capri

Cognitive Automation Platform
for European PRocess Industry
digital transformation

Deliverable

D4.1 Sensors Layer Reference Implementations

Deliverable Lead: BFI

Deliverable due date: 31/07/2022

Actual submission date: 29/07/2022

Version: 2.0

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 870062

Document Control Page	
Title	Sensors Layer Reference Implementations
Lead Beneficiary	BFI
Description	This deliverable of type OTHERS presents the implementation results of the Sensor Layer Reference Architecture for Cognitive Plants in the process industry, as developed in the CAPRI project. The implementation is based on Open-Source environments from bodies such as the APACHE Software Foundation or the FIWARE foundation and aims to be easily reproducible by third parties. As such, the platform reference implementation has been published as Open-Source software as well. In addition, we provide guidance on how to realize innovative smart sensor applications on top of the platform, by releasing explanatory videos, accompanying text documents, datasets and partly even the applications itself that have been implemented in the CAPRI project.
Contributors	Christoph Nölle, Norbert Holzknecht (BFI), Mattia Giuseppe Marzano, Gabriele De Luca, Antonio Salis (ENG), Selma Celikovic, Jakob Rehrl (RCPE), Ana Pinto (AIMEN), Cristina Vega, Anibal Reñones (CARTIF)
Creation date	03/05/2022
Type	Other
Language	English
Audience	☒ Public ☐ Confidential
Review status	☐ Draft ☒ WP leader accepted ☒ Coordinator accepted
Action requested	☐ to be revised by Partners ☐ for approval by the WP leader ☐ for approval by the Project Coordinator ☒ for acknowledgement by Partners

Document History

Version	Date	Author(s)/ Reviewer(s)	Status
0.1	03/05/2022	Christoph Nölle (BFI)	ToC
	18/05/2022	Comments by Antonio Salis (ENG)	ToC
	30/06/2022	Mattia Giuseppe Marzano, Gabriele De Luca, Antonio Salis (ENG)	Added chapter 3, 6.3.1 and 6.3.2

	04/07/2022	Selma Celikovic, Jakob Rehrl (RCPE)	Chapters 4.3 and 5.3; added contents to 6.3.1 and 6.3.2
	11/07/2022	Christoph Nölle (BFI)	Added introduction, conclusion, and sections 4.2 and 5.2
	12/07/2022	Ana Pinto (AIMEN)	Added 6.1.1.
	15/07/2022	Cristina Vega, Anibal Reñones (CARTIF)	Added 4.1, 5.1, 6.1.2
	15/07/2022	Mattia Giuseppe Marzano, Gabriele De Luca, Antonio Salis (ENG)	Updates on chapter 1, 6.3.1, 6.3.2
	18/07/2022	Anibal Reñones (CAR)	Added chapter 2
	19/07/2022	Jakob Rehrl (RCPE)	Add tables in chapter 6
1.0	20/07/2022	Ana Pinto (AIMEN)	Review of the whole document
1.1	21/07/2022	Antonio Salis (ENG)	Added Zenodo links for CAS1, CPS1, CPS2
1.2	23/07/2022	Asier Arteaga (SID)	Added 6.2
1.3	26/07/2022	Cristina Vega, Anibal Reñones (CAR)	Review contents, other files
2.0	29/07/2022	Cristina Vega (CAR)	Final Review

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 870062

Table of Contents

Table of Figures

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 870062

List of Tables

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

DISCLAIMER

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

EXECUTIVE SUMMARY / ABSTRACT SCOPE

This deliverable of type OTHERS presents the implementation results of the Sensor Layer Reference Architecture for Cognitive Plants in the process industry, as developed in the CAPRI project. The implementation is based on Open-Source environments from bodies such as the APACHE Software Foundation or the FIWARE foundation and aims to be easily reproducible by third parties. As such, the platform reference implementation has been published as Open-Source software as well. In addition, we provide guidance on how to realize innovative smart sensor applications on top of the platform, by releasing explanatory videos, accompanying text documents, datasets and partly even the applications itself that have been implemented in the CAPRI project.

The deliverable is part of a series of documents describing the different layers of the CAPRI Cognitive Application Platform (CAP). The other documents are:

- D4.3: Control Layer Reference Implementations
- D4.5: Operation Layer Reference Implementations
- D4.6: Planning Layer Reference Implementations

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 870062

1 Introduction

This deliverable describes the sensor layer reference architecture of the Cognitive Automation Platform (CAP) of the CAPRI project. The CAP aims to be a generic software architecture for process industries, through the usage of Apache and FIWARE open-source technologies, which facilitate the development of cognitive applications. The sensor layer is at the bottom of the CAP, providing the connectivity to sensors and other field devices. Three concrete implementations of sensor layer are reported, along with sample applications, for the three different domains considered in CAPRI, i.e., asphalt, steel, and pharma.

1.1 Audience

Target audience of this deliverable are software architects in the process industries, who want to implement and commission cognitive software applications, such as:

- Changes detected in the production process that require the plant to respond to these dynamic fluctuations by adapting the production to stay within the targets of costs and rate, as well as quality and sustainability thresholds.
- High-level business intelligence for the whole plant, focusing on the economic peculiarities.
- Decision support tools to reroute or cancel processing of those faulty produced materials or products or reallocating these products to alternative orders.

1.2 Relationship with other deliverables

The reference architecture of the higher layers of the CAP will be reported in the deliverables D4.3, D4.5, and D4.6. Whereas the present document focuses on the integration of cognitive solutions into the CAP, the solutions themselves were presented in deliverable D3.2.

1.3 Document Structure

The document is structured in 7 chapters. The first one is the introduction chapter where the contents of this deliverable are introduced. Chapter 2 provides an overview of the attached files that form the core of the deliverable. After that the technical part starts, divided in 4 chapters, ending with chapter 7 focused on the conclusions. Chapter 3 describes the general architecture of the sensor layer, with concrete implementations for the three domains detailed in Chapter 4. In the following Chapter 5, the three data modeling approaches are explained. Chapter 6 describes how the cognitive solutions developed in CAPRI have been integrated into the CAP, and gives a short introduction to the accompanying material in the form of screencasts, algorithms, data, etc. Finally, Chapter 0 provides conclusions and next steps.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

2 Attachment Structure Description

Since Deliverable D4.1 is of type OTHER, each Cognitive Solution (CS) integrated within the CAP platform is equipped with:

- A number of attachments of different nature (video, data, metadata, application, code, …), containing additional information that helps to better understand the final output of the different CSs and how they are being integrated within the CAP platform, providing concrete evidence of what has been implemented in WP4.
- A textual part, available in next chapters, that complements the "physical" content in attachment, explaining what it is and how to exploit it.

The attachments are available in their corresponding Folder in Zenodo repository. Due to limitations of space (52Mb) in EC portal not all the assets could be included into a single file. That is the reason we have decided to include all files into CAPRI's Zenodo account and CAPRI YouTube channel video, for the videos showing specific demonstrations.

Table 1 below lists all the links (Zenodo and, if applicable, YouTube) of the different files described in the present report.

Table 1: D4.1 – List of attachments

Cognitive Solution	Content	Type	Location
CAS1 - Cognitive bitumen sensor	CAS1_PPT1	PPT explaining CAS1 integration with CAP	https://doi.org/10.5281/zenodo.6866498
	CAS1_data1	data sent	
	CAS1_script1	python script for communications	
CAS2 - Cognitive sensor for amount of filler	CAS2_dataset_5.RData	Data set used in the CAS2 model	https://doi.org/10.5281/zenodo.6922992
	CAS2_DMP_dataset5	DMP of the data set used in the CAS2 model	
	CAS2_sourcecode_1.R	R code for processing the data set used in the CAS2 model	
	CAS2_sourcecode_2.sql	SQL code for processing the data with the CAS2 model	
	CAS2_video_5	Video of the Model implementation at IoT local system un in asphalt pilot plant	
	CAS2_video_6	Video of MQTT dataframes received from local IoT at CAP platform	
	CAS2_video_7	Video of Population of CAS2 database in the CAP platform	
CSS1 - Cognitive steel tracking sensor	CSS1_video_2	Video about the cognitive steel tracking sensor	https://zenodo.org/record/6899646

CPS1 - Cognitive sensor for blend uniformity	CPS1_App_and_Data_2.zip	Python implementation of CPS1 and sample data	https://doi.org/10.5281/zenodo.6875774
CPS2 - Cognitive sensor for granule quality	CPS2_App_2.zip	Python implementation CPS2 and sample data	https://doi.org/10.5281/zenodo.6875981
	CPS2_Data_2.zip	Sample data sets of LOLIMOT model	

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

3 Sensors Layer Architecture

The Reference Architecture of CAP platform (Figure 1) is designed to support the development of advanced cognitive software solutions. It is considered a digital enabler toward the innovation of process industry. In addition, it is an Open-Source solution having a wide range of applicability, supporting at the same time a large variety of applications. The challenge becomes ever harder inthe real industrial environment, for this reason the design was done in an iterative process started in D2.1, where as a first step there was a phase of functional and non-functional requirements collection followed by a continuous validation from the pilots with the common goal of introducing the cognitive automation processes in the process industry. In D3.1 the improvements done in the Reference Architecture are reported, highlighting the concept of edge and cloud cognitive computing with the aim of solving business challenges, creating new value from data and improving the product quality.

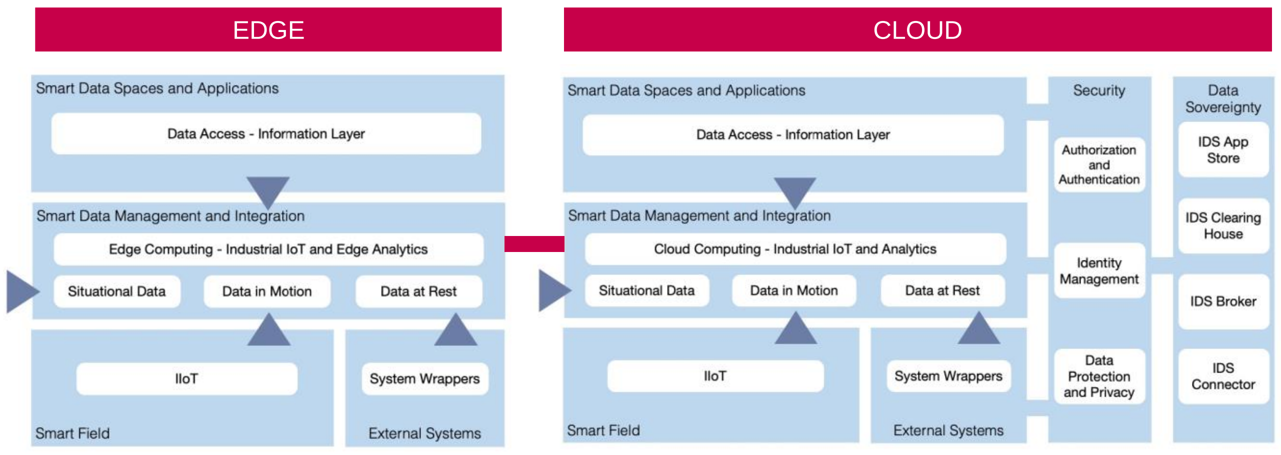

EDGE
CLOUD
Smart Data Spaces and Applications
Data Access - Information Layer
Smart Data Management and Integration
Edge Computing - Industrial IoT and Edge Analytics
Cloud Computing - Industrial IoT and Analytics
Situational Data
Data in Motion
Data at Rest
IIoT
System Wrappers
Smart Field
External Systems
Security
Authorization and Authentication
Identity Management
Data Protection and Privacy
Data Sovereignty
IDS App Store
IDS Clearing House
IDS Broker
IDS Connector

Figure 1: CAP Reference Architecture

The CAP Reference Architecture is structured in many horizontal layers able to guarantee the interoperability, privacy, protection and data sovereignty. In this section, the focus will be on the sensor layer, represented by the physical layer (as described in D3.6) where the industrial devices, machines, actuator, sensors, wearable devices, and robots are located. For their integration into the platform the most common industrial IoT protocols are used, like OPC UA, MQTT, etc. Standard interfaces and protocols must be used in order to represent the information collected from the plant, as well as to connect and integrate actuators to implement the sensing and control mechanisms. Typically, a sensor layer is based on the existing proprietary components, and a common scenario aims to integrate real-time data originated from the plant, so that an important interconnection of the solutions assumes high importance.

The next sections provide the description, for each domain, of how the sensor layer is actually implemented to satisfy the requirements of the cognitive solutions integrated in the CAP Platform.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

4 Sensors Layer Platform Reference Implementations

4.1 Asphalt domain

CAS1 - Cognitive Sensor of Bitumen content in recycled asphalt

This cognitive sensor's purpose is to provide the real bitumen content present in Reclaimed Asphalt Pavement (RAP). The development of this sensor, based on optical components and intelligent algorithms, will help to reduce the amount of virgin bitumen used in the asphalt mix, and enlarge the usage of RAP in the final asphalt mix, making it a more efficient and environmentally friendly process.

CAS2 – Cognitive Sensor of Amount of Filler

This cognitive sensor is developed to estimate and measure the fine filler quantity that goes out of the aggregates drying process and exits both to the baghouse filter and to the hot bins. The high-level outcome of this cognitive sensor is to obtain the real amount of filler flow present in the mix of aggregates, which will allow to waste less energy in the rotary drying drum and in the filtering (baghouse) process.

These two cognitive sensors are the core of the Sensor Layer of the Reference Architecture in the Asphalt Use case of the CAP system of the project. Hence, they are integrated in the overall CAP platform of the project, and they provide reference implementations that can serve as templates for other cognitive solutions. As Cognitive Sensors Solutions, CAS1 and CAS2 detailed description can be found in the following deliverables: D2.2 Use Case Requirements (confidential) and D3.2 CAPRI Industrial IoT Platform and Data Space (public).

CAS1 and CAS2 output results have been integrated in the corresponding platform developed for the Asphalt domain, as can be seen in Figure 2.

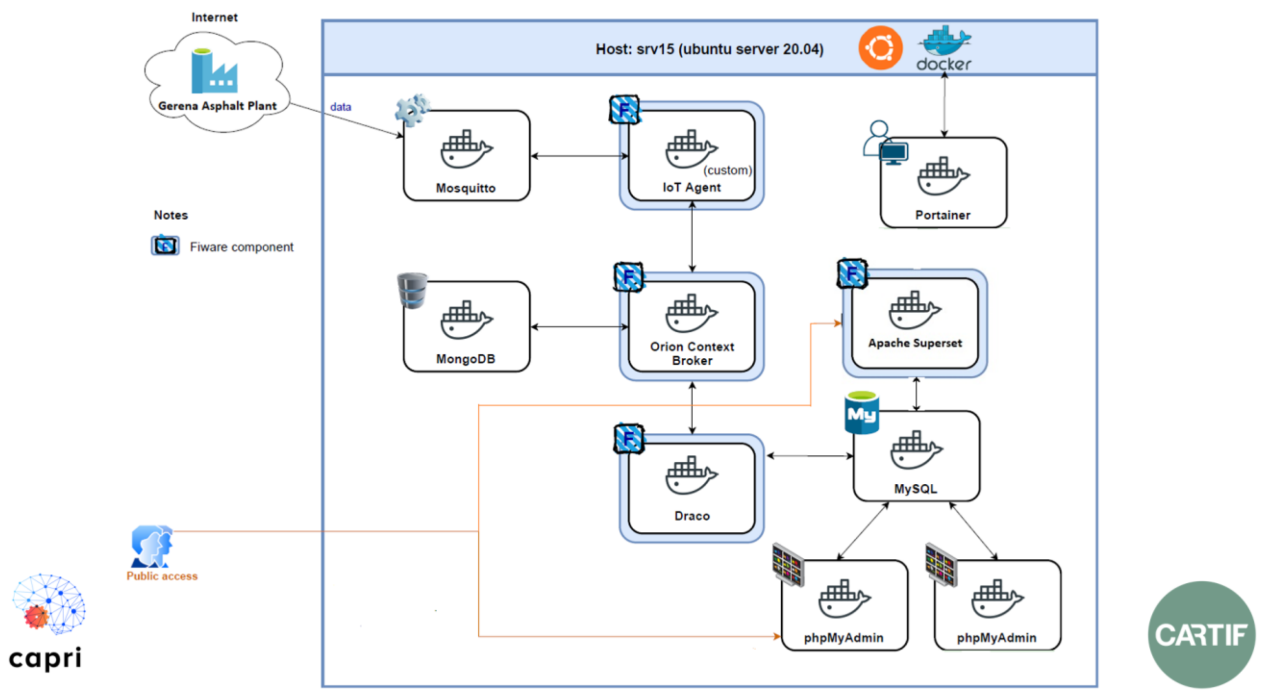


Figure 2: Basic CAP FIWARE-based platform implemented at Asphalt Use Case scenario

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

Based on **FIWARE** Reference Architecture, as described in D3.1, the asphalt domain platform has been implemented in a Linux **Ubuntu** distribution-based **server** where the different modules communicate and interact among each other, being deployed using the **Docker** platform. A docker is a container, a standard unit of software that packages up code and all its dependencies so that the application runs quickly and reliably from one computing environment to another. Each and every of the different modules have been implemented through a Docker that interacts with the corresponding others.

From the Gerena Asphalt Plant, real time data (with sampling times as low as 1 or 5 seconds, depending on the data source) is received from a WAGO PLC datalogger using **MQTT** protocol. In the asphalt domain CAP platform, a **mosquitto**-based broker receives those data, which is redirected through an **IoT Agent for JSON,** a bridge between HTTP/MQTT messaging (with a JSON payload) and NGSI. This IoT Agent has been customized to meet the asphalt domain requirements.

Also, as can be seen in Figure 3, **CAS1** and **CAS2** solutions are sending their corresponding data from their equipment, using their corresponding dedicated IP address and the MQTT protocol, being received by the same **mosquitto**-based broker and then their output data is redirected, as the rest of the production data coming from the plant.

This IoT Agent communicates and sends the corresponding data to the **Orion Context Broker** module, a Generic Enabler that provides the FIWARE NGSI v2 API, a simple yet powerful Restful API, enabling to perform updates, queries, or subscribe to changes on context information. This Broker is the core of the whole FIWARE-based Reference Architecture implemented in the Asphalt domain.

From this broker, a **Draco** module has been set up, which is a Generic Enabler that is an alternative data persistence mechanism for managing the history of context. It is based on Apache NiFi and it is a dataflow system based on the concepts of flow-based programming. In this case, it manages the data coming through the Orion Context Broker and sends it to a **MySQL** database which is used for data persistence within the CAP platform.

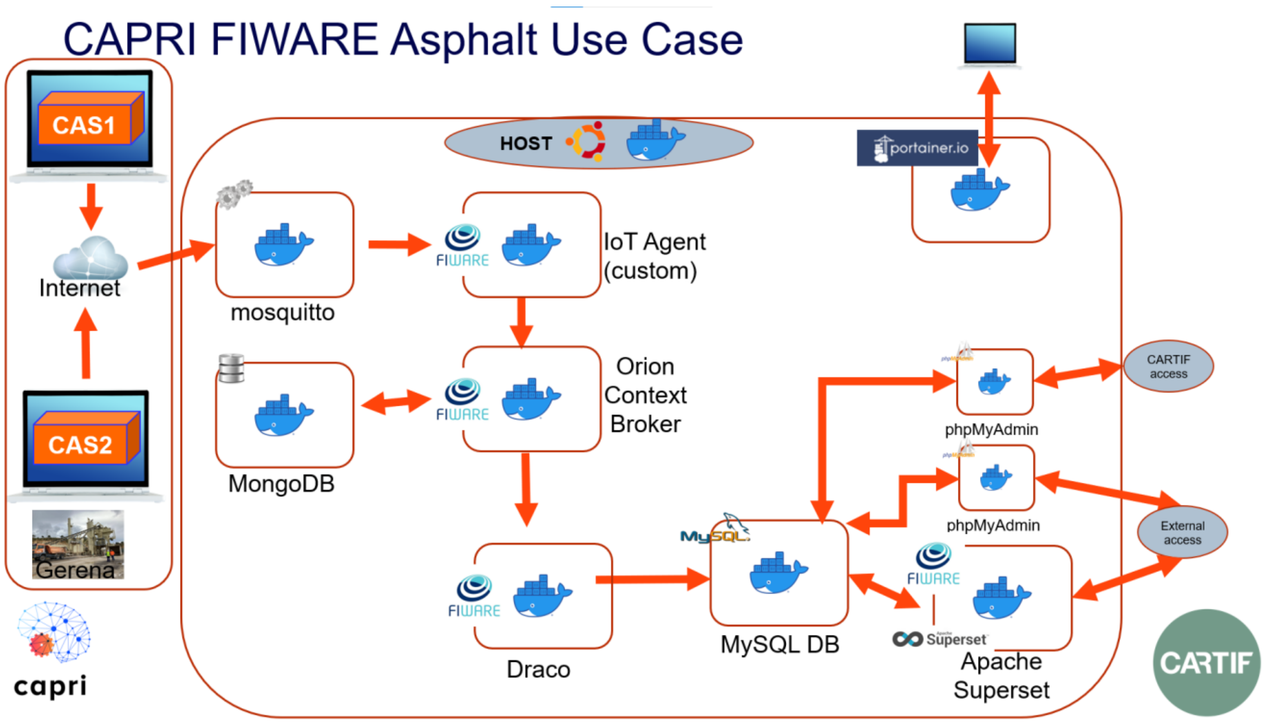


Figure 3: CAPRI FIWARE Asphalt domain schematics and CAS1 and CAS2 CSs integration

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

CAS1 and CAS2 cognitive solutions' outputs are stored within this MySQL database, where almost real time data and historical data are taken into account to feed the rest of the cognitive solutions that need their data.

This output data, which is received by the Orion Context Broker, is stored at the MySQL database through the **Draco** module and is also accessed through the visualization module based on **Apache Superset**, a lightweight module that connects to the **MySQL** database (see section 5.1) to be able to display the corresponding CAS1 and CAS2 outputs where the different information coming from this Cognitive Sensors are displayed through different fields and trend graphs. This visualization platform has been chosen for being a plug-in architecture that makes it easy to build custom visualizations that drop directly into **Superset**. This module can be accessed through a web-based interface to be accessible from anywhere, and especially, from the plant where managers and operators can see the corresponding values from both sensors and take actions based on them.

Schematics of the described implementation and the different FIWARE-based Reference Architecture for the Asphalt Domain can be seen in Figure 2 and Figure 3.

4.2 Steel domain

An architectural view of the CAP in the steel use case is shown in Figure 4. It consists of three main parts:

- The agents, collecting data from the automation system and transferring it to the CAP.
- The core cognitive automation platform (CAP), providing services such as persistence and a web interface, as well as a runtime for applications that need to work with low-latency data.
- The application runtime, which may or may not be integrated into the CAP. Applications deployed here access data from the CAP via the REST-based web interface.

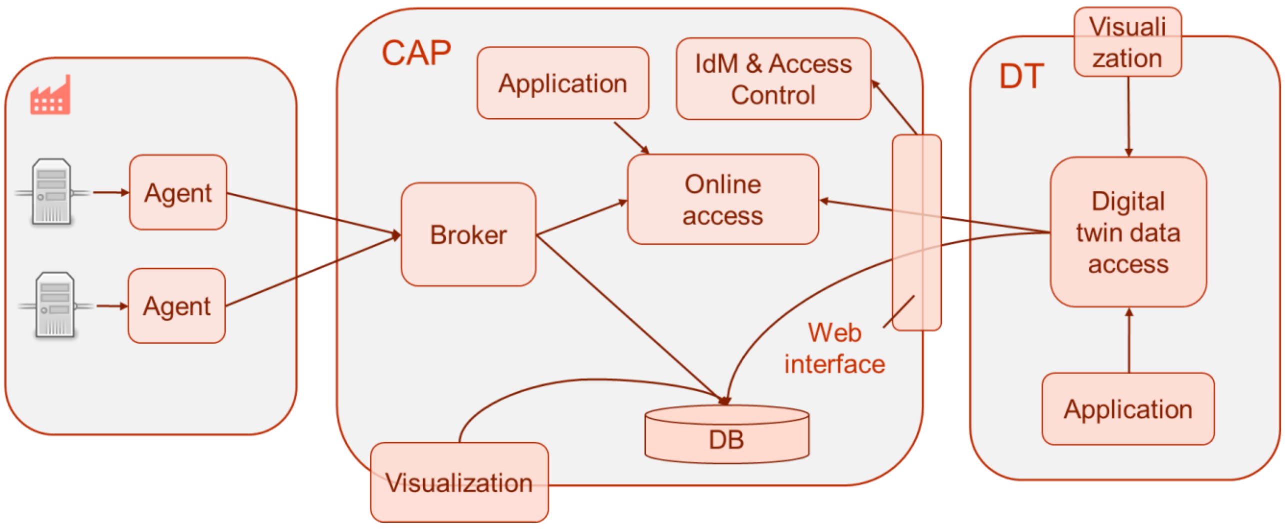


Figure 4: Architectural view of the CAP in the steel use case

The platform has been realized by means of open-source applications, see Figure 5. The data broker receiving process data from the agents is based on Node-RED (https://nodered.org/). The fast layer application runtime consists of a Kafka broker and Spark. The persistence layer has three types of databases: a timeseries database (InfluxDB), a SQL database for structured data (PostgreSQL), and a real-time database for instantaneous data access (REDIS Pub/sub).

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

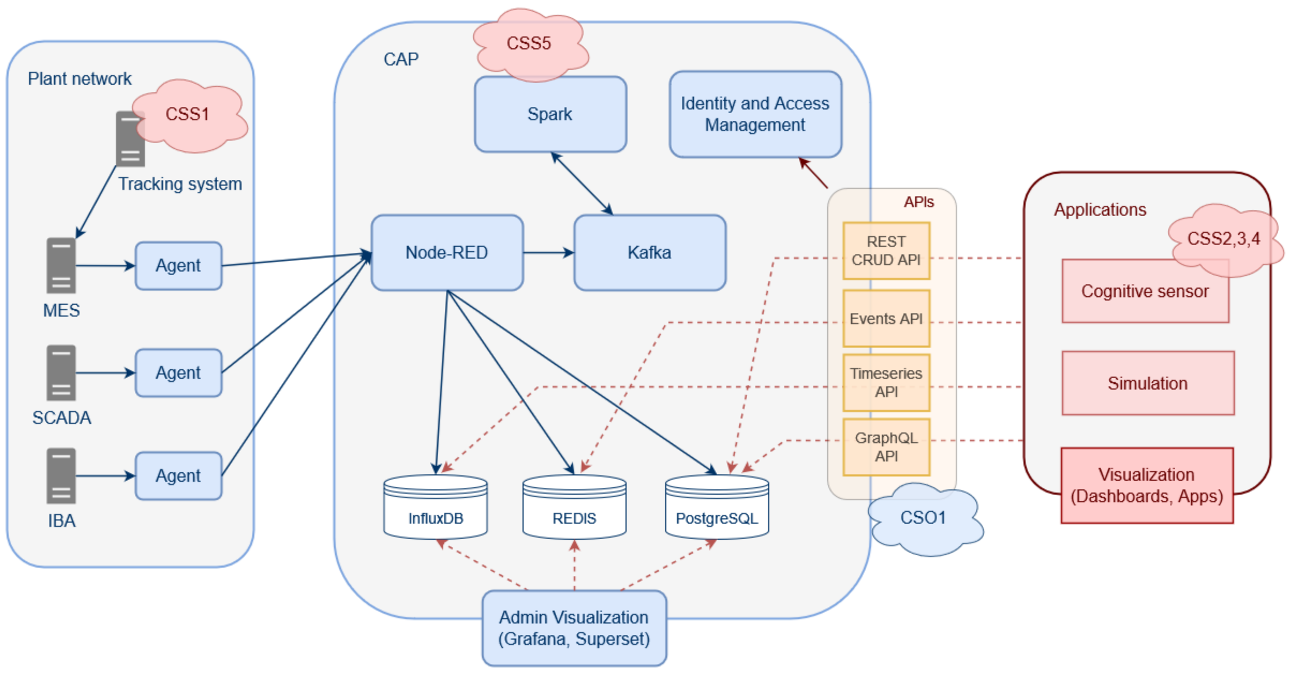


Figure 5: Components of the CAP in the steel use case

The cognitive steel tracking sensor (CSS1), the only proper sensor layer app in the steel use case, is shown located in the company network, since its software components are deployed on dedicated computers close to the hardware, evaluating the video streams of the cameras installed and trying to identify the QR codes on the steel items. Other applications consisting mainly of software only require access to the CAP data and can be deployed either as Spark applications in the CAP (fast lane apps, e.g., CSS5) or access the CAP via the API (e.g., CSS2).

4.3 Pharma domain

Figure 6 shows an overview of the CAP implementation of the sensor solutions CPS1-CPS5. CPS1 and CPS2 provide their respective raw data via OPC UA servers, which have been programmed using Python. They are executed on the respective measurement computers “Parsum & Charles Ischi Tablet Tester computer” and “Raman computer". CPS1, CPS2 and CPS4 are implemented in Python and executed in the CAP platform. They obtain the raw data that is required to perform the desired computations via OPC UA. CPS3, which is based on a dynamic model of the ConsiGma 25^{TM} dryer, is implemented in Matlab/Simulink. The data transfer between the CAP and CPS3 is done via OPC UA.

This project receives funding in the European Commission’s Horizon 2020 Research Programme under Grant Agreement Number 870062

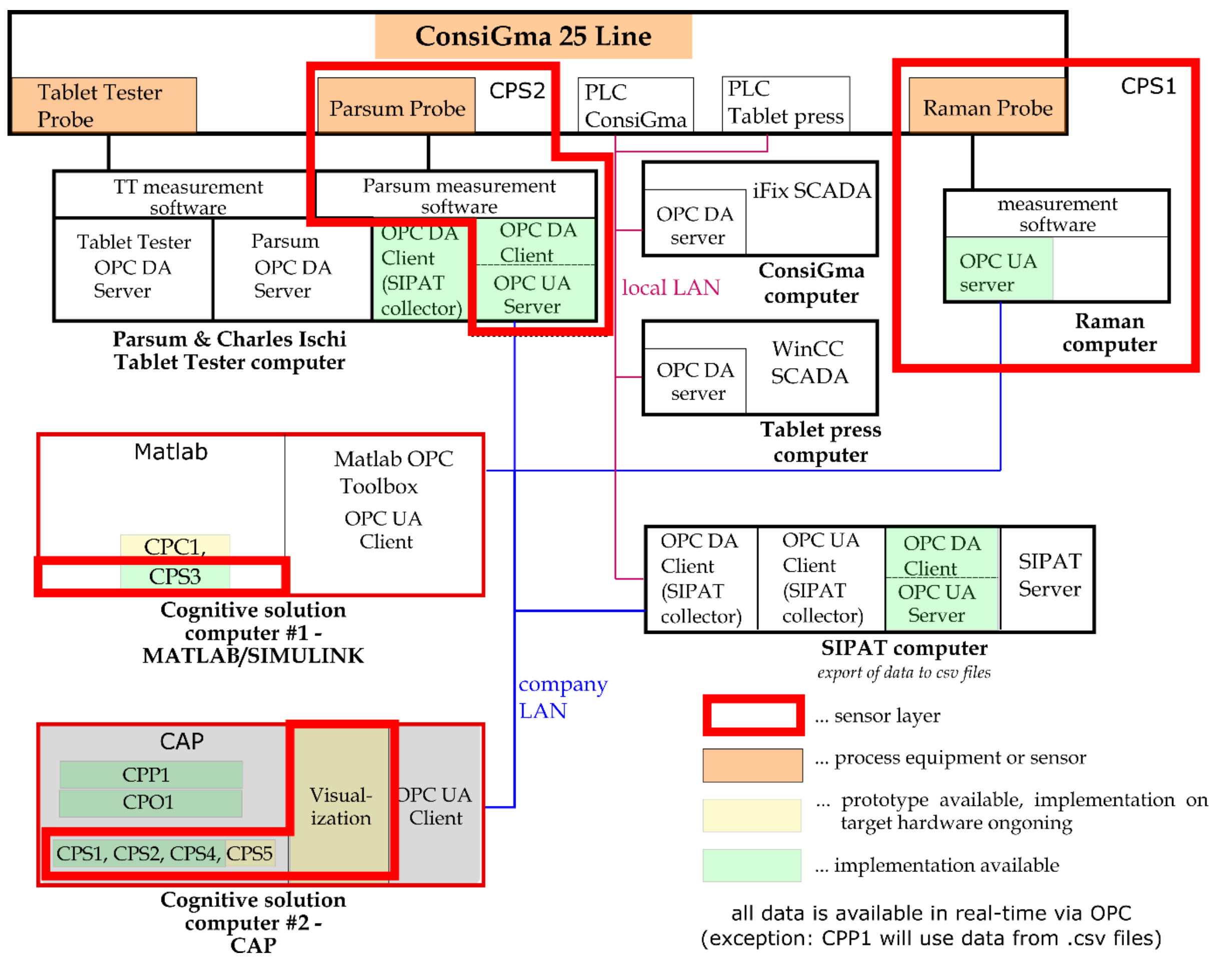

ConsiGma 25 Line
Tablet Tester Probe
Parsum Probe
CPS2
PLC ConsiGma
PLC Tablet press
Raman Probe
CPS1
TT measurement software
Parsum measurement software
Tablet Tester OPC DA Server
Parsum OPC DA Server
OPC DA Client (SIPAT collector)
OPC DA Client
OPC UA Server
Parsum & Charles Ischi Tablet Tester computer
OPC DA server
iFix SCADA
ConsiGma computer
local LAN
OPC DA server
WinCC SCADA
Tablet press computer
measurement software
OPC UA server
Raman computer
Matlab
CPC1, CPS3
Matlab OPC Toolbox
OPC UA Client
Cognitive solution computer #1 - MATLAB/SIMULINK
OPC DA Client (SIPAT collector)
OPC UA Client (SIPAT collector)
OPC DA Client
OPC UA Server
SIPAT Server
SIPAT computer
export of data to csv files
company LAN
CAP
CPP1
CPO1
CPS1, CPS2, CPS4, CPS5
Visual-ization
OPC UA Client
Cognitive solution computer #2 - CAP
... sensor layer
... process equipment or sensor
... prototype available, implementation on target hardware ongoing
... implementation available
all data is available in real-time via OPC
(exception: CPP1 will use data from .csv files)

Figure 6: Sensor layer reference implementation for the Pharma domain

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

5 Information Modelling

A core ingredient of a software platform enabling cognitive applications is the provision of semantic annotations along with the pure data. Information about the meaning of data points, including type of data and units, and relationships between entities need to be defined for applications to leverage their cognitive capabilities. This information must be available for all kinds of data, such as historical timeseries, instantaneous data, forecasts, and including single and multi-valued sensor measurements, images, videos, and others. A common base model (or top-level ontology) should define the formats of these different data types, as well as the names of recurring data fields, such as timestamps. On top of this base model, a domain-specific model reflects the characteristics of the specific use cases considered.

As an example, a characteristic feature of CAPRI's steel use case is the difficulty of tracking individual semi-products through the processing chain. The tracking sensor CSS1 aims to provide this information by means of marking the items with a QR code and using cameras to re-identify them later, but due to the harsh conditions in the steel mill errors cannot be completely avoided. The domain-specific model for the steel use case must hence be able to accommodate these errors, and applications should be able to filter for successfully tracked items only.

Applications may also need to integrate data from external sources. The semantic web is one approach for interlinking data between different repositories with different data models. It is based on technologies such as RDF, RDFS, OWL, SPARQL, and JSON-LD.

Beyond pure data models, ontologies provide a means to enable automated reasoning tools to check a model for consistency and deduce additional information from the available one, through rules encoded in the model. One drawback of ontologies, though, is the obscure modelling languages and tools that are typically used in this context, which means steeper learning curves for users such as application developers. Furthermore, to obtain interesting results from automated reasoning, a model usually needs to be fine-tuned for the desired applications. In the three CAPRI scenarios we do not make use of ontologies and automated reasoning, but this might be an interesting extension to consider in the future.

5.1 Asphalt domain

In the Asphalt domain, the whole dataset coming from the asphalt plant located at Gerena in Sevilla (Spain) is received using MQTT protocol. Two different types of data streams are sent from the plant:

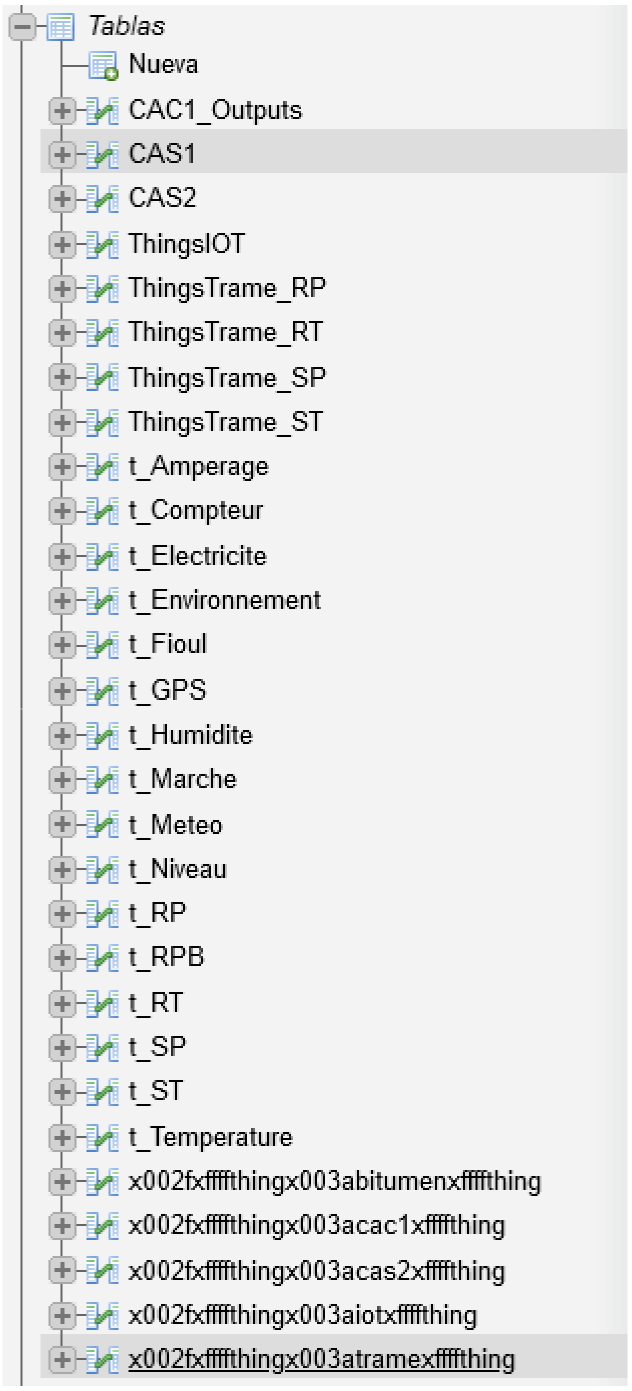
- Production dataframes: These datastreams are received according to the following conventions:
    - STOP DRYING PROCESS PATTERNS (Short RPA + RPD + RPE): The codes short RPA + RPD+ RPE are concatenated to form a single frame and are sent every 5 seconds, when the drying process is stopped (no formula and batch during the drying process).
    - DRYING SETPOINT FRAMES (SP): The SPA+SPB+SPC codes are concatenated into a single frame and sent once when the formula for drying is selected.
    - DRYING FRAME (RP): The codes RPA+RPB+RPC+RPD+RPD+RPE are concatenated to form a single frame and are sent every 5 seconds when the plant is in production (current formula and batch, ...).
    - MIX TOWER SETPOINT FRAMES (ST): The STA+STB codes are concatenated to form a single frame and are sent once when the formula is selected in the tower.
    - MIX TOWER PRODUCTION FRAMES (RT): The RTA+RTB+RTF codes are concatenated to form a single frame and are sent at the end of the mixer emptying, when the plant is in wrapper or mixer blending production (formula and batch already selected at the hot tower level).

- IoT dataframes: These datasets are sent every 1 second and they send data from all those sensors and equipment which have been directly connected to the asphalt WAGO PLC, responsible for sending all data coming from the plant. This comprises the different data coming from different temperature, flow, weather station, etc coming from the plant. More information can be found at deliverable D2.2 and the different D3.x deliverables.

In addition, all output data coming from **CAS1** and **CAS2** solutions are sent using the same MQTT protocol and received at the same point of the CAP platform.

As explained in chapter 4.1, the data, through the CAP platform based on FIWARE modules, is stored in a MySQL database with the structure that can be seen on Figure 7. The different data coming from IoT dataframes is stored in different tables with self-explanatory names (temperature, amperage, etc). All production related dataframes have been stored in the different tables depending on the data as explained in the previous paragraphs (RP, RT, SP, ST, etc) (see Figure 7).

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 870062



Tablas
Nueva
CAC1_Outputs
CAS1
CAS2
ThingsIOT
ThingsTrame_RP
ThingsTrame_RT
ThingsTrame_SP
ThingsTrame_ST
t_Amperage
t_Compteur
t_Electricite
t_Environnement
t_Fioul
t_GPS
t_Humidite
t_Marche
t_Meteo
t_Niveau
t_RP
t_RPB
t_RT
t_SP
t_ST
t_Temperature
x002fxffffthingx003abitumenxffffthing
x002fxffffthingx003acac1xffffthing
x002fxffffthingx003acas2xffffthing
x002fxffffthingx003aiotxffffthing
x002fxffffthingx003atramexffffthing

Figure 7: MySQL database structure (tables) for Asphalt Use Case data persistence

5.1.1 Cognitive sensor of bitumen content in recycled asphalt [CAS1]

CAS1 sensor aims to determine the bitumen content in RAP. This sensor output will be the % of bitumen content. No information regarding the internal information modeling can be provided since this is confidential information.

5.1.2 Cognitive sensor for amount of filler [CAS2]

CAS2 sensor aims to provide an estimation of flow of filler during the drying process of the aggregates. As such, the raw measurements need to be adjusted to compensate the undesired noise when the aspiration takes place as there is no aggregates in the drum to be dried. To compensate for this noise, a model based on the actual aspiration pressure has been created. The model obtained is explained below, together with the OTHER assets shared.

CAS2_dataset_5.RData: dataset of raw data and aspiration pressure.

The dataset shown in Figure 8 has been used for the creation of the model. The model tries to estimate the vibration measured (ACEL1_20-25 measured in g_{RMS}) based on the aspiration pressure (variable named as RPA2100 measured in mmH_2O).

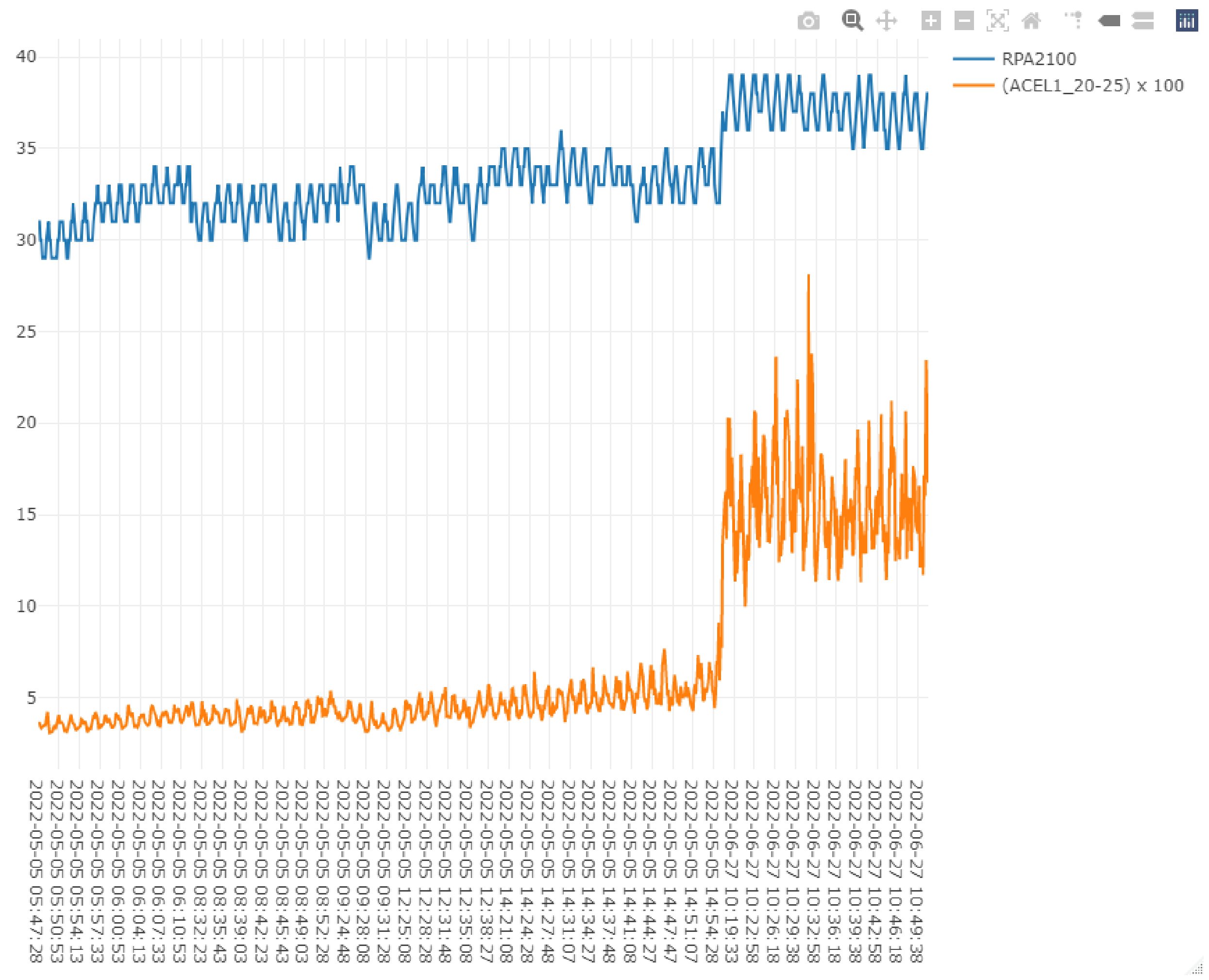


Figure 8: Dataset used for the creation of the model

This is a collection of temporary moments in time when the baghouse is running, but there is no material flow, so it is vacuuming. Figure 9 shows an operation of the baghouse during one day of production of the asphalt plant and how the different segments used for the creation of the model were selected (red rectangles).

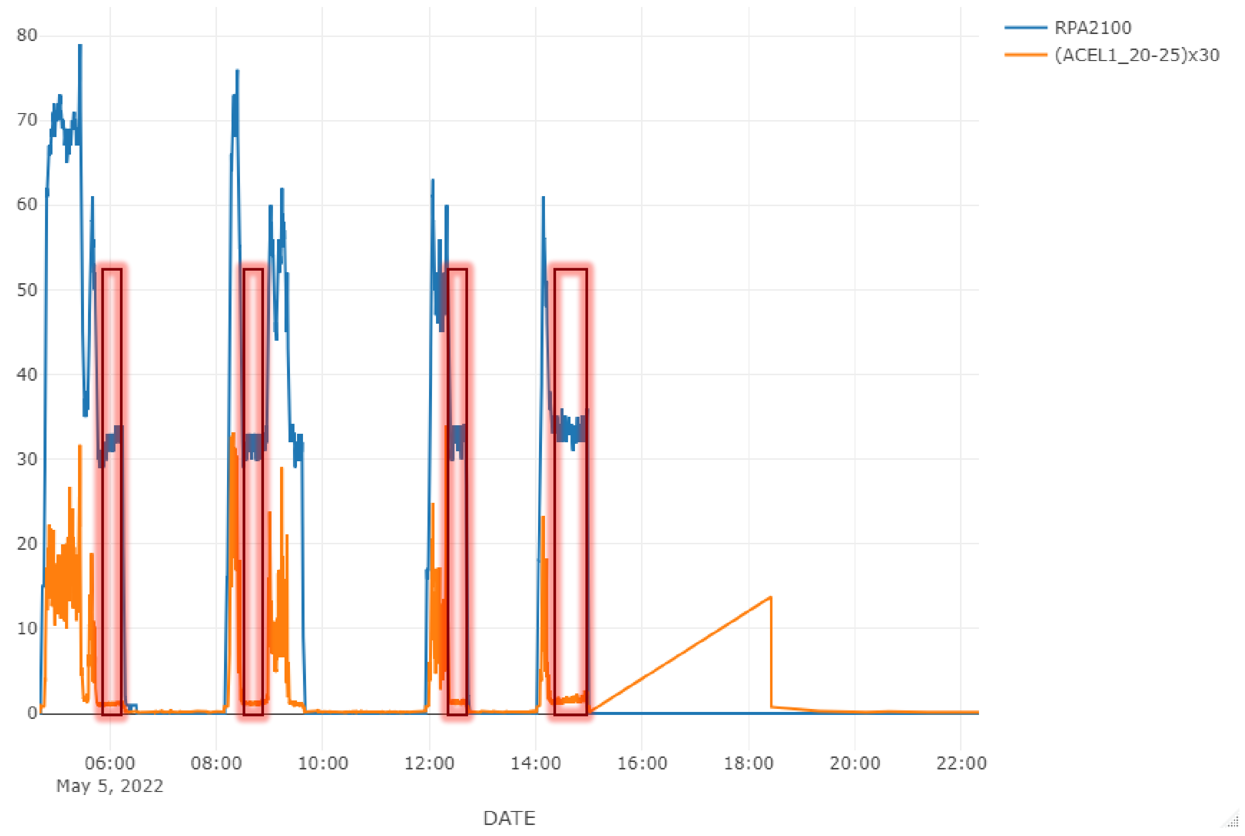


Figure 9: Filtering of data of interest

CAS2_DMP_dataset5: Data Management Plant of the CAS2_dataset_5.RDataCAS2.

This file includes the Data Management Plan associated to this data used.

CAS2_sourcecode_1.R: program used to estimate, graph and evaluate the model.

This file is an algorithm programmed in an open source R programming environment and language. This algorithm creates a model that relates the suction (x variable) and vibration variables (f function) in the suction process with the dataset described above. The developed model creates a piecewise linear relationship between the two variables for aspiration values as can be seen in Figure 10. It must be also noted that for a certain pressure below a threshold (27 mm H_2O) the output of the model is 0 as the baghouse does not operate.

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{cases} 0 & x < 27 \\ 0.00355 * \mathbf{x} - 0.07125 & 27 < x \leq 36 \\ 0.0119 * \mathbf{x} - 0.2872 & x > 36 \end{cases}$$

In fact, the model is divided in three intervals as the changes of the variable to be modelled is not continuous but sudden as shown in Figure 10.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

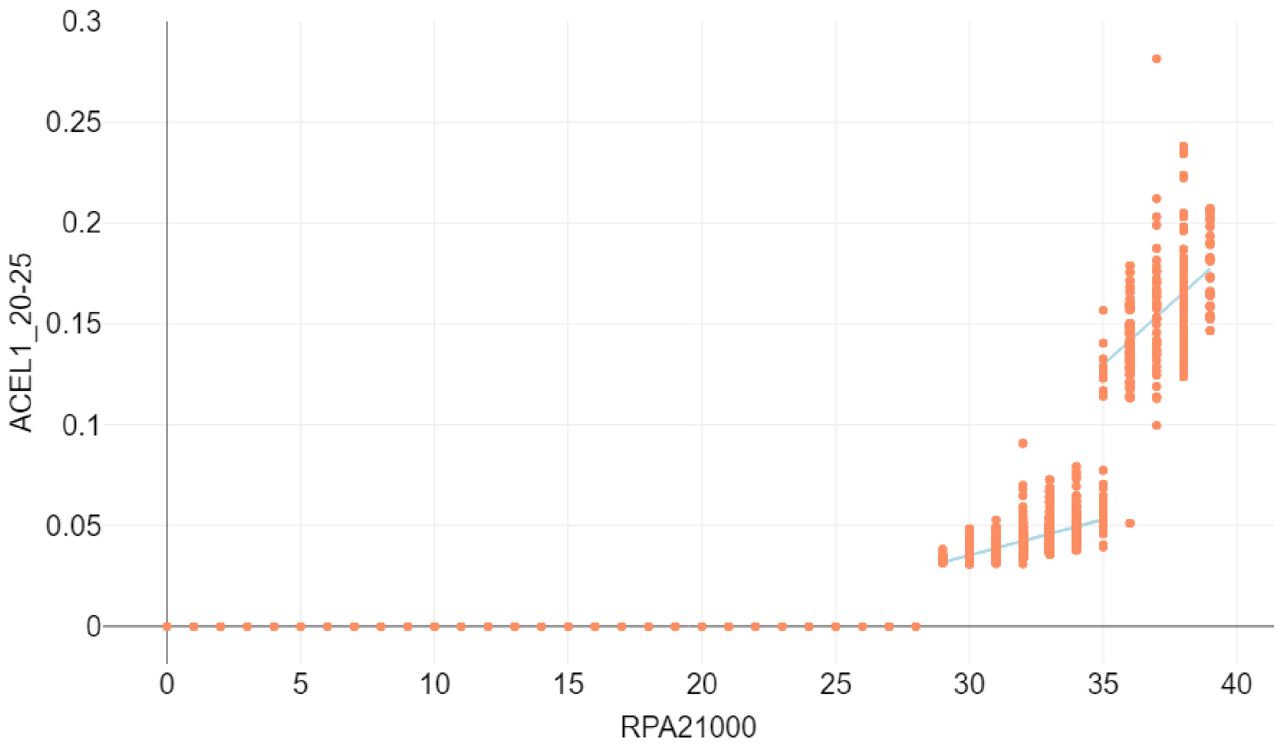


Figure 10: Model aspiration values between 0 and 40 mm H2O

CAS2_sourcecode_2.sql: Program to create the table of values for CAS2 solution.

The source code shows two different parts. The first one is the creation of the table with the whole set of parameters needed from the CAS2 information model into the database of the CAP. The next is explained in section 6.1.2.

- Date time of the measurement.
- Raw value of the sensor (in g_{RMS}).
- Measured pressure (in mm_{H2O}) at the IoT system.
- Calculated value of the sensor based on the model equation shown above and done at the local IoT embedded system.
- Calculated value of the sensor based on the model equation shown above and done at the platform.
- Measured pressure (in mm_{H2O}) as registered by the CAP from the Production dataframes explained in this section.

5.2 Steel domain

The modelling approach used in the steel domain follows the FIWARE modelling guidelines. The model classes are specified in terms of JSON schema and try to adhere to the NGSI-LD cross domain ontology, extending it with a domain-specific model for the steel production. The latter is based on the SAREF extension for industry and manufacturing (SAREF4INMA, https://saref.etsi.org/saref4inma), with certain adaptations. The model has been described to some extent in the CAPRI deliverable D3.4 already, and a standalone publication on the topic is planned. See Figure 11 for a high-level overview.

The model is a core component of the digital twin platform realized in the steel use case and as such it shall be reflected by the CAP REST interface, although at the time of writing some harmonization

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

work is still pending, with the interface following more closely the naming conventions of the data sources/agents than the green field model. Although the FIWARE software is not used in the steel use case, the model could be deployed to a FIWARE-based platform as well.

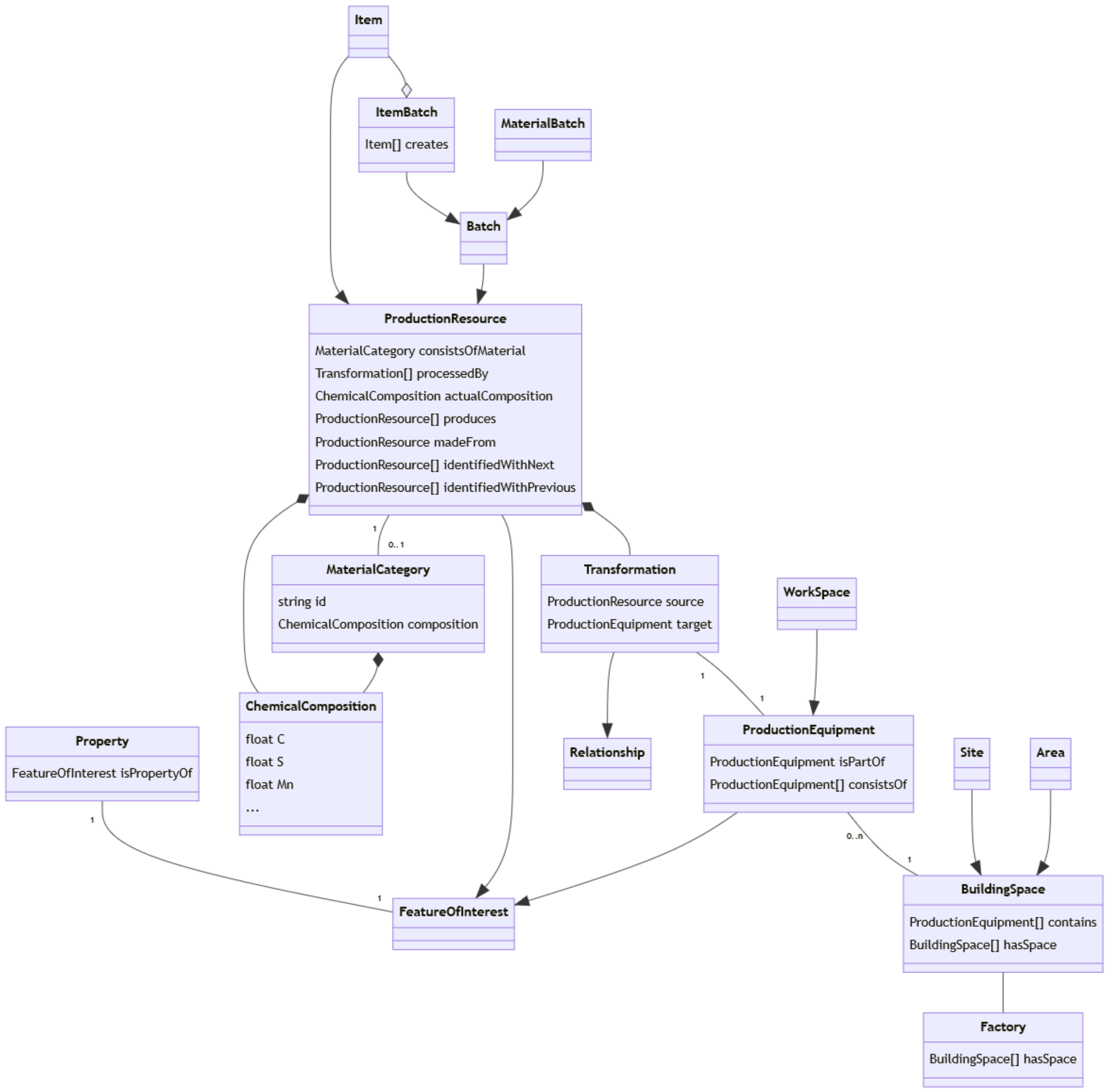

Item
ItemBatch
Item[] creates
MaterialBatch
Batch
ProductionResource
MaterialCategory consistsOfMaterial
Transformation[] processedBy
ChemicalComposition actualComposition
ProductionResource[] produces
ProductionResource madeFrom
ProductionResource[] identifiedWithNext
ProductionResource[] identifiedWithPrevious
1
0..1
MaterialCategory
string id
ChemicalComposition composition
Transformation
ProductionResource source
ProductionEquipment target
WorkSpace
ChemicalComposition
float C
float S
float Mn
...
Property
FeatureOfInterest isPropertyOf
Relationship
1
1
ProductionEquipment
ProductionEquipment isPartOf
ProductionEquipment[] consistsOf
Site
Area
1
0..n
1
1
FeatureOfInterest
BuildingSpace
ProductionEquipment[] contains
BuildingSpace[] hasSpace
Factory
BuildingSpace[] hasSpace

Figure 11: High-level class structure for the steel domain, based on the SAREF4INMA model

An important feature of the model is its ability to assign tracking information to items, by means of the *identifiedWithNext* and *identifiedWithPrevious* relationships. Since they are multi-valued, they can encompass the situation where a steel billet or bar is erroneously identified multiple times at a given location. This is illustrated in Figure 12 (from CAPRI deliverable D3.4).

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

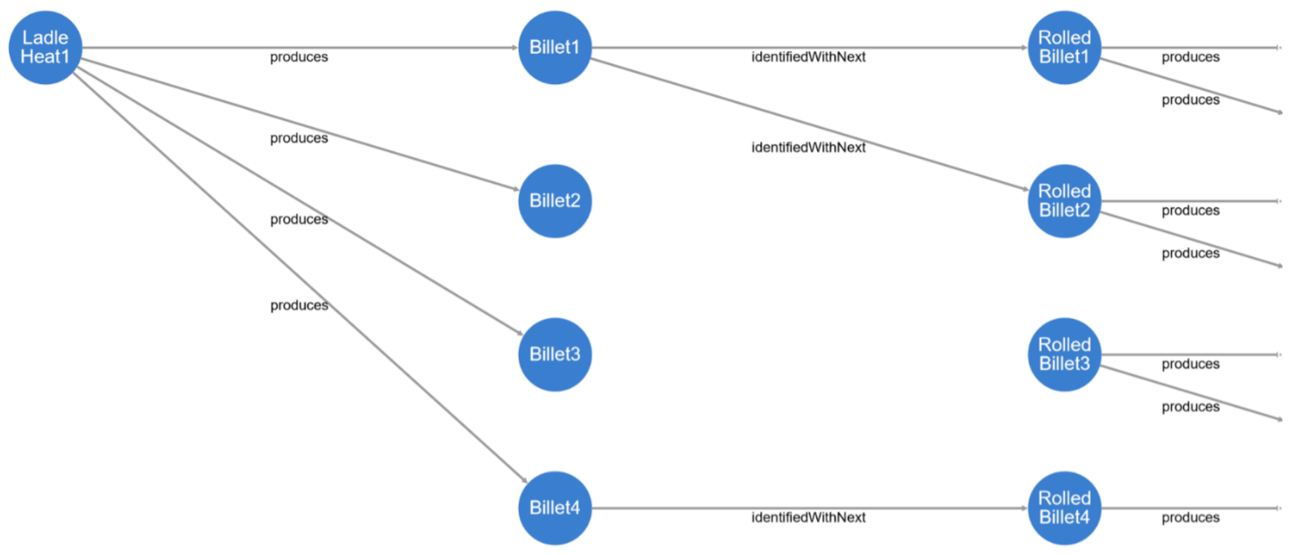


Figure 12: Modeling of tracking uncertainties in the steel use case

The model is not yet fully reflected in the CAP API, which follows more closely the structure and naming of the data sources. The provision of an API aligned to the model presented above is part of the ongoing activities related to digital twins (CSO1) in the CAPRI steel use case, to be reported in deliverable D4.5.

5.3 Pharma domain

In order to investigate the influence of the twin-screw granulation (TSG) process parameters on the quality attributes, e.g., granule particle size distribution (PSD), a screening design of experiments (DoE) was executed. This DoE consisted of 11 runs involving simultaneous variations of liquid-to-solid ratio (LS), solid feed rate (SFR), barrel temperature (BT), and screw speed (SS).

The obtained results indicated a strong relation between the granule PSD (more specifically its first, second, and fourth moment, as well as e_{ref}, which quantifies the deviation of the current PSD from a desired reference distribution) and LS. The influence of other granulation process parameters was not confirmed. The proposed structure of the TSG process model is depicted in Figure 13.

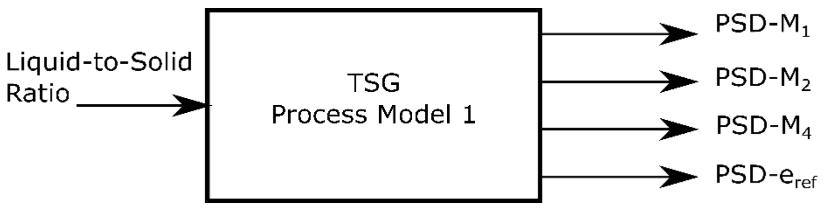


Figure 13: Structure of TSG process model capturing the influence of LS on the granulation quality attributes.

A non-linear process behaviour was observed in the obtained DoE results. Therefore, the model identification procedure was carried out by means of the local-linear-model-tree (LoLiMoT) approach. LoLiMoT is an algorithm for data-driven identification of non-linear systems via weighted local linear models. This implies that the choice of excitation signals is one of the crucial steps for obtaining a model of sufficient quality. In order to precisely reflect system behaviour in the complete operating range, two excitation runs were carried out. The first excitation run involved amplitude

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

modulated pseudo random binary signal (APRBS) LS variations, characterized by random amplitude levels and random level durations. The results obtained in this run were used for the design of the second excitation run. The second excitation run again involved APRBS-like LS variations, but with amplitude levels and level durations optimized such that the average distance between new and already existing data points is maximal.

The LoLiMoT model identification procedure can be summarized in the following steps:

1) The designed input (LS) sequence is applied to the real system (ConsiGma^{TM-}25).
2) The output sequences are collected via available process sensors (Parsum PAT probe + CPS2).
3) The collected input- and output sequences can be utilized as identification data for the LoLiMoT training. The training dataset is shown in Figure 14.
4) The LoLiMoT training is carried out for individual model outputs (M_1, M_2, M_4, e_{ref}).
5) The quality of trained models can be investigated on the validation data set, which is shown in Figure 15.

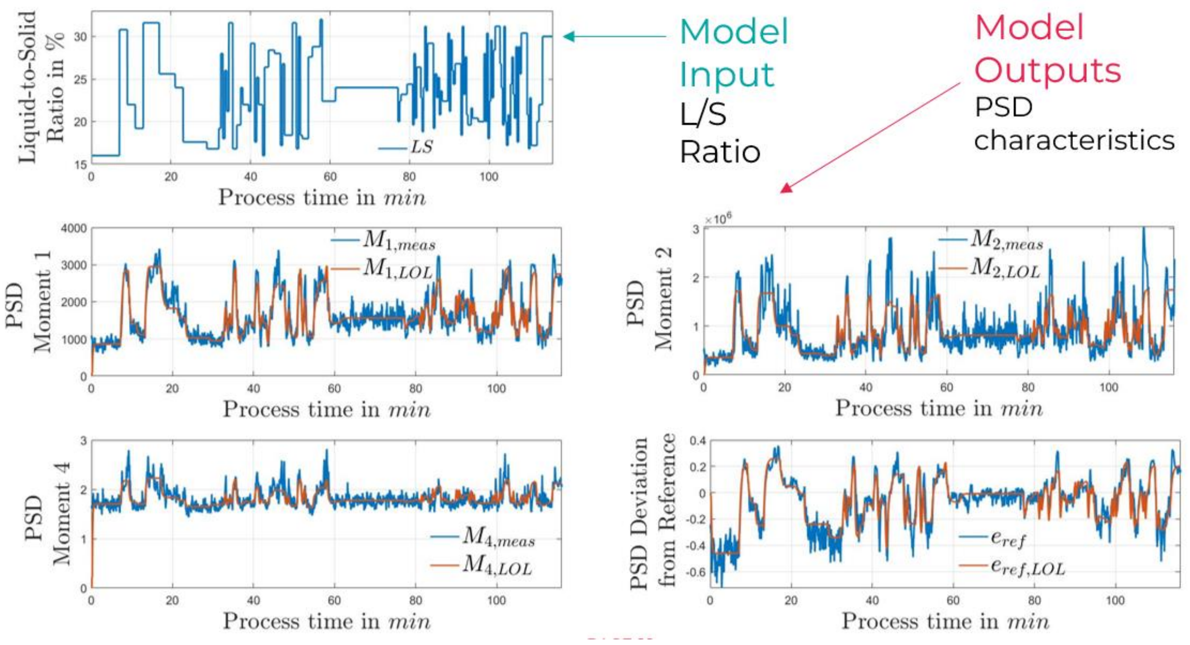


Figure 14: Training data is accurately predicted by the LoLiMoT model.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

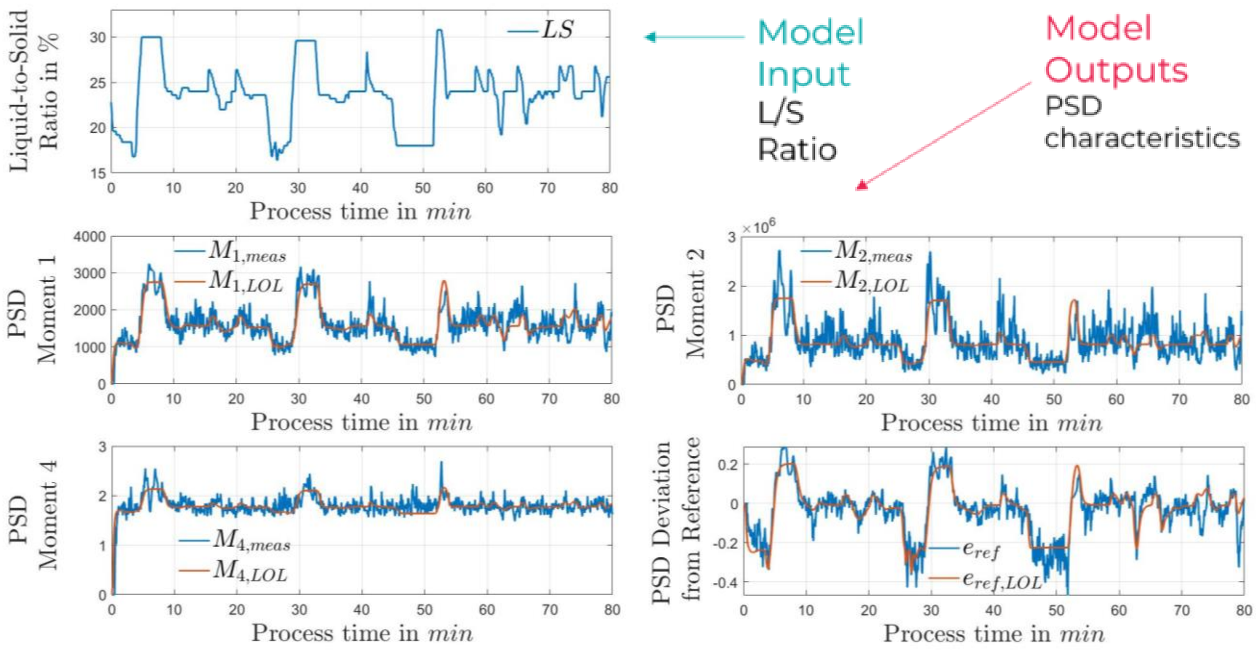


Figure 15: LoLiMoT predicts PSD characteristics accurately on the validation data set.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

6 Sensor Layer Applications

6.1 Asphalt domain

6.1.1 Cognitive sensor of bitumen content in recycled asphalt [CAS1]

CAS1 is a sensor that will provide to the CAP of the asphalt use case the percentage of bitumen content present in the RAP added to the asphalt mix. To do so, this sensor will be connected through an ethernet cable to an 3G router with a static IP, from were CAS1's data will be sent through MQTT protocol to CAPRI's server, where it will be stored and integrated into the Asphalt's CAP. On CAS1_PPT1 it is possible to see the how the data is sent, an example of the data received at CAPRI's server, a plot of the data, and an example of the data storaged at CAPRI's server. The type of data sent can be seen in the file CAS1_data1, and the script used to send the data is in the file CAS1_script1.

6.1.2 Cognitive sensor for amount of filler [CAS2]

Regarding cognitive sensor CAS2 for the measurement of amount of filler flow during the drying of aggregates there are two main applications: a local visualization of the sensor output, and a final application of visualization of input (sensor output) and output (processed) data at the Cognitive Automation Platform (CAP) level.

Below, the different objects included in the OTHER deliverable are presented.

CAS2_video_5.webm: Model implementation at IoT local system un pilot plant.

This video shows the output of the local IoT embedded system located in the Asphalt pilot plant and how the model estimated previously (see section 0 for model explanation) is running locally at the asphalt plant. The resulting model output is proportional to the actual filler flow during drying and can be compared to the same processing done online in the database, as explained below, to effectively validate the calculations done by the IoT system.

CAS2_sourcecode_2.sql: Program to process CAS2 IoT measurements in the database.

Source code that shows the step to recreate the calculations based on the model previously shown but calculated directly in the population of the CAS2 table in the database with the use of a trigger, which is procedural code that is automatically executed in response to certain events on a particular table or view in a database.

In the case of CAS2, each time a new MQTT output is sent from the IoT CAS2 system to the CAP together with the pressure measured from the production frames as explained in section 2, the trigger function '*processStreamCAS2*' is fired and it applies the model estimated (see section 0 for model explanation) and populates the cognitive solution table CAS2 appropriately.

CAS2_video_6.mp4: MQTT dataframes received from local IoT at CAP platform.

In this video it is shown how, in real time, the MQTT agent (mosquito) receives the values from the CAS2 local IoT system into the CAP.

CAS2_video_7.mp4: Population of CAS2 database in the CAP platform.

In this video, it is shown how the new table *CAS2* of the database is populated with the IoT frames together with the calculated values based on the SQL trigger source code shown earlier.

CAP FIWARE integration

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

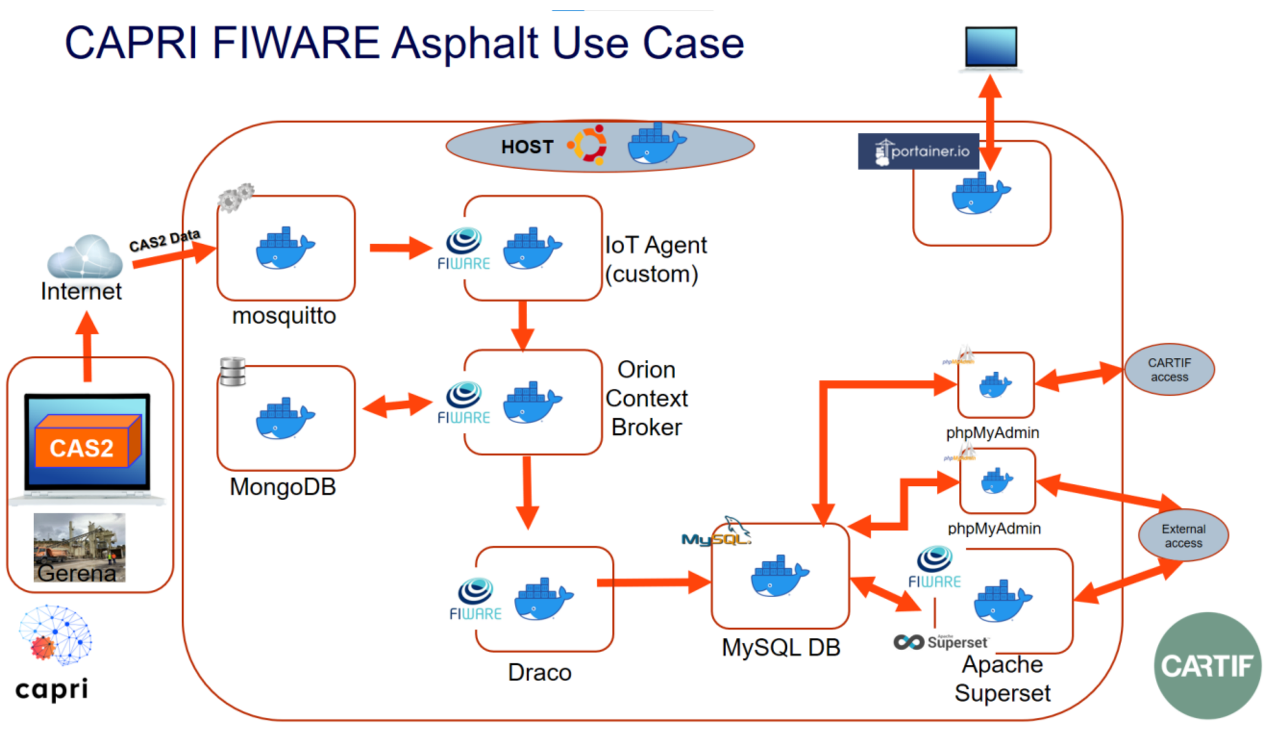


Figure 16: CAS2 integration into the CAP FIWARE-based platform. Main modules are indicated with the FIWARE logo.

As described in section 4.1, CAS2 physical sensors output data are sent using MQTT protocol to the CAP **FIWARE**-based Reference Architecture and stored at **MySQL** database table called *CAS2* where thanks to the trigger *‘processStreamCAS2’* the values of CAS2 are processed with the estimated model and stored in the mentioned *CAS2* table (see Figure 15).

Then again, from this point, it is used by the corresponding visualization module for CAS2 values to be shown at the actual asphalt plant. The Visualization Module, based on **Apache Superset**, can then be accessed through a web-based interface by the plant operators to see CAS2 measured and calculated data (see section 5.1.2). The web-based dashboard showcases some trending graphics and outputs fields.

Requirements to run CAS2 algorithm

The algorithm runs natively in the MySQL database of the CAP platform based on FIWARE.

This project receives funding in the European Commission’s Horizon 2020 Research Programme under Grant Agreement Number 870062

6.2 Steel domain

6.2.1 Cognitive Product Tracking Sensor [CSS1]

The Cognitive Product Tracking Sensor (CSS1) connects the intermediate product in the whole steelmaking production process through a combination of hardware and software that allows a jump of granularity in the tracking in the Basauri production site. In fact, the steel goes through two main transformations but there can be defined up to 5 different states of the product:

- Liquid steel: The whole amount of steel of one chemical composition is refined in the ladle, so, all the measured values correspond to the 130 tons of steel.
- Casting Machine Strand: After the steel is solidified, in the casting machine it becomes a long square bar of some meters. In this moment many relevant data are recorded, but they require a good data preparation as they are not divided in billets, and the billet is suffering different processes in different time moments in the casting machine.
- Billet: After casted, the long square bar is cut into “billets”, the intermediate product. Each billet is marked physically in the CSS1.
- Long bar from the rolling mill: The billet is stored for some days and after that is transformed in the rolling mill to the final round size. The first step of the process consists of checking the tracking data marked in the head of the billet during casting process. This way the Level 2 of the Rolling Mill has this information available. As the mark will be erased in the rolling process. The output of the rolling Mill is a long bar.
- In the last part of the rolling mill the long bar is cut into bars of the final length and it is the moment to mark the bars in their head so they can be tracked in the final finishing operation later.

In consequence, apart from the important hardware consideration for printing and reading information, or sensing, this cognitive sensor has important software development needs. The sensing needs related to the image analysis of the printed QR codes are clear, and the internal tracking inside the Level 2 systems too. However, there is another aspect important for the rest of CSS, it is the data transformation to correctly assign the rest of sensor data collected in time series format to the actual product.

In the video corresponding to this Deliverable this aspect is explained in a graphical view. There are two cases in which this is needed: in casting machine data; and rolling mill data. As the rolling mill marking part of the CSS1 was tried but it is not fully operative after the revamping of the rolling mill, this part is explained in less detail, but considered for future implementation.

The casting machine part of this data treatment is part of the CAPRI project.

6.3 Pharma domain

The solution, in terms of open-source components used and cognitive solutions integrated for Pharma domain, can be found in the relative section of the GitHub repository[1].

6.3.1 Cognitive sensor for blend uniformity [CPS1]

The cognitive sensor for blend uniformity (CPS1) integrates new sensors to the process, as well as the creation and implementation of respective algorithm. The hardware implementation of a Raman probe in the manufacturing process has been involved and the implementation of the algorithm needed to access the raw data from the Raman probe in order to compute meaningful characteristics. The measurement software on the computer stores the raw spectra in files, then an OPC UA server reads these files and provides the contents to the CAP platform via OPC UA protocol.

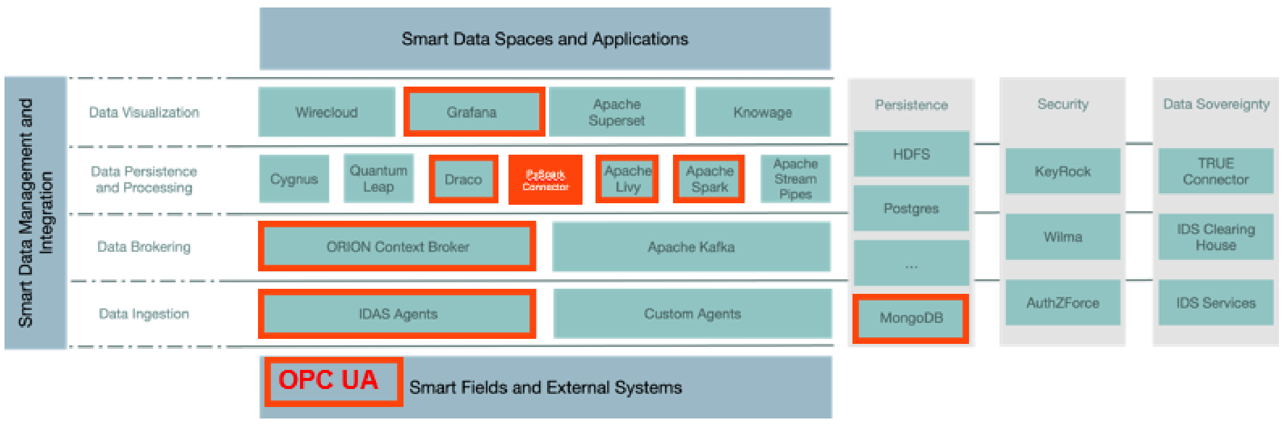


Figure 17: Pharma CAP Reference implementation

So that the entry-point of the CAP becomes the OPC UA Agent, a FIWARE Generic Enabler, which intercepts the data coming from an OPC UA server working as a bridge between the smart field and the Orion Context Broker converting the information collected in NGSI/NGSI-LD format. Since the process is in real-time and the cognitive solution was written using Python, during the integration phase in the platform, there was the need to develop a new tool, the PySpark connector (already incubated to be part of the FIWARE catalogue), which is in charge of creating a bidirectional bridge between the context broker and Apache PySpark. This component works on a low-level socket communication, implementing a message passing interface between the two aforementioned counterparts. The cognitive solution has been improved (from the performance perspective) and integrated here as a Spark job, for which the performance is increased, scaled as much as needed. In this way the feedback provided can be used to adjust the settings of the machine with the aim of improving the quality of the products and also the fault detection of the system. At the end of the process output (and input) data can be visualized using the CAP visualization layer providing interactive dashboards in order to support users in the situational awareness (and control). In Figure 18 and Figure 19 the inputs received from the sensor layer are reported and in Figure 20 the relative output computed in CPS1 can be seen.

[1] https://github.com/Engineering-Research-and-Development/dida/tree/CAPRI/Pharma

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

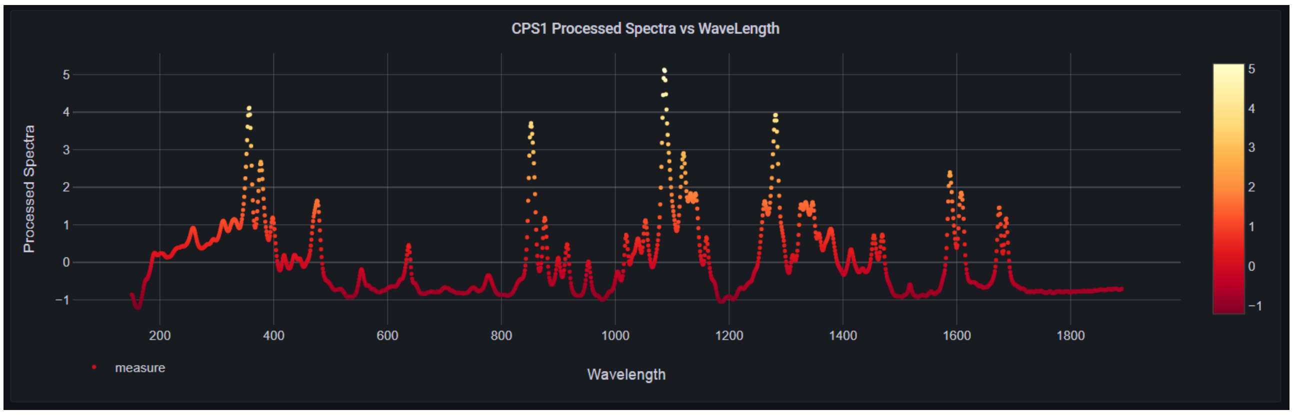


Figure 18: CPS1 Process Spectra vs Wavelength

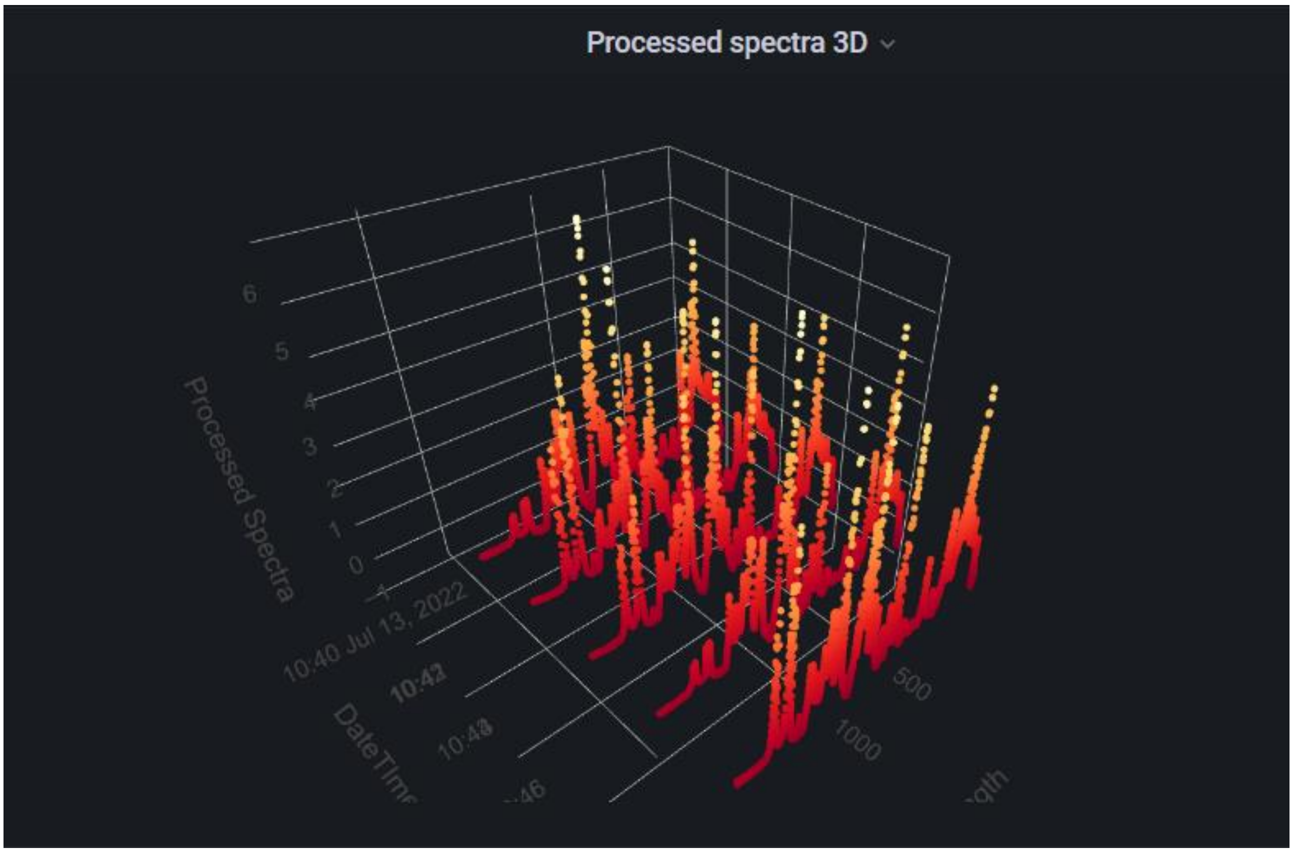


Figure 19: CPS1 Process Spectra vs Wavelength 3D scatter plot

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 870062

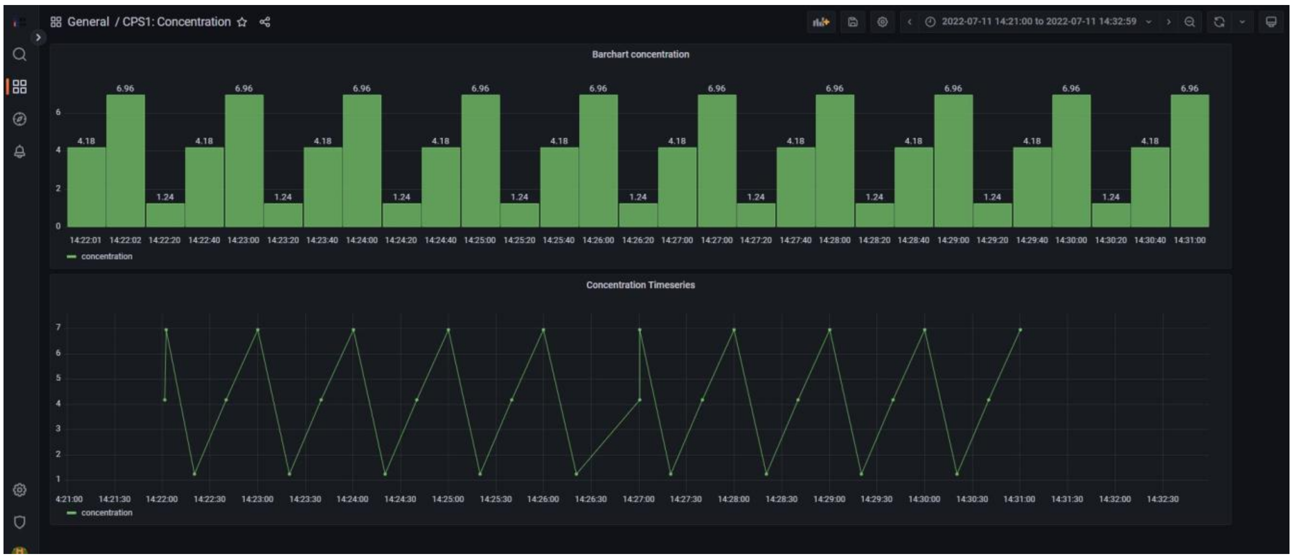


Figure 20: Concentration output from CPS1

In addition to the previously developed model for concentration prediction, a new chemometric model for the prediction of liquid-to-solid ratio (LS) from the Raman spectral data, has been developed. The pre-processing of spectral data has been carried out in the same manner as for the concentration model (please see deliverable D3.2), and the LS model has been identified by means of the principal components regression (PCR) algorithm. Figure 21 shows a comparison of predicted (LS_{Raman}) and actual (LS_{actual}) LS ratio.

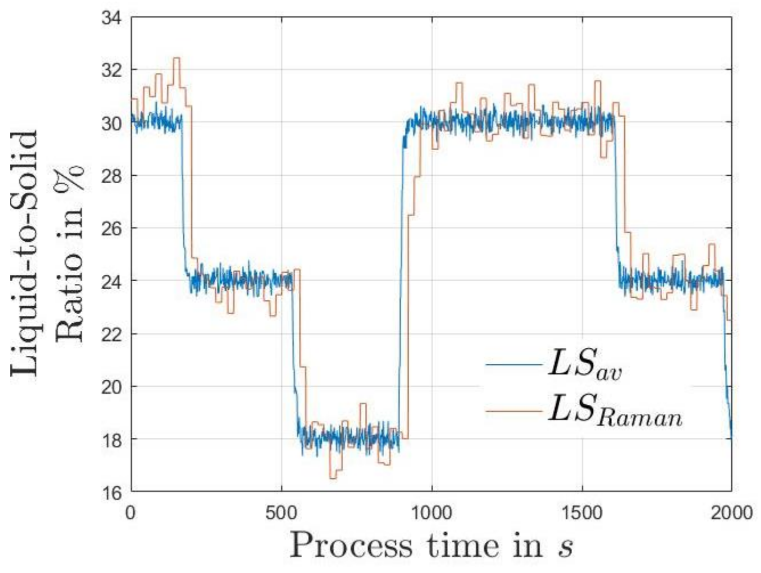


Figure 21: Comparison of real and predicted LS on the training data set.

The contents of the zip-archive related to CPS1 are summarized in Table 2.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

Table 2: Contents of zip-archive CPS1_App_and_Data_2.zip

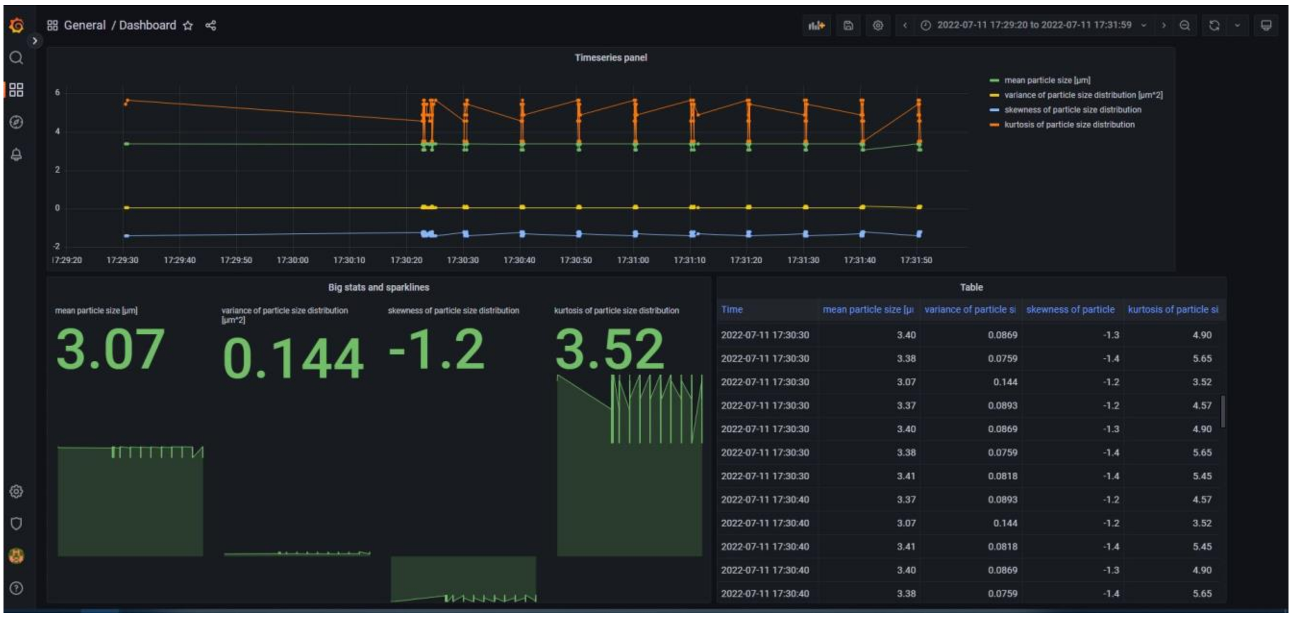
Filename	Description
OPCClient_Prediction_V06.py	Python implementation of the prediction engine used to compute the API concentration from the spectral data. Two different chemometric models are available to be used for API prediction. Another model is used to predict the liquid to solid ratio from the Raman spectra.
KaiserRamanVirtual_V04.py	A test version of the OPC UA server running on the KaiserRaman system. It reads the sample's spectra and provides them via OPC UA. In addition, the tag for the computed concentration value and the motor control tags are provided by that server.
calibspectra\01_0117_Spectrum_20211021-13_56_21_364446.csv calibspectra\05_0189_Spectrum_20211021-14_20_21_186483.csv calibspectra\11_0297_Spectrum_20211021-14_56_21_174045.csv	Sample spectra of the KaiserRaman system. These are being read by the KaiserRamanVirtual_V03.py script and provided via OPCUA.
model\General_List_M13.xlsx	Excel sheet containing the chemometric model data that is needed by OPCClient_Prediction_V06.py for the prediction of the API concentration (first version of the prediction model).
Model\Workset_Statistics_M13.xlsx	Excel sheet containing average spectrum that is needed by OPCClient_Prediction_V06.py for the prediction of the API concentration (first version of the prediction model).
model\General_List_M28.xlsx	Excel sheet containing the chemometric model data that is needed by OPCClient_Prediction_V06.py for the prediction of the API concentration (second version of the prediction model).
Model\Workset_Statistics_M28.xlsx	Excel sheet containing average spectrum that is needed by OPCClient_Prediction_V06.py for the prediction of the API concentration (second version of the prediction model).
model\General_List_M29.xlsx	Excel sheet containing the chemometric model data that is needed by OPCClient_Prediction_V06.py for the prediction of the liquid to solid ratio.
Model\Workset_Statistics_M29.xlsx	Excel sheet containing average spectrum that is needed by OPCClient_Prediction_V06.py for the prediction of the liquid to solid ratio.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

6.3.2 Cognitive sensor for granule quality [CPS2]

The Cognitive sensor for granule quality (CPS2) covered the implementation of a particle sensor (Parsum Probe) and the development of the relative algorithm to compute the relevant particle properties from the provided raw data. As already described in the previous section, the OPC UA protocol is used to transfer data from the physical layer to the CAP platform. In this case, to guarantee the compatibility with the already existing SIPAT server that is in charge of collecting process data, a custom component to convert OPC DA client to OPC UA server has been developed. After that the same technological components selected for CPS1 were used, the same flow was followed by the data and the cognitive solution was integrated as Spark Job. In particular, the PySpark connector, which is in charge of the bidirectional data exchange from FIWARE Orion Context Broker and PySpark jobs, has been validated and improved, making the component more robust and efficient.

The visualization of a sample dataset consisting of several different particle size distributions that are repeatedly provided via an OPC test-server is presented in Figure 22, Figure 23 and Figure 24. The first (mean particle size), the second (variance of particle size distribution), the third (skewness of particle size distribution) and the fourth (kurtosis of particle size distribution) characteristic moment of the size distribution are visualized in different manners. Depending on the operator needs, the most suitable diagram can be used. The gauges used in Figure 23 offer a quickly ascertainable status information on the particle size, whereas the timeseries in Figure 24 allow the identification of timely trends of the size distribution. The dashboard depicted in Figure 22 contains the most comprehensive overview of the CPS2 outcome.



mean particle size [µm]	variance of particle size distribution [µm^2]	skewness of particle size distribution	kurtosis of particle size distribution
3.07	0.144	-1.2	3.52

Table

Time	mean particle size [µ	variance of particle si	skewness of particle	kurtosis of particle si
2022-07-11 17:30:30	3.40	0.0869	-1.3	4.90
2022-07-11 17:30:30	3.38	0.0759	-1.4	5.65
2022-07-11 17:30:30	3.07	0.144	-1.2	3.52
2022-07-11 17:30:30	3.37	0.0893	-1.2	4.57
2022-07-11 17:30:30	3.40	0.0869	-1.3	4.90
2022-07-11 17:30:30	3.38	0.0759	-1.4	5.65
2022-07-11 17:30:30	3.41	0.0818	-1.4	5.45
2022-07-11 17:30:40	3.37	0.0893	-1.2	4.57
2022-07-11 17:30:40	3.07	0.144	-1.2	3.52
2022-07-11 17:30:40	3.41	0.0818	-1.4	5.45
2022-07-11 17:30:40	3.40	0.0869	-1.3	4.90
2022-07-11 17:30:40	3.38	0.0759	-1.4	5.65

Figure 22: CPS2 results dashboard

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

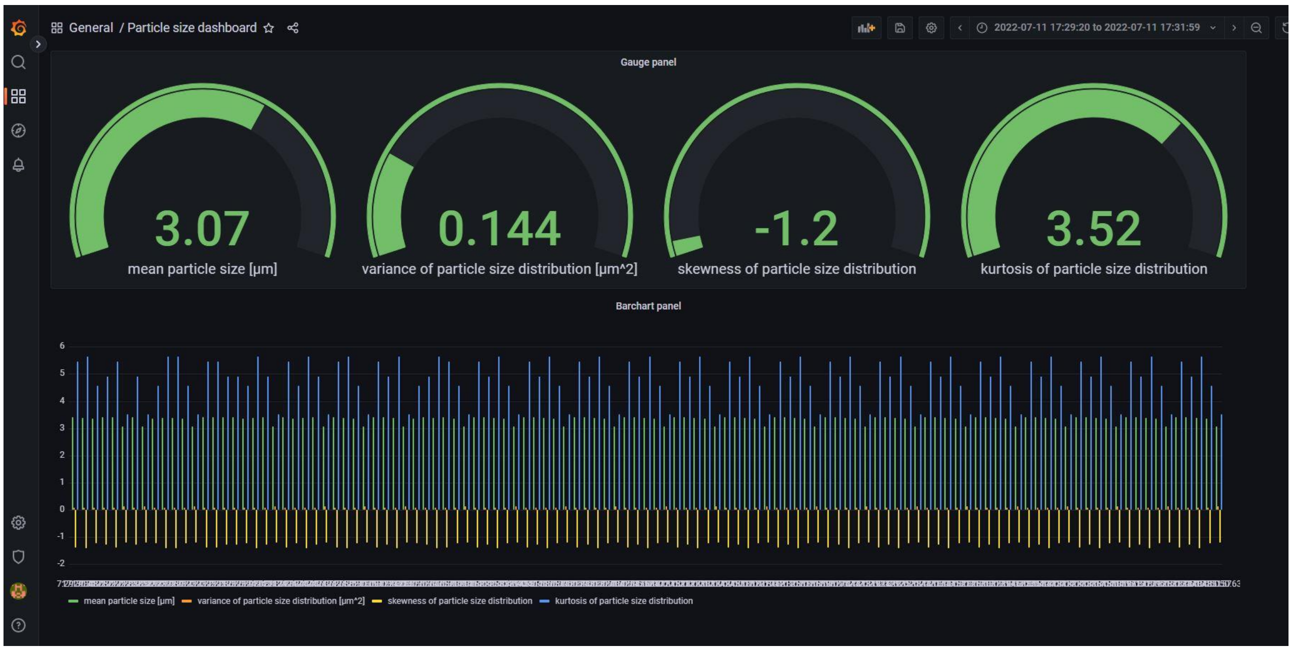

General / Particle size dashboard
2022-07-11 17:29:20 to 2022-07-11 17:31:59
Gauge panel
3.07
mean particle size [µm]
0.144
variance of particle size distribution [µm^2]
-1.2
skewness of particle size distribution
3.52
kurtosis of particle size distribution
Barchart panel
mean particle size [µm]
variance of particle size distribution [µm^2]
skewness of particle size distribution
kurtosis of particle size distribution

Figure 23: CPS2 results as Gauge charts

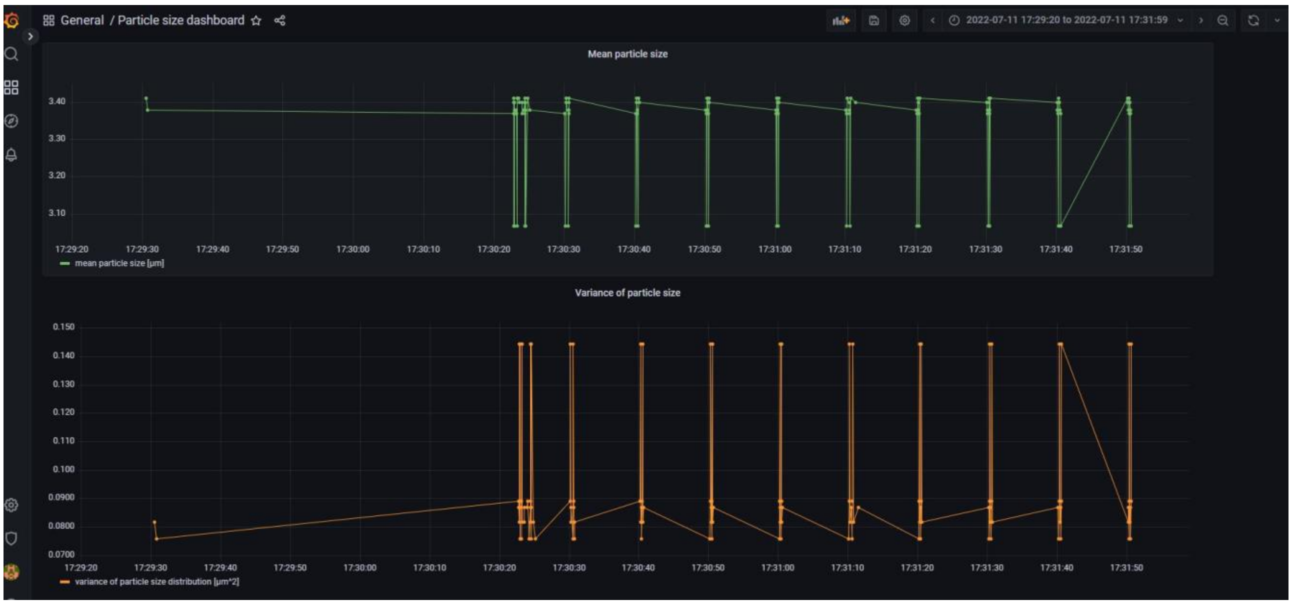

General / Particle size dashboard
2022-07-11 17:29:20 to 2022-07-11 17:31:59
Mean particle size
mean particle size [µm]
Variance of particle size
variance of particle size distribution [µm^2]

Figure 24: CPS2 results as timeseries chart

Characteristic properties

New characteristic granule size properties, i.e., characteristic deviations, have been extracted from the distribution data. The first characteristic deviation is defined as:

$$e_{normal} = \sum_{i=1}^{n} \left| q_{3,i} - q_{3normal,i}(M_1, M_2) \right|$$

and represents the deviation from a Gaussian distribution with the equivalent first and second characteristic moment. This quantity is particularly important for the development of the signal processing concept. Please see the paragraph “Signal processing” below for more details.

This project receives funding in the European Commission’s Horizon 2020 Research Programme under Grant Agreement Number 870062

The second characteristic deviation represents the deviation from the arbitrary reference distribution q_{3ref}, and is described by:

$$e_{ref} = (M_1 - M_{1,ref}) \sum_{i=1}^{n} |q_{3,i} - q_{3ref,i}|$$

For CPC1 purposes, a distribution correlating with the best final product properties and stable process performance will be selected as the reference distribution. Figure 25 shows three sample distributions and their characteristic values e_{normal} and e_{ref}.

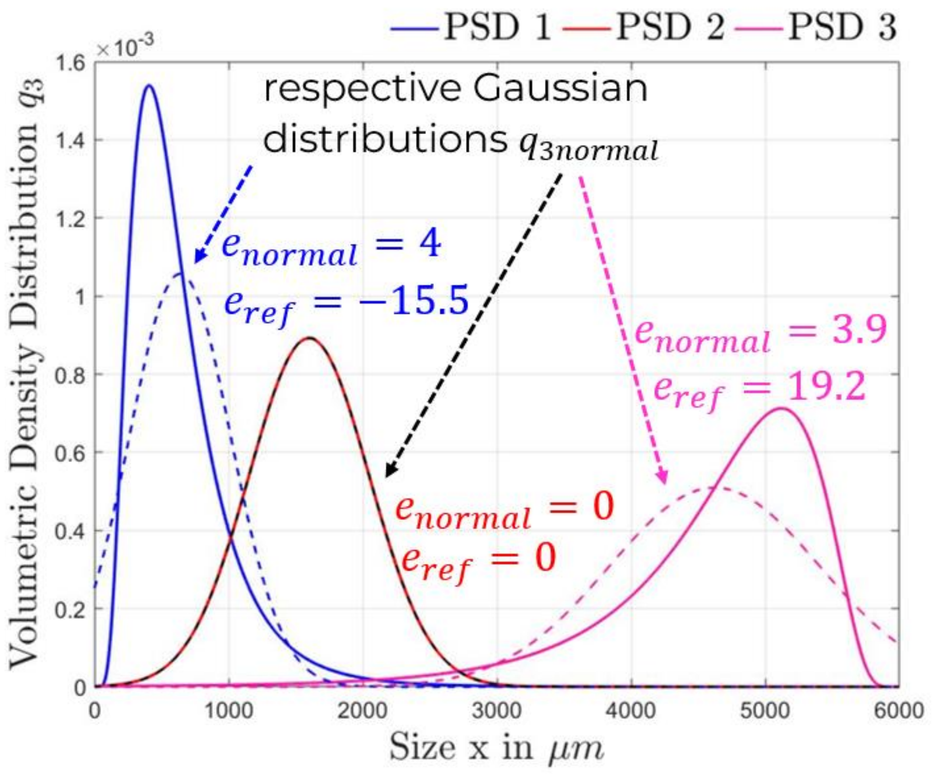


Figure 25: Calculation of characteristic deviations illustrated in the example with three arbitrary distributions. Additionally, previously introduced granule size characteristics, i.e., four statistic moments, are now calculated using the real instead of the logarithmic size axis.

Signal processing

In order to avoid material sticking and potential blockages, the Parsum probe is cleaned via pressurized air pulses. This probe cleaning occasionally results in corrupted PSD measurements, that have no informational contribution, and could even compromise the functionality of the control concept. Therefore, the corrupted measurements should be detected and replaced. For that purpose, the difference between the current e_{normal} and its respective mean value over a certain time window is calculated. If this difference exceeds 2.5 standard deviations, the measured PSD is detected as corrupted and replaced with the respective mean value. The functionality of this concept is illustrated in Figure 26.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

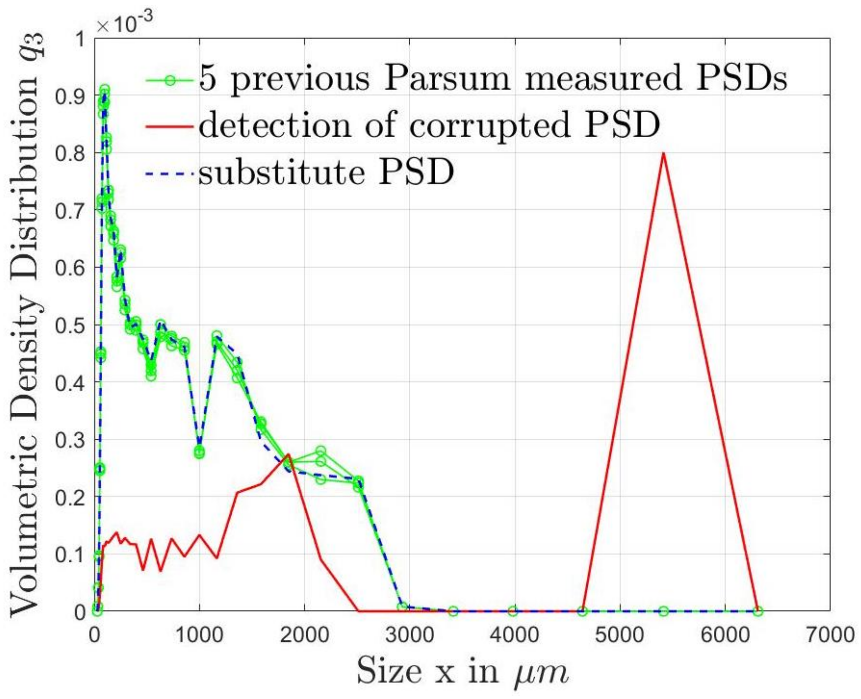


Figure 26: Corrupted Parsum measurement is detected and substituted using the CPS2 signal processing.

The contents of the zip-archives related to CPS2 are summarized in Table 3 and Table 4.

Table 3: Contents of the zip-archive CPS2_Data_2.zip

Filename	Description
description.txt	Text file including a description of the data provided in A3_DOK_CPS2_softsensor_training.xlsx and A3_DOK_CPS2_softsensor_validation.xlsx.
A3_DOK_CPS2_softsensor_training.xlsx	Training dataset of the LOLIMOT model.
A3_DOK_CPS2_softsensor_validation.xlsx	Validation dataset of the LOLIMOT model.

Table 4: Contents of the zip-archive CPS2_App_2.zip

Filename	Description
description.txt	Text file including a description of the provided files.
A3_exe_CPS2_realtime_V02.py	Python implementation of the function that performs the computation of moments.
A3_exe_CPS2_opc_testserver_v02	Test-server that reads sample sieve fractions (test_sieve_vec.csv) and sample psd data (test_q3_[1 .. 5].csv) and provides the information via OPCUA.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

A3_exe_CPS2_tests_V04	A test script that connects to the OPCUA server created by A3_exe_CPS2_opc_testserver_v02 and evaluates the provided sample distributions
test_q3_[1 … 5].csv	Sample size distribution to test the functionality.
test_sieve_vec.csv	Size bins corresponding to the q3 distribution.

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 870062

e

7 Conclusions and Next Steps

The sensor layer reference implementations provide the basis of the CAPRI cognitive automation platform, being responsible for the data ingestion from various sources. We have realised the layer for the three CAPRI use cases in the process industry domains asphalt, steel, and pharma. The implementations are based on different technologies but share a common architecture. They rely on open-source software and should be replicable in other process industries.

Connecting to a variety of data sources is an important feature that the sensor layer must support. Standardized data agents for different communication protocols simplify this task, such as the open-source OPC UA agent for an NGSI-LD broker, which is used in the pharma domain. For the data processing, a connector between the broker and the PySpark runtime has been developed, also for the pharma use case, and published under an open-source license. The PySpark Connector is now part of the FIWARE Catalogue as an incubated FIWARE Generic Enabler of the core chapter.

The importance of semantic metadata and linked data for cognitive applications has been emphasized and will be further elaborated in the upcoming deliverable D4.5.

With the sensor and control layers available, the next steps are the finalization of the upper CAP layers, related to operations and planning, as well as the prototype demonstration runs. For the latter purpose, the cognitive solutions will be deployed to a productive CAP instance and will operate in live mode for 6 months. At the end, the KPIs defined at the beginning of the project will be evaluated.

This project receives funding in the European Commission's Horizon 2020
Research Programme under Grant Agreement Number 870062