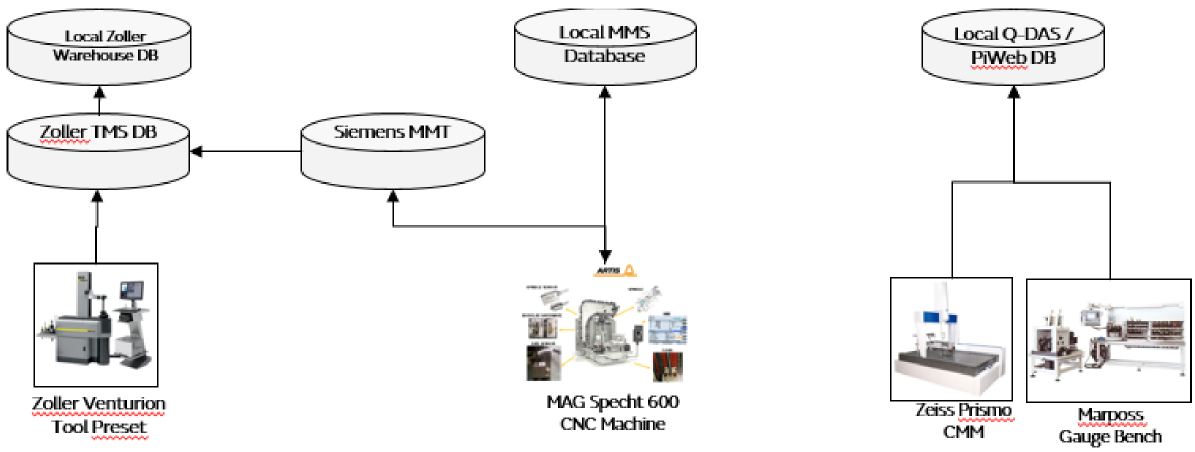


Figure 9 machine and systems

The sensors on the machine at JLR are linked to a Genior Modular (GEM). GEM-Data collector, which is installed at a local PC or server, collects data from the connected GEM using Ethernet connection (each batch contains 5MB of raw data). The GEM-Data collector also converts the binary GEM-Data batches and upload them into an external database (MS-SQL).

2.4.2 Data Collectors and Analytics

To fit the Prophesy middleware architecture, the development of an edge gateway for both use cases is under progress. The first version of this edge-gateway will be implemented in the JLR use cases. The main idea of the edge gateway is to provide a flexible and standardized platform, which is able to connect to different devices (e.g. Brankamp, GEM).

For the connection a specific driver for each device must be developed, which connect to the device using an event-based ethernet connection. The driver also provides a pre-processing function to transfer the sensor data to a Message-Bus.

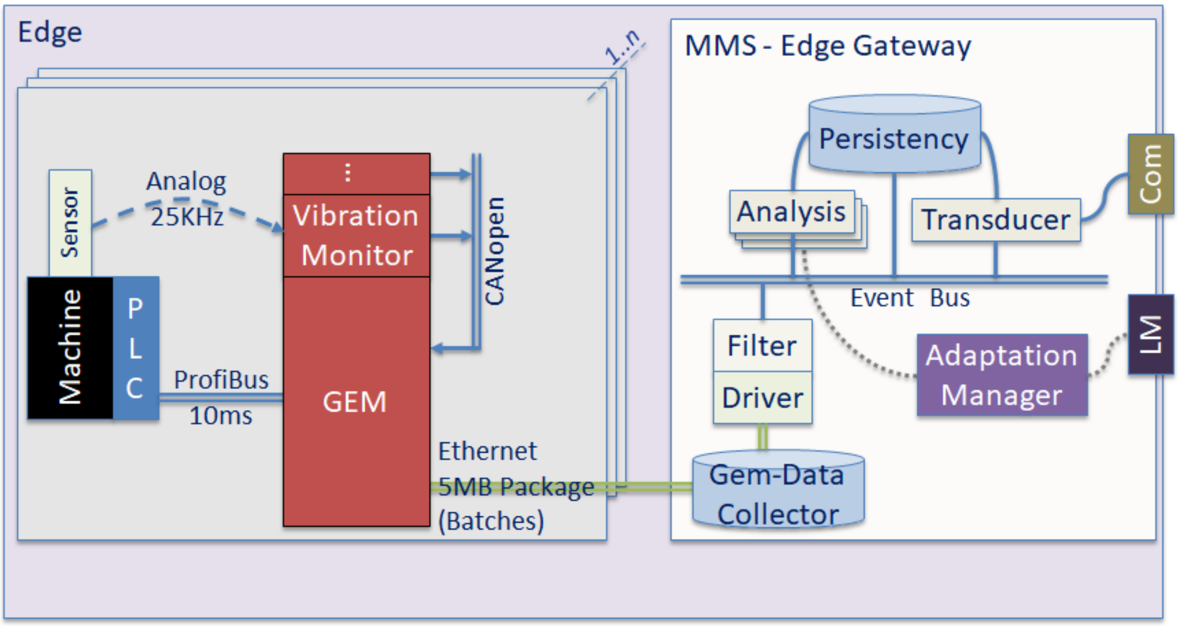


Figure 10 MMS Automated data collection

3 Legacy Systems

3.1 Introduction

This section introduces the context and describes legacy systems that shall be connected to the PROPHESY system in a near future.

As highlighted in previous sections, both PHI and JLR have platforms that they use as an intranet. Information about machines, processes and inventory are stored on premises and thus can be accessed internally.

The purpose of this section is to describe a way for each platform to collect information through an API or any other access that would be granted to the project developers.

3.2 PHI Use Cases – SharePoint platform connection

In order to retrieve sensors and semantics data from PHI Use Cases, it has been decided to connect to PHI SharePoint platform which will be fed with the needed data by PHI.

Microsoft SharePoint allows organizations to create websites and use it as a secure place to store, organize, share, and access information from any device. In its daily use, this platform is accessed by the company's intranet using a web browser of any kind.

Nowadays, this kind of platform also has to be accessed by many different services like required in the PROPHESY project. Hence the RESTful API from Microsoft for their SharePoint server.

Figure 11 SharePoint API Description

This API provides methods that allow developers to do the following requests on the server:

- GET: allows read operations from the server
- POST: allows creation operations on the server
- PUT: allows data modification such as updates on the server.
- PATCH, MERGE: is a POST method containing particular HTTP headers
- DELETE: is a POST method containing particular HTTP headers

Many functions and arithmetic operators are also available when using this API. These should help sort data and might be useful at the pre-processing phase of the data flow. Table 2 provides some examples of API functions for the SharePoint platform.

Table 2 Examples of API functions

Function	Description	Example
bool substringof(string searchString, string searchInString)	Returns a boolean value stating if the value provided in the first argument is a substring of the second argument. Can be used as a replacement for the contains method.	substringof('Alfreds',CompanyName)
bool endswith(string string, string suffixString)	Returns a boolean value declaring if the string provided in the first argument ends with the string provided in the second argument.	endswith(CompanyName,'Futterkiste')
bool startswith(string string, string prefixString)	Returns a boolean value declaring if the string provided in the first argument starts with the string provided in the second argument.	startswith(CompanyName,'Alfr')
int length(string string)	Returns an integer value representing the length of the string provided as argument.	length(CompanyName) eq 19
int indexof(string searchInString, string searchString)	Returns an integer value representing the index of the string provided in the second argument, which is searched within the string provided in the first argument.	indexof(CompanyName,'lfreds') eq 1
string replace(string searchInString, string searchString, string replaceString)	Replaces the string provided in the second argument with the string provided in the third argument, searching within the first string argument.	replace(CompanyName,' ', '') eq 'AlfredsFutterkiste'
string substring(string string, int pos)	Returns a substring of the string provided in the first argument, starting from the integer position provided in the second argument.	substring(CompanyName,1) eq 'lfreds Futterkiste'

These functions and their names might change, depending on the kind of client that is used to implement the API connection.

In a near future, the architecture shown on Figure 12 shall be implemented:

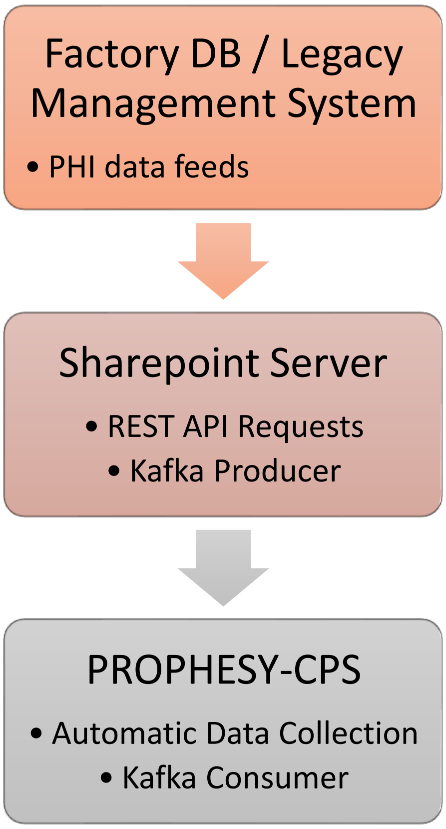


Figure 12 PHI Data collection architecture

However, there might be different data collectors, depending on the type of information that should be retrieved. The data mapping that was shown in Figure 2 covers this part in more details.

3.3 JLR Use Cases – 4Projects platform connection

In order to retrieve sensors and semantics data from JLR Use Cases, it has been decided to connect to their Viewpoint For Projects™ platform (4Projects) which will be fed with the data.

4Projects is a collaboration solution for project management and documents sharing. This cloud-based document control solution helps contractors alleviate project uncertainties by streamlining documentation, reducing errors, mitigating risks, and avoiding duplication of efforts.

A short description of the 4Projects API is given below.

SOAP XML Web Services that provide Create, Retrieve, Update and Delete operations for the following functionality:

- Personal Containers
- Unregistered Files
- Enterprises (Retrieve functionality only)
- Sites
- Projects
- Document Containers
- Documents
- Task and Discussion Containers
- Tasks and Discussions
- SVC's and DVC's
- Transmittals (formerly: Issue)
- Notifications
- Batch Printing

Along with the following capability to retrieve read only information about contexts:

- A tree to retrieve containers in a given context so that developers can navigate through an enterprise structure.
- A query capability to list all items in a context (and all contexts below the context if required).

Generally, to use 4Projects API services directly, you will need to:

- Generate proxy classes and configuration settings from the API Web Service Description Language (.wsdl) files.
- Create instances of the info classes specific to the entity you are dealing with (i.e. if you are creating a new personal container then create a new instance of

PersonalContainerInfo class generated from the wsdl files, if you are updating an existing personal container then call retrieve on the personal container service to retrieve a populated *PersonalContainerInfo* class)

- Modify the relevant properties on the info class.
- Pass the info class back to the relevant service (i.e. the Create or Update on the *PersonalContainer* service) along with a valid token to save the new values to data.

The API operations all follow this pattern:

- Create and Update
  - Receives: Specific message request containing specific info objects in the body and the standard token in the header.
  - Returns: Standard message response with only *OperationResults* in header.
- Retrieve
  - Receives: Standard message request containing Guids in body and token in the header.
  - Returns: Specific message response with specific Info objects in the body and *OperationResults* in the header
- *RetrieveDefaultInfo*
  - Receives: *IDMessageRequest* containing Guids of the container you are creating the item in or the thing you are revising
  - Returns: Specific populated info objects containing things such as custom fields in the context, keywords, workflows etc.
- Delete
  - Receives: Standard message request containing Guids in body and token in the header.
  - Returns: Standard message response with only OperationResults in the header.

All the API's Create, Retrieve, Update and Delete operations take the same format that they allow for collections of IDs or objects as input and return collections of operation results and objects if applicable.

4 Automatic data collection demonstration

This section is intended to link each data source with the part of the framework that will collect data from the data source. The purpose of each collection component will be clearly identified.

These components will be described for each use case, following the same logic as the D3.5 deliverable.

Moreover, analytics functions that were implemented in order to pre-process data coming from these sources will also be detailed here.

4.1 Philips use case 1 – Cold forming tool

A Brankamp branded MMS X5 system is already installed at the demonstrator side (UC1). To allow an extended process monitoring by integration of further sensors and interfaces to PROPHESY, an upgrade to Brankamp branded X7 systems is necessary.

Figure 13 MMS Brankamp X7

Figure 13 shows a Brankamp X7 system as well as some possible process curves. The X7 system allows up to 24 channels for an extended process monitoring. The HMI part of the system runs on a Windows Operating System, so that an easy connection to other PROPHESY parts is possible.

The X7 system has different options to transmit the process data to the PROPHESY platform. For the Philips demonstrator, a standard PC (Edge-PC) will be used to gather all data from the X7 and provide the data in a secure way to PROPHESY. The HFML as well as further data processing could be performed at this Edge-PC.

4.2 Philips Use-case 2 – 5-fold cutting

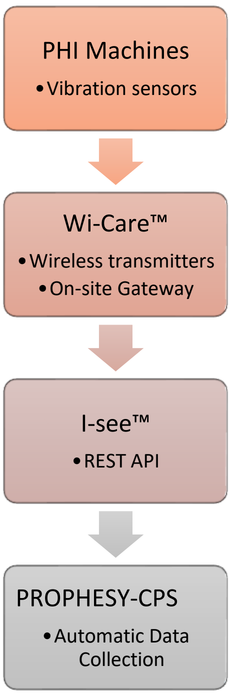
As described earlier, this use case has two different data sources that are related to it.

The first one is the one related to the MMS system, which was already described in the previous section. The part of this demonstrator that shall be described in this section is related to the cloud/remote connection.

Indeed, an automatic routine shall be done in order to retrieve data from this system.

The first possibility that has been considered is to automatically push data from this system to this local legacy system, which is a Sharepoint server. A description about that process shall be given in a future version of this document.

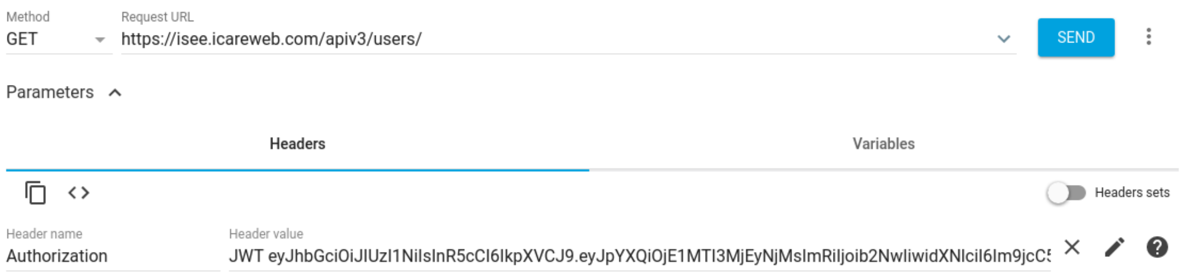
The second data source, related to the I-CARE system, has been provided with an API in order to automatically collect data from it. A figure describing the components around that collector is given below.



Here are the steps that are needed in order to use that API:

a) POST request, with username and password, to get a token

To start working with the API, a HTTP POST request has to be sent to this URL: https://isee.icareweb.com/apiv3/login, containing a username and a password in a JSON format. If the provided username and passwords are valid, then a token should be received in response to this POST request.
Here is an example with the format of the token that should be received:



This token has to be sent, as a part of the header, in every next HTTP request to the platform.

b) GET request with the previously received token to get another token

Once the first token has been received, a GET request to this URL: https://isee.icareweb.com/apiv3/login/databasename, containing the first token, shall be done. Version implementing actions of T4.1 within the PROPHESY framework.

The response to that request should contain a new token related to that particular database.

c) GET asset list with your token

By executing a GET request to this URL: https://isee.icareweb.com/apiv3/assets/ , using the last received token as headers, it is possible to retrieve every asset that is present in the database structure because it should be contained in the server's response under the form of a unique asset id.

```
(base) C:\Users\ADE\Documents\GitFolder\prophesy-datacollector\prophesy-data-connectors\src>python main.py
Prophesy MP1
5accd34975c9f309c60a990b
Prophesy MP3
5accd74875c9f30a120a984a
Prophesy MP4
5accd83475c9f30a3f0a9885
Please write the asset you want to get data from :Prophesy MP1
```

Figure 14 Get Request for Assets List - Screenshot of Results

In the above picture, a first run of the client that I-CARE provided to the project is shown. When executed, the script first displays a list with every asset that is available in that database (assets ids have been highlighted in yellow and are below assets names).

d) GET real data from a particular/numerous asset(s).

Continuing with this first client run, once the chosen asset has been manually entered, results that are related to this particular asset should be retrieved under the form of a JSON response (see Figure 15).


```
"_updated": "Tue, 13 Nov 2018 23:27:57 GMT",
"acqend": "Tue, 13 Nov 2018 23:27:57 GMT",
"alarm": {
    "status": 0,
    "code": 0
},
"globals": {
    "vib": {
        "ave12": 0.06533493041992187,
        "dna12": [
            0.140380859375,
            0
        ],
        "stickslip": 2.0345052083333335,
        "dna500": [
            0.27618408203125,
            0
        ]
    },
    "tempint": null,
    "temp": null
},
"_links": {
    "self": {
        "href": "/apiv3/assets/5accd34975c9f309c60a990b/results"
    }
},
"asset": "5accd34975c9f309c60a990b",
"_created": "Tue, 13 Nov 2018 23:27:57 GMT",
"acqinfo": {
    "factor": 400
},
```


Figure 15 Example of json response from GET request

In this JSON response, different keys can be observed but the “global” one is the one containing acquisition data.

4.3 Pre-processing functions

This part describes the process to collect data from the Sharepoint and use it to fill the dashboards. As a middleware, the client that has been developed will use analytics functions to transform original .csv file to a JSON file which is PROPHESY-DM compliant.

Analytics concerning that part will mainly consist in extracting the right data which will be useful for final dashboards from the original files, such as to represent the cost for a maintenance operation on a specific tool. The goal is to match data required for the dashboards with data provided on SharePoint. The first task is to identify the data input flow from the Sharepoint files to Dashboards inputs and identify missing information on the Sharepoint side of the flow.

After all this matching of data, information needs to use a specific format on the Kafka bus. This format is called the "PROPHESY-DM infrastructure" and has been driven by the data integration and interoperability requirements of the project. For more information about PROPHESY-DM, please see D3.3.

Moreover, the background technologies of the partners, such as Kafka and XML formatted document, and their capabilities have been considered to create it. For the first version of these pre-processing functions, the client will follow the template that was derived from this model but it is important to keep in mind that it might evolve later, regarding usability and performance.

All the process described above is called, according to Kafka terms, "Producer" and can be pictured as shown on Figure 16.

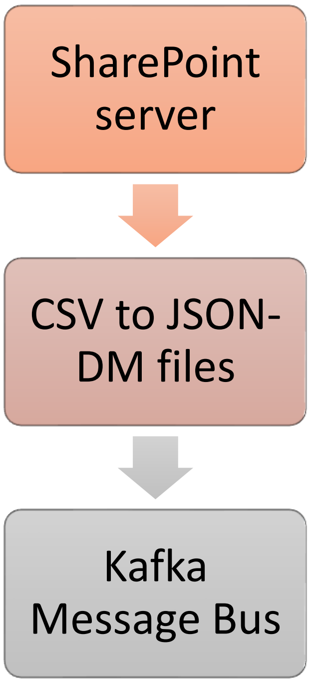


Figure 16 Data Producer Schema

Now that data is present on the bus in a compliant format, the second task consists in using it to fill dashboards. The client that will perform this task will be called the "Consumer" part of the data flow.

Next, to collect messages formatted as described by the PROPHESY-DM from the bus and consume it, the message should be a JSON file on which other analytics functions can be performed.

Analytics concerning this part will take place by using functions to transform data and process it to retrieve compatible information with the dashboard's library used to show different KPI on different tools.

All the process described above called "Consumer part of the project" is summarized on Figure 17.

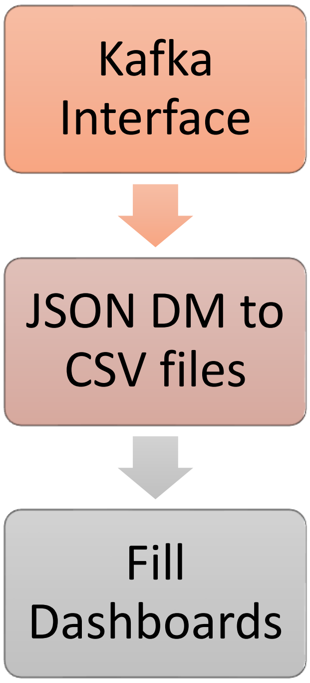


Figure 17 Data Consumer Schema

Since there is no possible way yet to have a direct interface between SharePoint and our client in v1, the temporary solution consists in retrieving data from a PROPHESY Database, which is then populated with data using a personal login with a two-step authentication system (i.e.: password with an additional random code generated by a MobilePass application on a smartphone).

5 Conclusion

PROPHESY's WP4 is devoted to the implementation of a framework for automated collection of data from various maintenance related data sources and systems, including equipment sensors, production systems, production quality systems, enterprise system, shopfloor devices and more. This work package will deliver the data services that will comprise the PROPHESY-ML toolkit.

A framework for integrating data from multiple (fragmented) data sources, in-line with PROPHESY-CPS data sharing and interoperability techniques is being built. At the moment of writing this document, various API connectors are under development and some of them have been finalised.

Physical models of machines and tools, derived from collected field data, will be specified in a near future as an invaluable input for specifying and implementing effective predictive analytics techniques for PdM. Effective data analytics techniques, such as sorting techniques for raw files, have been implemented and shall be used within others parts of the project.

The bundling of data collection and data analytics assets in an integrated and reusable toolkit, which will be used in conjunction with PROPHESY-SOE and PROPHESY-CPS, shall be done after the first demonstration.

During the 15 months of the project, data exchange models have been considered creating an automated data collection scheme that would meet the project's needs without violating IT policies at PHI and JLR. A solution was collectively agreed. This solution would use approved cloud platforms with secure and controlled access, enabling the automated data collection demonstration outside of the end user's factories and enabled data access to other task leaders. It was also demonstrated that the automated data collection process could follow the PROPHESY-DM, thus contributing to the use cases at demonstrators at JLR and PHI through PROPHESY-ML and PROPHESY-AR.

It is expected that the next version of this deliverable will provide an even more automated data collection scheme that will use the latest implementation of the PROPHESY-DM and that will account for potentially new sensors integration and feedback from the remaining PROPHESY bricks: PROPHESY-ML, PROPHESY-AR, PROPHESY-PdM, PROPHESY-CPS and PROPHESY-SOE.