PROPHESY

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

DELIVERABLE

D2.3 – PROPHESY-SOE Specifications

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 optimized predictive maintenance services
Project Coordinator:	INTRASOFT International SA

This project is co-funded by the European Union

DELIVERABLE

D2.3 – PROPHESY-SOE Specifications

Dissemination level	PU – Public
Type of Document	(R) Report
Contractual date of delivery	M08, 31/05/2018
Deliverable Leader	UNPARALLEL
Status - version, date	Final, v1.2, 13/07/2018
WP / Task responsible	WP2 / UNPARALLEL
Keywords:	PROPHESY-SOE; Service Bricks; Turn-key solutions

Executive Summary

The purpose of the deliverable D2.3 – PROPHESY-SOE Specifications is to provide an overview of the PROPHESY-SOE and a detailed specification of its components. After a brief introduction to the main components of the PROPHESY project and a definition of the audience, the document presents a scope and an overview of the PROPHESY-SOE layer, describing the scope of the component, its purposes and an overview to its concept.

A chapter dedicated to the PROPHESY-SOE components describes their concept components, the service bricks, the turn-key solution composing tool, the KPI specification, the business models and the cost-benefit analyses. The service bricks description includes its main features, how they should interact and be classified. The turn-key composing tool should manage to create the turn-key solutions based on service bricks and KPI specifications. The KPI specification presents a description of what should be a KPI purpose. The business model section bases its information in several definitions essentials to the specification of a use-case business models. Key concepts as the Client/Beneficiaries, Goals/Scope, Solutions Constituents, Consultant/Value chain role and Business model are defined to enable a better understanding about the role that the business models take in the architecture. And finally, the cost-benefit analysis presents a description of what they are and how can become valuable to the user.

Deliverable Leader:	UNPARALLEL Innovation, Lda
Contributors:	UNPARALLEL, ICARE, MAG, AIT, OCULAVIS, JLR, PHILIPS, INTRASOFT
Reviewers:	ALL
Approved by:	ALL

Document History			
Version	**Date**	**Contributor(s)**	**Description**
V0.1	10/04/2018	UNPARALLEL	Initial ToC
V0.2	26/04/2018	AIT	Update structure and the first two sections
V0.3	9/05/2018	UNPARALLEL	Scope and overview update
V0.4	15/05/2018	AIT	Introduction update and cost-analyses contribution
V0.5	23/05/2018	ICARE	KPI specification update
V0.6	31/05/2018	OCULAVIS, ICARE, JLR, PHILIPS	Update to the Service brick list
V0.7	18/06/2018	UNPARALLEL	Update SOE components &Include appendix and graphs
V0.8	26/06/2018	MAG	Business Models specification
V0.9	27/06/2018	INTRASOFT, AIT	Service brick packaging and deployment
V1.0	02/07/2018	UNPARALLEL	Version ready for review
V1.1	11/07/2018	ALL	Feedback from review process
V1.2	13/07/2018	UNPARALLEL	Final version

Table of Contents

List of Figures

List of Tables

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
AM	Asset Management
AR	Augmented Reality
ERP	Enterprise Resource Planning
FMECA	Failure Mode Effects and Criticality Analysis
IRR	Internal Rate of Return
KPI	Key Performance Indicators
PdM	Predictive Maintenance
ROI	Return-On-Investment
SOE	Service Optimization Engine
CPS	Cyber-Physical System
ISA	International Society of Automation
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 Optimization Engine
RAModel	Reference Architectural Model
SotA	State-of-the-Art
WP	Work Package
WS	Web Service

1 Introduction

1.1 The PROPHESY Concept

Despite the proclaimed benefits of predictive maintenance, the majority of manufacturers are still disposing with preventive and condition-based maintenance approaches which result in suboptimal OEE (Overall Equipment Efficiency). 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 visualization 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 will specify a novel CPS (Cyber Physical System) platform for predictive maintenance, 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 visualization 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 customized for use in maintenance scenarios with particular emphasis on remote maintenance. It will be also combined with a number of visualization technologies such as ergonomic dashboards, as a means of enhancing worker's support and safety. The project's 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 turn-key solutions, notably solutions that comprise multiple components and technologies of the PROHPESY project (e.g., data collection, data analytics, data visualization and AR components in an integrated solution). The project will **provide the means for evaluating such configurations against various business 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 optimization engine, which we call **PROPHESY-SOE**.

1.2 Document Purpose and Audience

This document is providing the main specifications that will drive the detailed design and implementation PROPHESY-SOE in WP6. Therefore, the target audience of the deliverable includes:

- **The research and IT partners of the consortium**, which will engage in the IT-based development/implementation of the PROPHESY-SOE.
- **Industrial partners with an interest in using the PROPHESY-SOE** for business modelling and evaluation of the project's results, including the two end-users of the project (PHILIPS, JLR).
- **Researchers and maintenance solution providers (including third-parties)**, which would like to understand the scope and usage of PROPHESY-SOE in the evaluation, optimization and exploitation of the project's results.

1.3 Document Structure

The current document is structured as follow:

- Section 1. **Introduction**: details the document context, purpose and intended audience, as well as, the overall strategy applied in the WP2 while underlining the role played by this document with respect to the whole project;
- Section 2. **Scope and Overview**: This section presents the role and scope of the PROPHESY-SOE layer. A conceptual module which tries to divide the layer into simpler problems is presented;
- Section 3. **PROPHESY-SOE components**: Presents the five components of the service optimization engine, giving examples and specifications to their behaviours;
- Section 4. **Conclusion**: The conclusion presents a description of the developed work and presents the work package 6 which will continue the work of this document;
- Section 5. **Appendix A**: Table with the results from the questionnaire about the service bricks that will be implemented.

2 Scope and Overview of PROPHESY-SOE Layer

2.1 Role & Purpose of PROPHESY-SOE

PROPHESY aims at lowering the barriers for the development and deployment of novel predictive maintenance solutions, through alleviating common problems faced by manufacturers and solution integrators in their efforts to adopt and fully leverage predictive maintenance solutions. In particular, PROPHESY designs and develops a wide range of components that are destined to address issues such as:

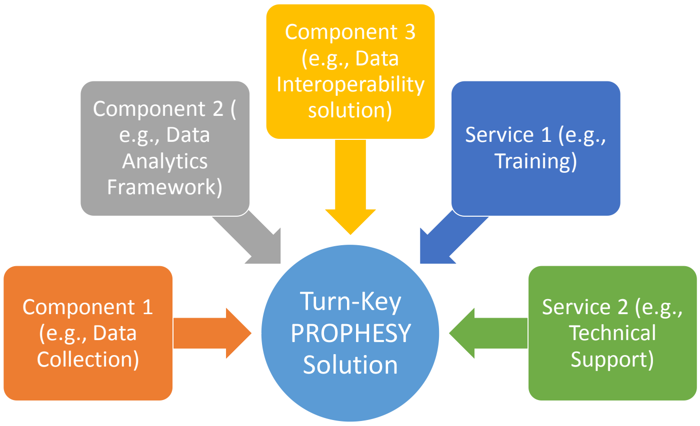
- **The reliable and high-performance collection of maintenance-related data**, including both static data and dynamic data stemming from machines, sensors and field devices.
- **The unification of datasets that are fragmented across different "siloed" systems and databases** such as ERP (Enterprise Resource Planning) systems, MRP (Manufacturing Resource Planning) systems, AM (Asset Management) systems and more.
- **The deployment of advanced data analytics algorithms that are able to identify maintenance insights and failure modes** in-line with an enterprise's FMECA (Failure Mode Effects and Criticality Analysis).
- **The ergonomic and intelligent visualization of maintenance data and maintenance information**, as a means of supporting workers in a wide range of human-in-the-loop scenarios. The project's visualization solutions including Augmented Reality (AR) solutions for training and remote maintenance.
- **The specification and validation of viable business models** for the deployment and operation of predictive maintenance solutions.

As such PROPHESY develops and validates a wide range of components, which can be considered as "building blocks" of integrated maintenance solutions. Therefore, the deployment of a PROPHESY solution entails typically the integration of various building blocks in a "turn-key" solution. Likewise, a turn-key solution is likely to be supported by complementary assets that will be provided by the PROPHESY partners, such as technical support and training services. Overall, PROPHESY is not an "all-of-nothing" solution that mandates the deployment and use of all the components that are developed in the project. Rather, the project will provide manufacturers and solution providers with flexibility in terms of the components that they will deployed, towards composing a solution that meets their needs.

The composition of turn-key PdM solutions from individual PROPHESY components and related support services will be driven by a range of different parameters and practical constraints, including:

- **The needs of the end-users (i.e. manufacturers) in terms of PdM**, which will define the main characteristics of the turn-key solution. For example, manufacturers may or may not need an AR solution, which leads to alternative configurations for a PROPHESY deployment.

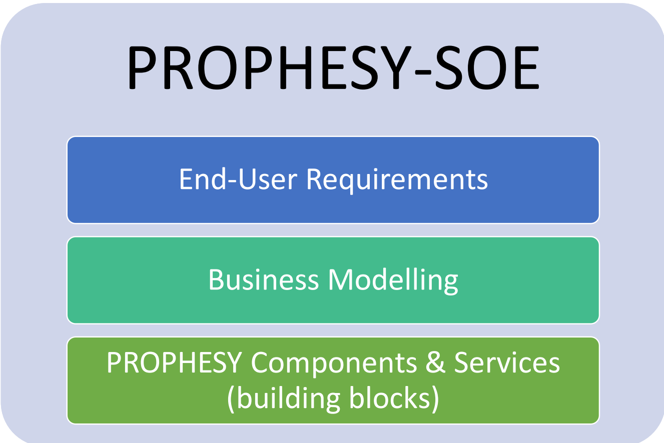
- **The technical viability of the integration of the different components**, which imposes certain constraints about which components can be integrated and how.
- **The business viability of the turn-key solutions,** which shall be driven by a viable business model that leads to positive Return-On-Investment (ROI) for the deployer of the solution (i.e. manufacturer or solution integrator).



Hence, there are different ways to combine PROPHESY components in turn-key integrated solutions, based on various composition scenarios and deployment configurations of the PROPHESY components. In this context, the purpose of the PROPHESY-SOE engine is to provide the means for composing individual components and services (i.e. "building blocks") to turn-key solutions in a way that is technically feasible and viable from the business viewpoint. As its name indicates, PROPHESY-SOE is destined to facilitate the development of optimized configurations i.e. configurations that meet end-users' needs in an optimal way i.e. subject to the optimization of certain technical (e.g., performance, latency) or business (e.g., ROI, IRR) criteria.

The production of optimal configurations will be based on the following parameters and artefacts:

- The end-user requirements that should be respected at all times.
- Appropriate business modelling that yields a positive and acceptable ROI.
- The available building blocks and their functionalities, as the latter will be developed in the scope of the project.



These parameters represent inputs to the PROPHESY-SOE. The outputs of the engine will then include a set of turn-key solutions definitions and configurations, including the building blocks to be deployed and integrated.

There is also a close linking between PROPHESY-SOE and the exploitation of the project's results, as part of the project's ecosystem. This linking is reflecting on:

- The fact that the cost-benefit analysis and KPI calculation tools of the PROPHESY-SOE will be made available to third-parties that will endeavour to develop or deploy predictive maintenance solutions based on PROPHESY components.

- The fact that the PROPHESY-SOE will provide the means for combining multiple PROPHESY components in turn-key solutions in-line with the project's business modelling results. Hence, PROPHESY partners will rely on PROPHESY-SOE in their efforts to commercialize the project's components and solutions.

2.2 PROPHESY-SOE Concept

The PROPHESY-SOE main objective is to be a delivery platform which provides a single-entry point to all the PROPHESY services. Therefore, the next figure represents the concept of the PROPHESY-SOE component which is composed of several other parts that combined will allow a real usage of the platform itself.

The main structure is composed of three parts, the model definition which combines brick services with the KPI specifications and generates the turn-key solutions, the deployment part which is the implementation of a turn-key solution in a real case and the analytics phase which combines the results of the application of turn-key solutions.

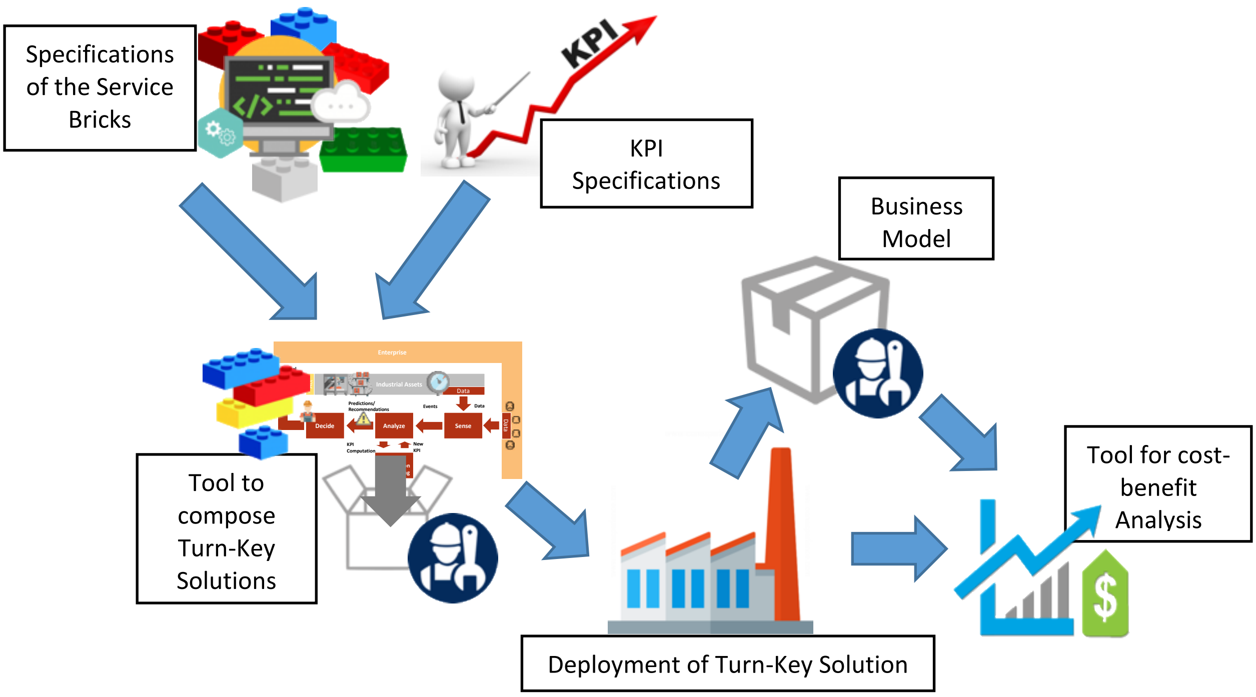


Figure 1: PROPHESY-SOE Ecosystem Concept

The first part which is responsible to create the turn-key solutions, combines service bricks, atomic services which were partners contribution and will be available at the platform, with KPI specifications, which are mathematical formulas or algorithms that enable smart data analytics. With a specific tool, turn-key solutions are generated as a recipe to apply to real-world problems.

The second part is composed of the deployment of the turn-key solution in the physical locations with specific considerations and implementations due to local conditions, targets, and limitations. Each deployment has an associated business model that describes all the deployment variables and business-related features.

The last part of the concept is the analytic module, which with deployments data and business models will compose cost-benefit analysis that will allow future interested users to analyse the efficiency and the return of the investment of the turn-key solution.

3 PROPHESY-SOE Components

3.1 Service Bricks

3.1.1 Overview and Definition

A service brick should be an independent service which can have or not inputs and outputs. Considering it as a black box with optional I/O interactions opens the possibilities of combining two or more of them to composite complex operations over data. The intended behaviour should become as close as it can from the Lego pieces design, where despite every block differences in shape and size they all come together to generate bigger and more complex structures with useful meanings due to their standard interaction/interface which fits all the components.

There are several aspects that should be considered when defining such a standard node. One of them is the communication standard or data format which each service brick uses to communicate. In order to give an easy to understand example, the data standards will be replaced by letters, in the Figure below two service bricks which use different standards can't directly connect to each other. The service brick 1 (SB1) uses the standard A to communicate his output, but the service brick 2 (SB2) only receives data in the B data format. It's needed an intermediate brick which translates between data formats.

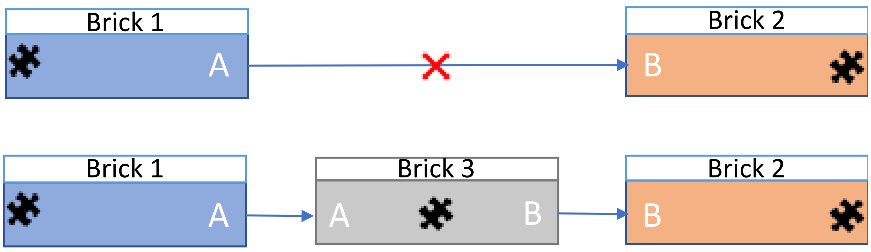


Figure 2: Service brick interaction example

Another fundamental parameter in the service bricks specification is the interaction type it provides or expects. That means each module can have its communication method. It could be a frequency of sending data (streaming/subscription) or a passive data source (request and response) as shown in Figure 3. Such specification will enable the other nodes how to connect and communicate with each other. For example, a connector brick which streams is output with a specific or configurable frequency must be connected to an element which can handle this active input, another way the modules will not communicate properly. On the other hand, a brick which only sends data when its requested, needs to be connected to an active input receiver which will query data when it's needed.

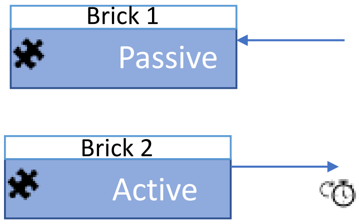


Figure 3: Service brick interaction type draft

3.1.2 Template for service bricks specification

The service bricks are fundamental pieces in this architecture. In order to achieve a better model of the services it was developed and distributed by partners a questionnaire with the purpose of creating a first overview of the service bricks, their needs, and types. The file has six columns each one with a relevant characteristic of the service bricks. Down below, a table describes the fields and its purpose.

Table 1: Service bricks questionnaire fields

Field	Description
Service Brick	Name of the service
WP	Work package where the service will be implemented
Category	The purpose of category is to group the services by its main function, if it is a middleware, a connector, an analytics algorithm or other
Input	Which inputs are needed by the service to work
Output	Which are the outputs of the service
Description	A brief description of the service

In order to consolidate all the service types that could outcome from the questionnaire, there was defined some categories to classify the service bricks. The categories were chosen as generic as possible and try to preview all the service needs. The categories lists have only six distinct classes to reduce de complexity. Aggregate services under standard categories will allow a better structured service search and will bring the possibility of aggregate similar services under unique domains, that means the abstraction to the user of the provider. Each service type has their own characteristics, implementation methods, and interactions, but some of them could be easily confused. Due to clarify each service category an example of use case will be given. The proposal final categories are:

- **Security service:** The security service category meant to group all secure services, an example of use case is the anonymization or the encryption of the data. It could be useful to secure data results from being stolen or to prevent tracking of the source of some data results.
- **Modelling service:** The modelling service category intends to group all the modelling services. Similar to middleware services these modules receive raw data and translate

it to a known data type or structure. The main purpose of these services is to add context like metadata or to model data to a specific standard.

- **Middleware service:** The middleware service category group services which provide data translation, it meant to be a block which converts between standards or communication formats. It should be used as an interface between two different systems which have different formatted I/O representation but the same information.
- **Connector service:** The connector service category meant to group all the services which represent connectors to other modules or to another layer of the architecture. A use case example of these services is the custom connection which physical machine type needs.
- **Data analytics service:** The data analytics service category represents the group of services which receives data, analyses it with some rules or a mathematical formula and returns or forward a result to another module. These services include big computers with big data analytics and local services as an alarm or a minor data processing unit.
- **Visualization service:** The visualization service category meant to group all services of data visualization or presentation. Dashboard, result tables, graphs and others. Mostly an end of line service which shows data in a specific way that gives the user a most valuable experience than check raw data.

3.1.3 Packaging and Deployment of Service Bricks

In order to standardize the development of the service bricks and to facilitate their deployment, a workplan is considered in this section.

The implementation of the Service Bricks involves the usage of different programming languages and managing different integration tools & frameworks. To deal with the deployment and orchestration of such diverse technologies these Service Bricks need to be encapsulated in self-contained environments benefiting from quicker deployments, scalability and closer parity between the different environments.

Containerization (also known as Operating-system-level virtualization) achieves that encapsulation, and Docker is selected as the containerization solution.

With containers, isolation is done on the kernel level without the need for a guest operating system (and thus pre-defining resources that often are not utilized). This leads to more efficient, fast and lightweight solution.

Containers are flexible since even the most complex applications can be containerized and offer great portability, they can be built locally, deploy on the cloud (or in single server) and run anywhere.

Another feature of Containers is the fact that they are stackable. This way a Service Brick can be composed by several others. For example, Data Analytics bricks that are associated with Data Collection can be bundled together with Modelling bricks (like Digital Models bricks) and offered by the same container.

A container is launched by running an image. An image is an executable package that includes everything needed to run an application - the code, a runtime, libraries, environment variables and any configuration files. The image contains the instructions for creating a container. Often, an image is based on another image, with some additional customization.

In Docker, images are defined by "DockerFiles". A DockerFile defines what goes on in the environment inside each container. This may include access to resources like networking interfaces, disk drives that are virtualized inside this environment, which are isolated from the rest of the system, map ports to the outside world, and be specific about other files that are needed to that environment. This way, the build of each service defined in this Dockerfile behaves exactly the same wherever it runs.

3.1.4 List of Service-Bricks candidates

The following service bricks candidates are the result taken from the questionnaire which had as mission aggregate all the information possible, considering the planning phase. The entire table could be found in Appendix A at the end of this document.

Table 2: Service brick by category

Category	Names
Connector	Automatic Data Collection, GEM Connector, Kafka connector, MAG CM Box Connector, MAG Specht 600 Siemens Connector, Marposs gauge Connector, Pi Web Connector, Pi Web to Zeiss CMM Connector, PROPHESY Sensor Data Collection, Prophesy-data connector, Ptc Windchill connector, Q-DAS Connector, SAP connector, SAP ERP Connector, SAP MES Connector, Sharepoint connector, Siemens MMT Connector, Zoller TMS Connector, Zoller Warehouse Connector
Visualization	Dashboard to visualize the defined KPI's, PROPHESY-AR Remote Support Technology (based on oculavis SHARE) as Training and Knowledge Sharing Platform, PROPHESY-AR Software (PHI-AR viewer)
Security	Security, Trustworthiness and Data Protection Framework
Data analytics	Data Collection & Analytics Middleware, Data Mining Techniques, Data Streaming & Analytics Infrastructure, Machine learning algorithms, Maintenance-Driven CPS Systems and Processes, Optimal Stock-level prediction, PROPHESY Data Visualization Portal, PROPHESY-ML Toolkit, Remaining Useful Life prediction, Repair Time prediction,
Modelling	Data Analysis & Fine-Tuning, Digital Models, Machines and Tools Models
Middleware	Interoperability & Data Sharing Middleware

In order to better understand the proposed service bricks, they were presented in groups in the table above. As expected, the majority of the service bricks are connectors to a variety of software and hardware components.

Based on the obtained results, the data were processed and became convenient the creation of two graphs. The distribution of the workload by work package, which can show how the development will take place during the project, and the categories distribution which presents an idea of the percentage for each service category.

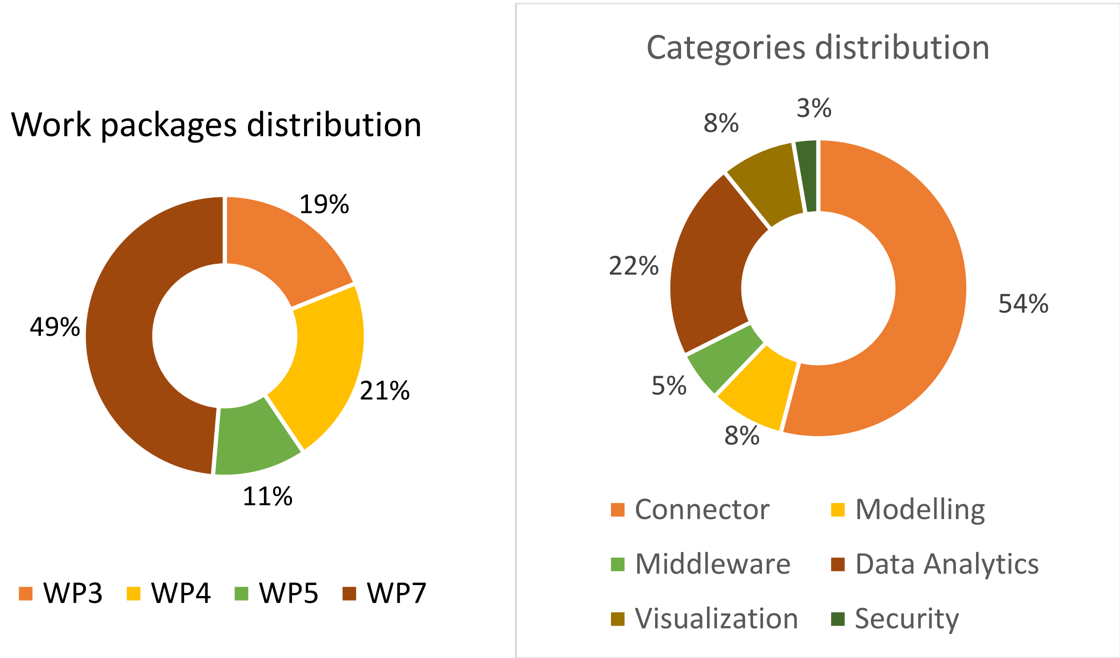


Figure 4: Service Bricks by Work Package and by Category

3.2 Turn-key Solutions Composition

3.2.1 Overview and Definition

The turn-key solutions composition will be the core module of the architecture, it will piece together all the specifications and service bricks to implement the features needed. Its main purpose is to interpret the KPI specifications, understand which services will be needed to satisfy the specifications and create all the conditions based on their characteristics to get the desired result. The ambition is to generate an autonomous module which will be moved between deployments and will take care of all the dependencies and configurations needed to achieve the specified objective.

Due to the variety of deployment options, develop a tool with static input modules would be impossible and unsearchable, so the best approach is to get a list of the available assets at the deployment site and start to build a turn-key solution from them.

Other input required by the tool would be the list of the implemented service bricks. Beyond the service names the turn-key composition tool will need to be aware of the configuration parameters of each module, more precisely their communication data formats and interaction types, as descripted in the previous section.

The last input on the tool should be the specifications of the KPIs, that include their formulas and requirements, as explained with more detail the next section.

3.2.2 Turn-key composition tool mock-up

In order to work this tool will need to receive a deployment assets list, which will describe all the machines and data sources available, a list of the available service bricks and a list of the specified KPIs. The Figure 5 presents a mock-up of the pretended look of the composition tool. In the left pane it's a list of the deployment assets which the user can use to gather the data. In the right side there are two different menus, the service bricks list which handle all the implemented services, and the KPIs specification. The last section is the middle one where the user drops all the selected modules and compose the turn-key solution.

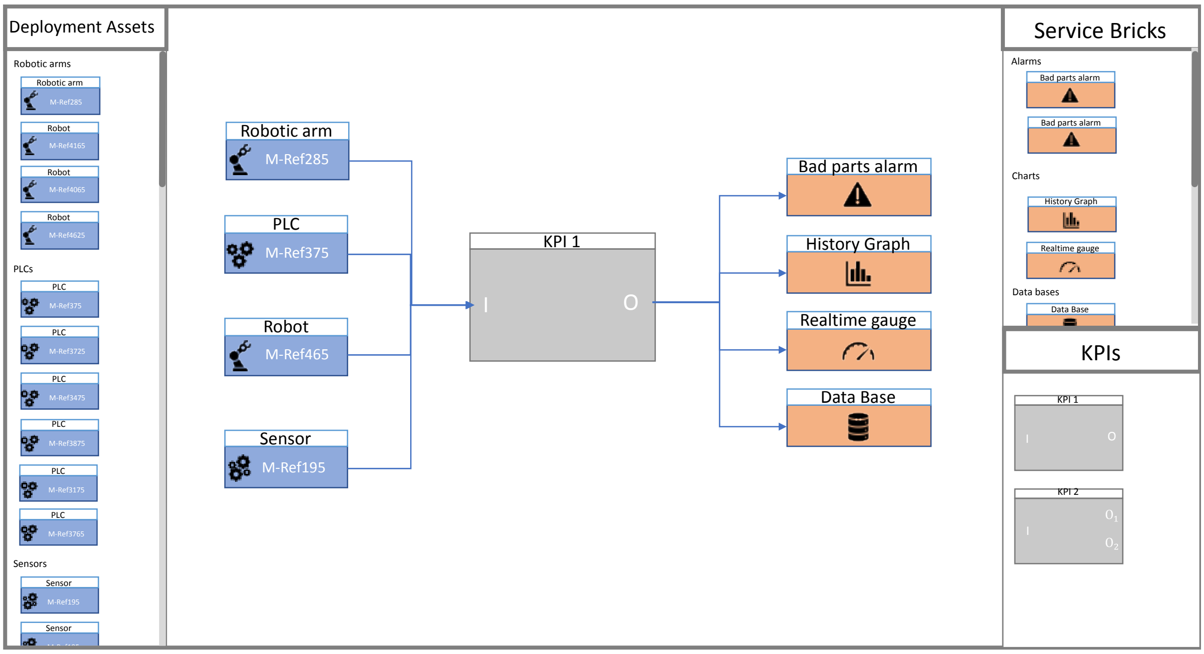


Figure 5: Turn-key composition tool mock-up

The composing of a turn-key solution is divided into various phases. The first of them will be the specification of the data sources, the KPI specification, and the end usage modules which will display or store the KPI calculation results. Figure 6 shows an example of an input with four sources of data, a KPI specification, and four end modules. This example is the same gathered from the mock-up figure. The user specifies that the KPI calculation will be based on data from two robotic arms, one PLC, and one sensor, all of them accessible in the network and with implemented connectors. It will use the KPI 1, that has a mathematical formula, which transforms the input and stream data through the output. As final modules the user choose an alarm which will monitor the bad parts count, a history graph which stores the data as a buffer and show it in different ways based on filters and axis parameters, a real-time gauge which the intention is to show an on the go KPI result visualization and a database which will store all the past output from the KPI.

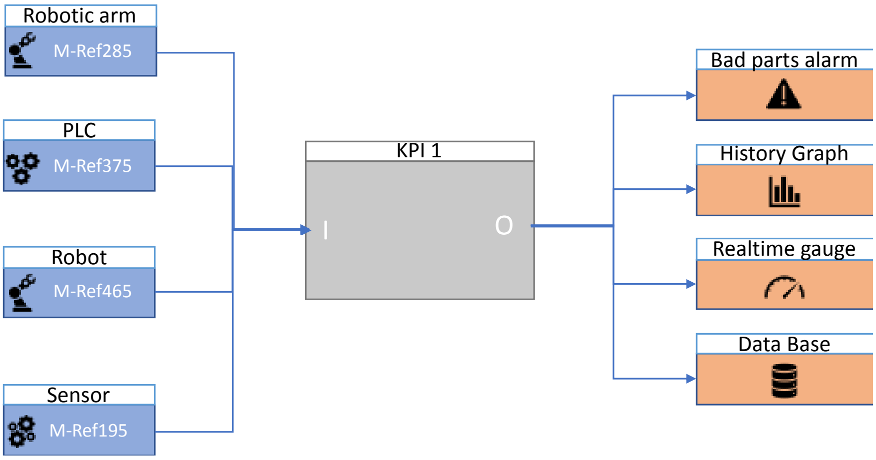


Figure 6: Turn-key solution composition input example

The composition tool will get each component of this specified input and will build a turn-key solution. The second phase is the optimization of the workflow and it involves a lot of parameters, the optimization engine will understand which connectors are needed to the specified machines, what are their interaction mechanisms and communication standards.

Figure 7 shows an overview of what should be the workflow after the composition tool takes care of all the dependencies. It's noticeable the usage of different modules depending on the connections. The letters are there to abstract the data structures supported by each module. To implement the desired KPI the composition tool needed a module which received raw data from the sensor and modelled it to the A data structure and a module to translate the output of the PLC which was in the B standard and needs to be converted to integrate with the data analytic phase. The KPI specification, in this example, is representing the scrap rate, where the sum of the bad parts is divided by the total of the parts. Considering that the standard A contains a flag which represents if the part is good or not, it will be needed tree analytics modules. One of them will count the bad parts, other counts the total of products and the final module devices both of the previous results. Between these modules, a conversion is needed because the module which divides just receives C standard data.

This is a simple example, but it was chosen because it could show the different category roles in the architecture.

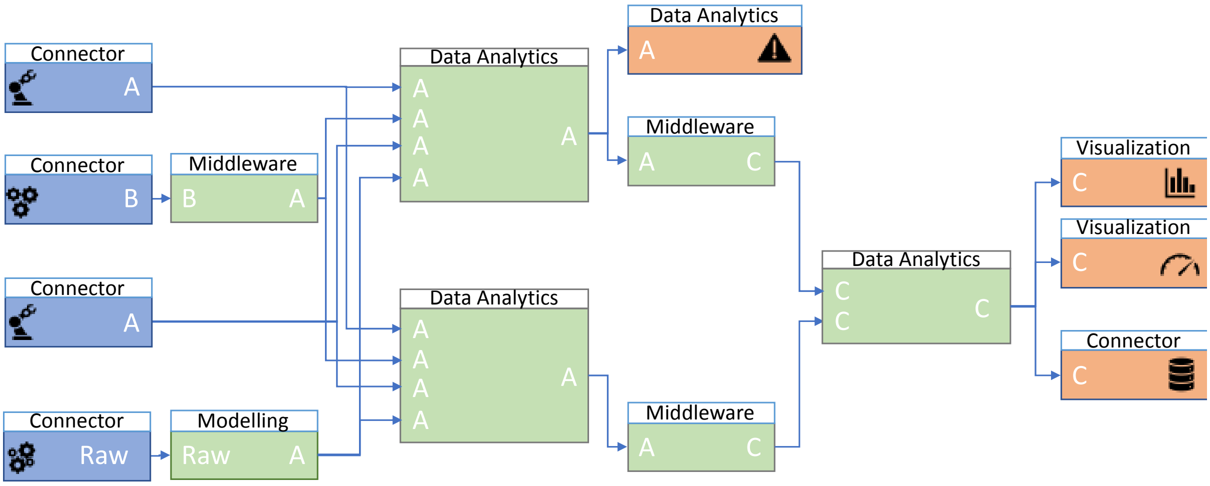


Figure 7: Turn-key solution composition output example

The last part of the image shows the end usage modules, with an orange colour, that are the modules which stores and shows data to users. In this example was chosen four modules each one with a different approach to the received data. The visualization modules and the database connection are related with the final result of the calculation, therefore are connected to the output of the last module. The alarm that should track the bad parts count is connected to the data analytic module which takes care of the bad parts count.

The composition tool will try to implement the workflow as simple as possible, it should be capable of reusing the output of the modules instead of calculating them multiple times in the same workflow.

After the calculations, the turn-key solution module is generated, and the module becomes available for future use. The implementation of the turn-key solution should be as direct as possible with the addition of the minimum number of new bricks possible.

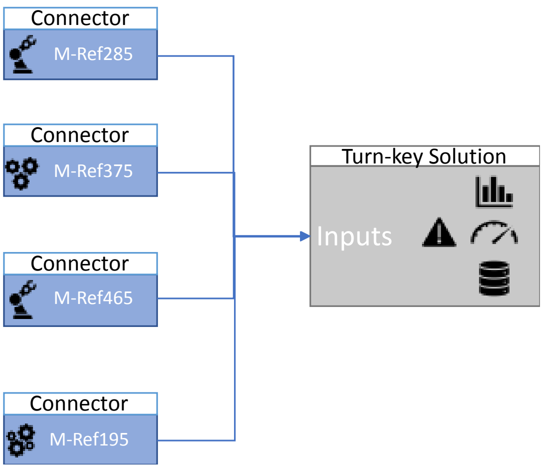


Figure 8: Using already defined Turn-key solution example

3.3 KPI Specifications

3.3.1 Context

The Key Performance Indicators are a way of measuring the effectiveness of an organization and its progress towards achieving its goals[1]. The purpose of the KPI specifications task is to define the requirements for the computation of KPI's. An example would be the Scrap Rate, already spoken in this document. Its definition is the count of the bad parts divided by the total of products that came out of the production line. With this KPI the company could understand if the process is near the objectives or not and can decide if an investment could overcome the expectations and increase the global performance of the production.

Therefore, a KPI could have two types of inputs, the raw data from the machines or indicators given by the implemented tools, which with metadata, or other form of data description, could facilitate the calculation process. The case of the scrap rate, if a machine already has the counters of the bad parts and the counter of the total of pieces, a simple divide operation would make the trick and give the result of the KPI.

3.3.2 Methodology

Useful KPI's will be designed upon basis of Normalized KPI's, I-care KPI's and other project stakeholders KPI's.

The overall methodology is the following

1. Define generic commonly used KPI's
2. Provide the means for calculating the KPI's
3. Specify the sources of data for the KPI calculations

3.3.3 Expected inputs

Data sources for the data will be coming from PROPHESY-CPS, PROPHESY-ML and PROPHESY-AR (e.g., Predicted downtimes, time to complete a remote maintenance service etc.).

3.3.4 Expected outputs

The outcome of the KPI specification should be the specified mathematical formula and what dependencies it has. Each KPI could have its own requirements and input data.

3.4 Business Models

As the name "Service Optimisation Engine" suggests, an iterative process of specification for the turn-key solutions that will be composed and deployed based on the PROPHESY-SOE engine/platform will be performed according to the task at hand. The most "optimal" solution for each client/beneficiary will then be selected, and the appropriate "business model" for exploitation by the project partners will be formulated on top. The brief business cycle as well as relevant definitions are explained below:

[1] Source : www.macmillandictionary.com

Client/Beneficiary: Any factory or production plant, or machine vendor, within the project as well as in the future (during sustainability and exploitation stages of the project assets), requiring specific cutting edge PdM solutions from the PROPHESY ecosystem to achieve specific business goals of their operation. Hence this is not necessarily restricted to the "demonstrators" of this stage (i.e PHILIPS, JAGUAR LAND ROVER and MAG) but also extended to any relevant "end-user".

Goals/Scope: Any KPI or FMECA (Failure mode, effects and criticality analysis) or RAMS quantitative target to be achieved as clarified by the "consultant role" (see below) based on the business needs of the client/beneficiary. Moreover, any related business metric concerning cost cutting, time optimisation, asset efficiency, lifetime of equipment, operational indices, critical failure part identification etc as dictated by the end-user as being essential and requiring PdM optimisation.

Solution Constituents: Specific Service Bricks (i.e Data Analytics, Data Collection and Visualisation modules properly composed) combined with the relevant accompanying services (Technical support, Business support and Training of the "customer") to formulate a solution (UNP composition) that will then be optimised by the SOE for use.

Consultant/ Value Chain Role: A classification of project partners as will be analysed in detail within Task 8.2 and Deliverables 8.7-8.9 refining exploitation strategies, in a form of "Value chain" that co-operates to offer the proposed solution through the [PROPHESY] ecosystem. In a quite loose and oversimplified version of the value chain roles for explanation reasons we could propose the following schema: An integrator (ex. INTRA) or PdM consultant (ex. ICARE) with established market presence, approaches a prospective beneficiary/client (PHI, JLR, MAG or see above for extended beneficiaries), to clarify Goals/Scope and KPIs, and use the resources of the "Service Bricks" and "Solution constituents" provided in the ecosystem by "Expert Roles" (PdM Data Acquisition Experts such as ARTIS, OCULAVIS, SENSAP and UNPARALLEL) with the consulting and know-how of IoT and PdM Analytics experts (such as TUE, AIT, NOVAID, MONTRAGON, FHD) etc.

Business Model: Any individual or joint/collaborative value chain that:

- Has precisely described business roles within partners
- Makes sure that the business requirements of the end-user can be actually realized on the basis of the functionalities of PROPHESY-SOE, PROPHESY-CPS and PROPHESY-AR components through precise composition and then optimisation by comparing alternatives
- Provides consultancy on the end-user/beneficiary to achieve business goals/scope (see definitions above)
- Formulates an optimal solution based on the service bricks and accompanying services of the ecosystem (possibly slightly augmented or optimised for each case)
- Follows up effectiveness and re-iterates optimisation, but most importantly follows up cost-benefit analysis (see below) to validate the efficiency of the proposed schema.

- Charges through a pricing scheme (pay as you go, freemium SaaS/MaaS model, FFP contract, contractual remote support or on-demand services etc) and of course clarifies ownership, licensing and IPR issues while it manages relevant operations
- In order to offer an integrated PdM solution which can be either turn-key (as presented by the PROPHESY ecosystem) or only a custom modified subset of service bricks and accompanying services, or even a “Sandbox /Test bench” for evaluation of such solutions for the future.
- It is envisioned that such successful use-cases will be also presented upon completion of the PROPHESY ecosystem as guidelines for prospective new end-users and beneficiaries.

These concepts are refined and placed in a more precise context within:

- Task 6.4 Business Modelling for PdM Services (Leader: MAG; Duration: M12 – M36)
- Task 8.2 Business and Exploitation Planning (Leader: INTRA; Duration: M1 – M36)

3.5 Cost-benefit Analysis

The cost-benefits analysis module of the PROPHESY-SOE will enable interested stakeholders to:

- Calculate various economical KPIs that will be associated with actual or planned PROPHESY deployments.
- Evaluate alternative predictive maintenance deployment components in terms of their cost-benefit outcomes.

As such the operation of the component will be based on the following inputs and outputs:

- **Inputs to Cost-Benefit Analysis Modules**: The cost-benefit analysis components will take as input a set of parameters that are associated with the costs a PROPHESY deployment (e.g., costs of software licenses, costs of developers’ resources, hardware and equipment costs, training and consulting costs, deployment costs), as well as with estimates of the anticipated benefits (e.g., maintenance savings, increased OEE, savings on inventory costs, saving on maintenance engineering costs). These inputs will be provided in either of the following two ways: (i) Based on data entry processes from maintenance and/or business experts; (ii) Calculated based on data from PROPHESY systems and databases.
- **Output of the Cost-Benefit Analysis Modules**: The output of the cost-benefit analysis tool will include: (i) Capital Budgeting Indicators (e.g., ROI (Return-on-Investment), IRR (Internal Rate of Return) and Payback Period); (ii) The estimated total (annual) savings resulting from the predictive maintenance deployment;

The outputs will be visualized and presented properly to the users of the cost-benefit analytics module, in order to facilitate their decision making.

4 Conclusions

The efforts in this deliverable were turned to the definition of what are the PROPHESY-SOE layer purposes, its concept, and its components. At first, an overview of an architectural concept for the service optimization engine was developed to split the objectives into simpler problems. After that contextualization, the parts of the concept were described as trying to achieve conceptual models. The definition of some categories and key specifications in the service bricks, the overview of the turn-key solution composing tool workflow and interactions, the KPI purpose, and definition specification, the business models overview with the definition of key concepts and the cost-benefit analysis definition and its major advantages to the users were analysed and described.

This deliverable main purpose was to define the overview of the PROPHESY service optimization engine platform which will be the focus of the work in the work package six, WP6.

Appendix A Service Bricks table

Table 3: Service Bricks

Service Brick	WP	Category	Description
Artic C-Thru Connector	WP7	Connector	Connector to integrate with Artic C-Thru
Automatic Data Collection	WP4	Connector	Framework for automated collection of data from various maintenance related data sources and systems, including equipment sensors, production systems, production quality systems, enterprise system, shop floor devices and more.
Dashboard to visualize the defined KPI's	WP5	Visualization	Dashboard diagrams to visualize live data of the use-cases on tablets, smartphones or other display devices (PROPHESY-AR dashboard module)
Data Analysis & Fine-Tuning	WP4	Modelling	Provides Statistical algorithms based on dynamic decision thresholds for to predict the maintenance status of machines and components in the factory.
Data Collection & Analytics Middleware	WP3	Middleware	Middleware infrastructure that involve edge servers and devices to support data collection and analytics in PROPHESY-CPS platform
Data Mining Techniques	WP4	Data analytics	Provides distributed machine learning techniques focus on PdM and to accurate maintenance predictions
Data Streaming & Analytics Infrastructure	WP3	Data analytics	Data streaming and analytics infrastructure that enable analytics both at the edge and at the cloud layers of the PROPHESY-CPS solutions
Digital Models	WP3	Modelling	Agnostic digital models for representing PdM related datasets
GEM Connector	WP7	Connector	Connector to integrate with GEM

Service Brick	WP	Category	Description
Interoperability & Data Sharing Middleware	WP3	Middleware	Middleware for transforming sensor data to semantically annotated formats and facilitate sharing and reuse of datasets across PdM stakeholders
Kafka connector		Connector	Connector to integrate with Apache Kafka
Machine learning algorithms	WP4	Data analytics	Implementation algorithms for Machine Learning
Machines and Tools Models	WP4	Modelling	Provides guidelines and insights to anticipating the useful life of machines and predictive data analytics.
MAG CM Box Connector	WP7	Connector	Connector to integrate with Q-DAS. Required to integrate with ARTIS C-Thru
MAG Specht 600 Siemens Connector	WP7	Connector	Connector to integrate with Siemens MAG Specht 600
Maintenance-Driven CPS Systems and Processes	WP3	Data analytics	Tools and techniques for Self-Configuration and Self-Adaption that enable a maintenance prediction to the fields
Marposs gauge Connector	WP7	Connector	Connector to integrate with Marposs Gauge
Optimal Stock-level prediction	WP4	Data analytics	Machine learning algorithm to optimal stock-level prediction
Pi Web Connector	WP7	Connector	Connector to integrate with Pi Web
Pi Web to Zeiss CMM Connector	WP7	Connector	Connector to integrate between Pi Web to Zeiss CMM
PROPHESY Sensor Data Collection	WP3	Connector	Framework for data monitoring and data sharing from various sources that facilitate the integration and interoperability
PROPHESY-AR Remote Support Technology and Training and Knowledge Sharing Platform (based on oculavis SHARE)	WP5	Visualization	Knowledge sharing platform enhanced with remote and advanced visualization capabilities including AR. Will guide maintenance employees in performing and completing maintenance processes, based on remote support from the machine vendor

Service Brick	WP	Category	Description
PROPHESY-AR Software (PHI-AR viewer)	WP5	Visualization	Enables the integration of advanced visualization techniques based on augmented reality in the loop of PdM solutions as a necessary condition for reducing the maintenance costs. Will be created for Philips in PROPHESY specifically.
Prophesy-data connector	WP7	Connector	Connector for Prophesy-data to be used in Philips environment/dashboards (e.g. QlikSense)
PROPHESY-ML Toolkit	WP4	Data analytics	Unified and reusable framework that integrate IoT data collection and selected data analytics schemes for Machine Learning, Data Mining and Statistical techniques for Adaptive Self-Configuring PdM programs
Ptc Windchill connector	WP7	Connector	Connector to integrate with PTC Windchill
Q-DAS Connector	WP7	Connector	Connector to integrate with Q-DAS
Remaining Useful Life prediction algorithm	WP7	Data analytics	Machine Learning algorithm for remaining useful life prediction
Repair Time prediction algorithm	WP4	Data analytics	Machine learning algorithm to predict repair time prediction
SAP connector	WP7	Connector	Connector to integrate with SAP
SAP ERP Connector	WP7	Connector	Connector to integrate with SAP ERP CPS Manufacturing
SAP MES Connector	WP7	Connector	Connector to integrate with SAP MÊS
Security, Trustworthiness and Data Protection Framework	WP3	Security	A robust framework that implement mechanisms to provide security the of PROPHESY-CPS platform at the network, device and cloud levels.
SharePoint connector	WP7	Connector	Connector to integrate with SharePoint
Siemens MMT Connector	WP7	Connector	Connector to integrate with Siemens MMT
Zoller TMS Connector	WP7	Connector	Connector to integrate with Zoller TMS
Zoller Warehouse Connector	WP7	Connector	Connector to integrate with Zoller Warehouse