Industrielle Gemeinschaftsforschung

Schlussbericht

zu CORNET / IGF-Vorhaben Nr. 228 EN

Thema

Optimierte Sensorverteilung für qualitative Datenerfassung

Titel des CORNET-Gesamtprojekts:
(Actionable Insights into Product Service Delivery – "Insight Products")

Berichtszeitraum

01.09.2018 bis 31.12.2020

Forschungseinrichtung(en)

Sirris, Brüssel, BE

Institut für Mikro- und Informationstechnik, Villingen-Schwenningen, DE
(Forschungsvereinigung: Hahn-Schickard-Gesellschaft für angewandte Forschung e.V.)

FZI Forschungszentrum Informatik, Karlsruhe, DE

J. Dehnert S. Reiter

Villingen-Schwenningen, 30.06.2021 — Jan Dehnert (Hahn-Schickard), Sebastian Reiter (FZI)

Ort, Datum — Name und Unterschrift aller Projektleiterinnen und Projektleiter der Forschungseinrichtung(en)

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

Forschungsnetzwerk Mittelstand

Der vorliegende Bericht ist sowohl als **CORNET Gesamtschlussbericht** als auch als **IGF-Schlussbericht** zu betrachten. Die durchgeführten Arbeiten und erzielten Ergebnisse sind der jeweiligen Forschungsstelle zugeordnet.

The present report is to be considered both as a **CORNET overall final report** and as an **IGF final report**. The work performed and results achieved are assigned to the respective research unit.

1 Table of contents

2 Introduction

Many companies, in particular manufacturing, are eager to learn how to collect and mine data about their products and use it in order to improve their service offering. However, they lack the necessary insights into novel technologies as well as how to deploy and use them.

Acquiring qualitative product data requires a well-thought and optimized sensor selection and deployment, adapted to the product infrastructure, configuration and operating environment.

To help companies in this process, Sirris, Hahn-Schickard and FZI Forschungszentrum Informatik have initiated the collective research project 'InsightProducts - Actionable Insights into Product Service Delivery' under the CORNET framework.

The rise of sensing, IoT, Big Data for fleet management and the shift from preventive to predictive maintenance drive the growth of the machine condition monitoring market. For instance, electricity utility companies are capitalizing on the increasingly large amount of data available by opening up their wind farm data sets in order to benefit from a community effort of improved wind turbine operation.

However, the increasing complexity of the manufacturing, automotive, energy and medical systems domains requires robust techniques for real-time monitoring, for detecting the inception and progression of faults, for reducing downtime and increasing reliability, and for enabling flexible maintenance scheduling.

These require optimal deployment of (wireless) sensor nodes, taking into account:

- trade-offs between sensor count reduction and a high degree of system reliability
- data analysis overload & system cost due to the generation of irrelevant/conflicting data
- focus shift from product delivery to agile delivery of product-enabled / -enabling services

3 Tasks performed and results

The following section is a summary and technical documentation of the main results and work carried out in the project InsightProducts, - IGF-Vorhaben Nr. 228 EN – performed in the period 01.09.2018 until 31.12.2020.

3.1 WP1: Project management

This work package addressed the overall project management, communication with the project steering committee, communication between the German research partners and the Belgian project coordinator.

After the initial national organization of the project steering committee, from the German side 10 companies participated, virtual meetings of the overall transnational project steering committee (Germany, Belgium) were organized. On the one hand, this promoted the exchange between the industrial partners, but above all, the competences and results of the research partners could be made available to the industrial users on a transnational level. In addition to the requirements specification, given in an initial phase, the project steering committee provided significant support in the design of the different project demonstrators.

The organizational and technical exchange between the German and Belgian research partners took place primarily via monthly virtual coordination. For the work on joint demonstrators and for kick-offs, corresponding face-to-face meetings were organized.

3.2 WP2: Specification of SME requirements & validation data

Work performed

In coordination with the project steering committee and based on general SotA research, requirements for comprehensive data recording and evaluation in technical systems were established within the scope of this work package. In addition to the discussion during the first meeting of the project steering committee, the status of the current systems of industrial partners as well as desired improvements were queried with the help of a questionnaire. Based on this information, the research partners derived requirements for the solutions to be developed.

In addition to guiding the technical research activities, the results of this discussion led to the definition of demonstrators and validation metrics. Both were used to show the realization of the industrial requirements to the partners.

Results achieved by Sirris

In order to extract the requirements of the Belgian industrial steering committee members, the InsightProducts consortium organized a first UG meeting in Belgium on October 11, 2018. During this meeting each company was asked to provide (based on a template provided upfront) a short presentation of the company and of its activities, including an overview of the current product insights and challenges and of the future product enhancements and needs specifically matching the InsightProducts project objectives. Two versions of the presentation template were provided, one targeted to problem owners and another one targeted to technology providers. Based on these presentations, a preliminary analysis was executed. For some companies, when more detailed information was deemed needed, a bilateral meeting was planned during the months that followed that first UG meeting. The challenges and expectations expressed by these companies were then consolidated and a set of UG requirements in order to drive the rest of the project were identified.

In order to define several demonstrators in the scope of the project, we have identified several conceptual setups and related metrics supporting multiple identified requirements. Based on these a preliminary infrastructure has been identified. During the 2nd user group meeting in March 25th, the 5 identified demonstrators were presented together with the suggested test infrastructure. The companies were given a questionnaire in order to capture their preferred setup and possible suggestions for infrastructure or data sets that could be used.

Based on these, additional meetings were planned and a detailed description of the selected demonstrator infrastructure further specified.

Results achieved by Hahn-Schickard

When considering the SME requirements at Hahn-Schickard, particular attention was paid to useful sensors and sensor systems, sensor network topologies, transmission technologies, storage and data processing.

To obtain the requirements of the SMEs, they were interviewed at the UC meetings and bilateral discussions were held with some of the individual UC members. Together with one UC company, a producer of assembly line transfer systems (~conveyor belt), Hahn-Schickard had a discussion about surveying and enhancing the reliability of their system with the help of sensor based monitoring system. A demonstration system was already available at the company and could have been used for tests. But as this transport system already works very reliable and it was not likely to observe degradation/malfunctions within a reasonable period of project time, Hahn-Schickard preferred a more general setup approach directly on site at Hahn-Schickard to demonstrate the use of modular sensor building blocks and the assembly of a distributed sensor system, also taking into account the input of the other UC members.

The demonstrators available for selection were the transport system just mentioned, a simple condition monitoring system for an electric motor, and a cleanroom demonstrator for monitoring various production machines, pumps and systems. Hahn-Schickard decided to proceed with the last approach because it promised a very extensive and versatile realization. Different sensors for measuring physical quantities such as temperature, acceleration, body born noise, acoustic noise air-humidity, light intensity and power consumption should be applied to the machines for condition monitoring and predictive maintenance applications.

Important points regarding validation criteria were: simple possibility of mounting the sensor(s) wireless and wire-based, data rate and latency, transmission reliability, organization and structuring of gained data, easy visualization of gained data and results, detection of special events, fails and anomaly, monitoring the current state of machines

Results achieved by FZI

The FZI focused on requirements and use cases regarding the availability and interpretation of data in complex systems.

Here, the communication infrastructure and the data available in today's vehicles were discussed with an industrial partner of the project steering committee. Of particular interest were the available data formats and their underlying semantics. Some of these semantics have already been explicitly specified via separate documents, e.g. in the case of the CAN bus. The goal was the transfer and further development of current automotive solutions for data interpretation to the domain of industrial automation.

Approaches and requirements for the interpretation of data were discussed with another steering committee partner. The industrial partner is active in the domain: simulation of complex systems and described problems with the subsequent interpretation of simulation results, e.g. of archived data. The presented problems could be mapped, in a comparable way, to the recording of sensor data in industrial automation. In addition, the partner uses a comparable tool environment as the solution targeted by the FZI, which is why, in addition to the requirements at system level, an exchange regarding tool realization also took place within the project. In particular, the integration of the Web Ontology Language (OWL) into the Eclipse Modeling Framework (EMF) and thus the potential use in the context of SysML modelling represented an interesting exchange. These discussions were complemented with requirements and concepts for the visualization of extensive datasets.

Based on these discussions and taking into account the further project-wide coordination, a demonstrator concept to be implemented by the FZI was developed. The focus was on the demonstration of the concept of an explicit and always available semantic of the data (demonstrator part: combination of semantic enrichment & EMF) as well as concepts for the visualization and interpretation of the data (demonstrator part: visual WSN editor). In coordination with the partners, relevant tools, namely EMF, were considered in the demonstrator. The work on the demonstrator concept was completed by the definition of metrics, which were evaluated with the help of the demonstrator at the end of the project. The metrics mainly focuses on the overhead created by the targeted solution with an explicit semantic.

3.3 WP3: Technological analysis and knowledge transfer

Work performed

Based on the requirements and industrial challenges from WP2, this work package reviewed the state of the art and reflected corresponding technologies back to the project committee. Three areas were identified in the project and used to group the SotA. Namely, these were the fields of *Technology*, *Operation* and *Business*. The Technology field relates to architectural design solutions, sensor and communication solutions, and data exchange and storage solutions. Operations deals with how to ensure product operation while at the same time deriving benefits from intelligent, networked products. The third area Business focuses on how to innovate within a company's business goals based on the data collected from smart, connected products. The main goal is to develop and offer data-driven digital services (digital servitization).

Within this work package, a repository of existing technologies was documented, to be provided to the industrial steering committee. This covers the technologies for exiting sensors, commercial condition monitoring solutions or different communication protocols/standards. Supported by an exemplary overview of existing execution platforms, both commercial and open source. For the application domain (compare WP4), different data processing frameworks are listed. While working jointly in created this overview, each partner was responsible for a dedicated area, based on his experience and expertise. In the following, each partner gives a short introduction in his monitored SotA.

Results achieved by Sirris

For this technology repository we provide a detailed study of the industrial leaders in the scope of InsightProducts and of the positioning of members of the industrial steering committee with respect to the former. The study is conducted based on the requirements of the UG linked to the InsightProducts project objectives.

Especially, solutions for data acquisition and processing as well as servitization and business models where reviewed and documented.

IoT platforms collect via the Internet data from hundreds of sensors embedded into connected devices, and enable to monitor and manage these devices. Such platforms are often referred to as a middleware solution, as they mainly act as a messaging system between the edges, i.e. the devices and a backend system processing the collected data. Often, open source IoT platforms offer some (basic) functionally for data transformation, analysis and visualization. Commercial IoT platforms can be more extensive. In the scope of the projected we analyzed a (non-exhaustive) list of IoT platforms from different vendors, each with different functionalities and scalabilities.

For data processing also a (non-exhaustive), but extensive list of solutions for processing data was researched and presented to the industrial steering committee. The solutions where structured in two groups covering, respectively, general-purpose solutions as well as specific-purpose solutions. In the following two exemplary entries for each category are given.

General-purpose solution: Apache PredictionIO

- Description: Apache PredictionIO provides an open source machine learning server, including Apache Spark, MLlib, HBase, Spray and Elasticsearch. Applications built with PredictionIO are composed of a so-called Event Server, responsible for collecting and unifying data (possibly in real-time), and of a so-called Engine, responsible for making a "prediction" based on that data, upon a request received by the application. The term "prediction" is to be understood in a very broad sense, as this can be any data science task, including classification or clustering. PredictionIO provides several Engine templates, focusing on product recommendation and on performing natural language processing related tasks.
- Application context:
    - Domain agnostic
- References:
    - http://predictionio.apache.org/index.html

Specific-purpose solution: Yanomaly by the Belgian UC-member Yazzoom

- Description: Yanomaly is a software product built by Yazzoom that supports the extraction of valuable insights from machine data and log files. Yanomaly is built on top of open-source technologies such as Logstash, Apache Kafka and Elastic Search and provides process mining (i.e. extracting from machine data information about how the machine was actually used) and real-time and context-dependent anomaly detection functionality.
- Application context:
    - Industrial domains
        - Constraints and limitations:
    - Works on machine data and on log files.
- References:
    - https://www.yazzoom.com/
    - https://www.machine-analytics.com/

Results achieved by Hahn-Schickard

The focus of Hahn-Schickard regarding the SotA assessment was on existing MEMS Sensors and sensor systems as well as wireless sensor platforms and technologies.

Existing Sensors w.r.t. condition monitoring

As very basic building block of a sensor system, MEMS-Sensors for different physical quantities from several manufactures were researched and listed, as well as known sensors already used in previous projects

Inertial Sensors:
Measured quantities: acceleration, angular rate, (magnetic field) for measuring vibrations, structure-borne sonic, rotation speed, orientation in space
Samples: Bosch BMA255, SMB470 (high bandwidth), BMI085, InvenSense 20602

Environmental Sensors:
Measured quantities: Temperature, humidity, pressure, (air quality/CO_2)
Samples: Bosch BME280, BME680

Temperature: TMP112

Light intensity: OPT3001, VCNL4020

Microphones: Bosch AKU350, Infineon XENSIV™, TDK SmartSound™

Data Processing/Wireless processors:

Wireless Processors: TI CC2650, TI CC1350, TI CC1310, Nordic processors

Calculation Processors: STM Cortex M4F

Known existing commercial condition monitoring solutions

At the beginning of the project, the following commercial condition monitoring solutions were available to the public or announced.

ABB

- Description: ABB Ability™ - Condition Monitoring for motors and other components
- Reference: https://new.abb.com/drives/de/condition-monitoring-antriebsstrang

Siemens

- Description: SIPLUS CMS - Condition Monitoring for motors and others
- Reference: https://new.siemens.com/global/en/products/automation/products-for-specific-requirements/siplus-cms.html

Bosch

- Description: Intelligent Vibration Analysis Sensor (IVAS) - Condition Monitoring Sensor based on structure-borne sound for condition monitoring of moving parts like pumps, motors, linear drives, ventilation systems, bearings
- Reference: https://www.bosch-connectivity.com/products/intelligent-vibration-analysis-sensor

Existing low-performance pre-processing algorithms

- Name of the solution: Random Forest Classifier

- Description: A Random Forest is a classification procedure that consists of several uncorrelated decision trees. All decision trees have grown under a certain kind of randomization during the learning process. For a classification, each tree in that forest may make a decision and the class with the most votes decides the final classification. Random Forests can also be used for regression.

- Application context: The classifier trains very fast. Efficient for large amounts of data. Already implemented in Scikit-learn (Python)

Existing wireless communication protocols

Existing wireless communication technologies that could be considered for wireless data transmission in the project were WiFi, Bluetooth Low Energy (BLE), ZigBee, Z-Wave, 6LoWPAN, EnOcean, LoRa, Ant and NFC. The comparison of the technical parameters of the different technologies as well as parameters like availability, ease of use and price were considered when selecting the protocols for use in the sensor modules and in the gateway components for the general system architecture.

Results achieved by FZI

The FZI supported the technology repository with SotA research regarding potential system architectures and data exchange protocols. With the sketch of a generic architecture as shown in Figure 1 a gateway centric architecture as well as hierarchical, peer-to-peer and a service-oriented architecture where described. For the different architectures, various flavours such as pre-processing in the gateway or on a separate device are presented.

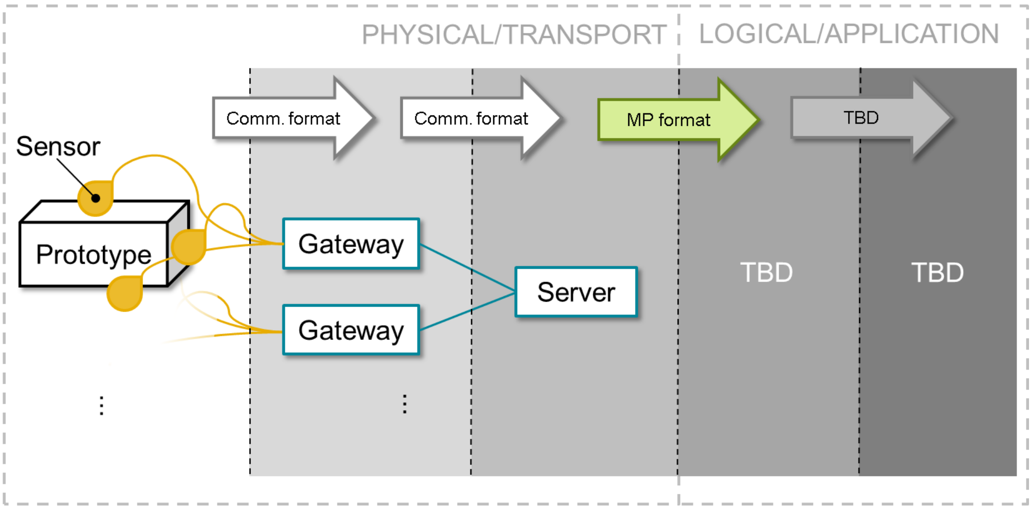

Figure 1: Generic gateway based architecture

Based on this discussion different SotA protocols such as the Message Queuing Telemetry Transport (MQTT), Open Platform Communications Unified Architecture (OPC UA), Wireless Profibus, Mission Profile Format (MPFO), Building Automation and Control Networks (BACnet), Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protoctol (XMPP), Data Distribution Service (DDS), Streaming Text Oriented Messaging Protoctol (STOMP) or Konnex (KNX) are presented. Even some obsolete techniques such as Simple Object Access Protocol (SOAP) or Simple Sensor Interface Protocol (SSI).

To give an impression of the presented information, exemplary the OPC UA entry should be highlighted here:

Open Platform Communications Unified Architecture (OPC UA)

Description: OPC UA is a M2M communication protocol targeting industry, which is based on a service-oriented architecture (SOA). Network outages and malfunctions addresses the protocol through buffering, redundancy and heartbeat features. The protocol provides security via authentication, authorization, encryption and data signing. Regarding the network transport layer, the OPC UA specification currently contains two protocols: A TCP/IP based binary protocol and a SOAP based web service protocol. As OPC UA service descriptions are basically abstract method descriptions which are not protocol dependent it is therefore possible to use other transport protocols.

Message Queuing Telemetry Transport (MQTT)

- Description: This standardized M2M communication protocol is based on a client-server publish-subscribe architecture. The protocol takes network limitations such as delays, downtimes and other network deficiencies into account. In addition to that performance differences between participating devices are considered. Therefore, it is highly suitable for IoT applications. Moreover, encryption of network traffic is supported, to protect sensitive data. MQTT is flexible with regard to the transport network layer and supports TCP/IP as well as non-TCP/IP based networks such as ZigBee. In particular, for sensor-devices exists a dedicated specification MQTT For Sensor Networks (MQTT-SN), which is for Wireless Sensor Networks (WSN).
- Application context:
  - Telemetry data transmission
  - IoT applications in general
- Constraints and limitations:
  - Enforces a client-server architecture (server is message broker)
- References:
  - https://mqtt.org/ (official website)
  - https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html(specification)
  - http://www.mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf(MQTT-SN specification)

3.4 WP4: Technological building blocks, integrated architecture and business models

Work performed

In WP4, the research partners developed a generic architecture concept that relates the essential components of a sensor architecture. Based on the SotA research in WP3, a technology catalogue was created in which available technologies, such as IoT platforms, communication protocols, algorithms for pre-processing, communication technologies (both wired and wireless) were briefly presented. Both the generic architecture concept and the technology catalogue, which provides solutions for instantiating the architecture, are intended to provide industrial partners with an overview and facilitate their entry. Furthermore, this catalogue was the basis for the research partners to identify gaps or weaknesses in current solutions available on the market.

The procedure for instantiating the generic architecture is presented here as an example. The process of defining the InsightProducts architectural layers involves splitting up the problem in the three decision points as depicted in Figure 2:

- Selection of suitable communication & sensor solutions - depending on several application requirements such as operating environment, usage & fault-tolerance. The result is an optimal communication & messaging in view of relevant data acquisition.
- Optimal use & placement of sensors – in order to guarantee qualitative data acquisition, the amount, use, e.g. sense rate, and location of the sensors plays a major role.

- Qualitative data acquisition – this process includes several steps (depending on the type and format of the data) involving cleaning, preprocessing, annotation & removal of redundant measurements.

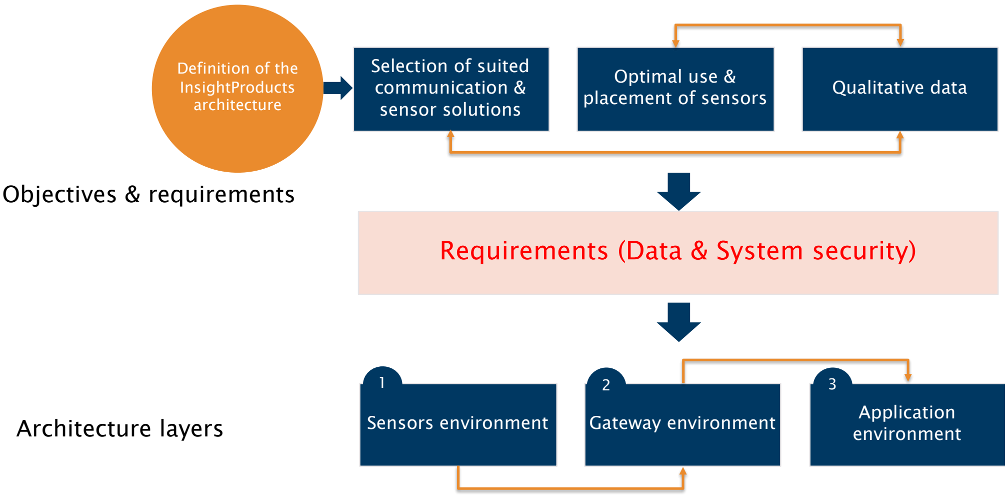

Figure 2. Architecture definition process.

In the context of this process definition, three major building block groups — sensor, gateway and application environment — depicted in the last row of Figure 2 have been identified. These building block groups, further detailed in Figure 3 and in Figure 4, can be instantiated depending on specific industrial requirements. Figure 3 depicts a simplified design of the three building block groups, which include:

- Sensors – a micro-controller (MC) that has built-in communication standard such as WIFI or Bluetooth, and connects and publishes data to the gateway.
- Gateway – a device gateway with two main functions: basic verification of the sensor data and anomaly detection.
- Application – encapsulating the main data analysis activities.

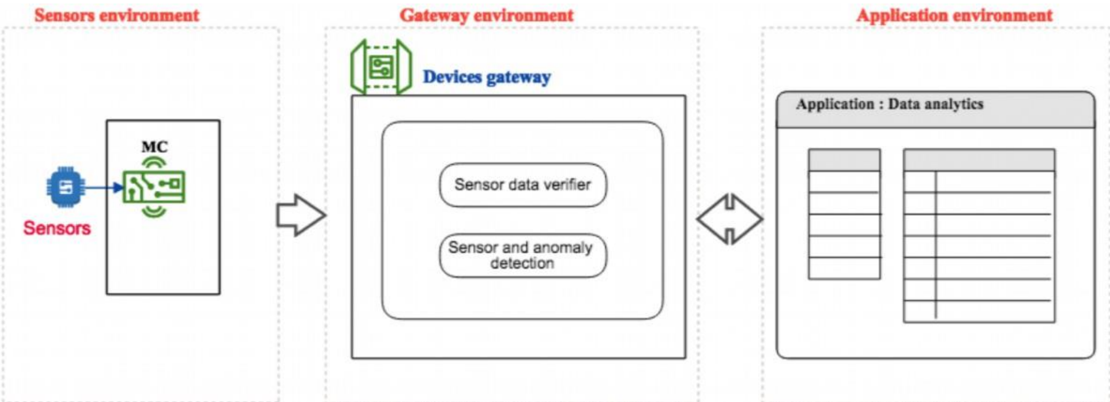

Figure 3. Generic architectural building blocks.

Sensors environment: This module consists of sensors and micro-controllers. A sensor is used to detect and respond to specific inputs from the physical environment. A micro-controller is used to communicate the outputs of the sensors to a gateway. Different types of sensors and micro-controllers with different communication technologies (e.g. wireless or wired) can be used.

Gateway environment: This module is responsible for relaying and pre-processing data from the sensors.

Application environment: The main role of the application module is to analyse the sensor data and detect anomalies during system operation. The application takes data from the sensors as input and provides feedback on the state of system as output.

Results achieved by Sirris

Sirris was in lead on designing the generic architecture. In the following an instantiation for the generic architecture is shown. Figure 4 instantiates the building blocks of the generic architecture applied to one of the InsightProducts demonstrators (SPICY demonstrator).

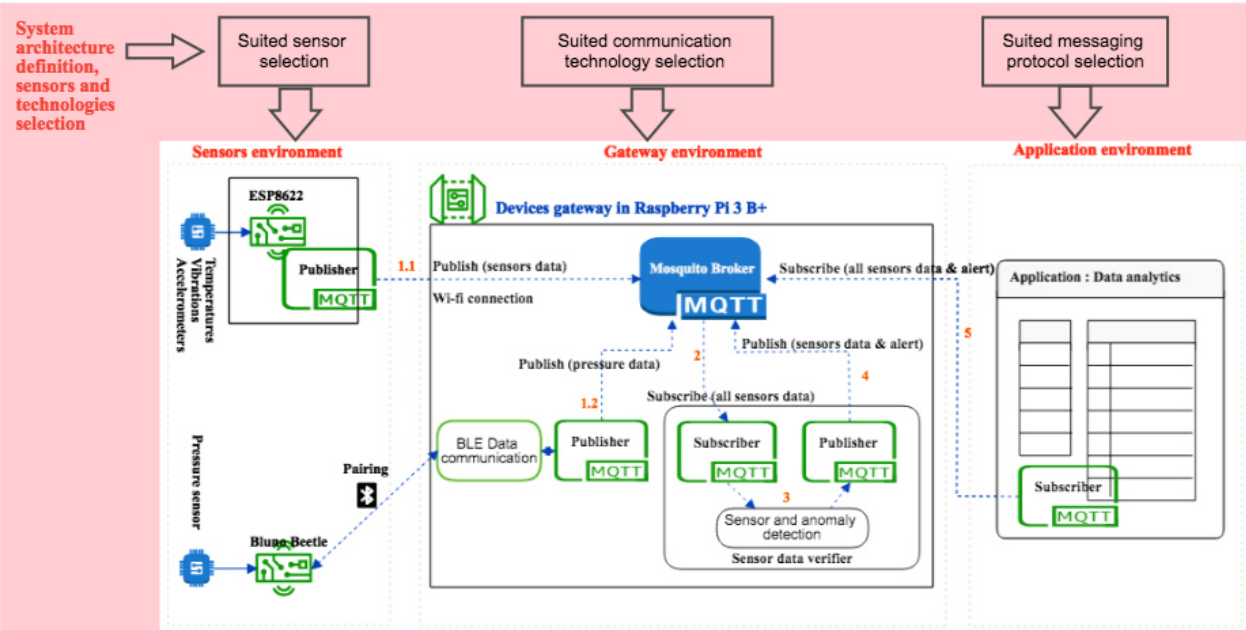

Figure 4. Translation from the generic building blocks to the SPICY demonstration architecture.

Sensors domain: In the proposed architecture, the MQTT[1] messaging protocol is used as a means for data transmission between the sensors and the gateway. However, other messaging protocols can also be used, depending on the use case, requirements and use of communication technology.

The sensors are divided into two groups: (1) sensors that connect to a micro-controller with built-in WIFI technology, and (2) sensors that connect to a micro-controller with built-in Bluetooth (BLE) technology. These two groups connect to the gateway and MQTT broker differently. The micro-controller (ESP8622) that has built-in WiFi, connects and publishes data directly to the MQTT broker installed at the gateway. This is because ESP8622 can run an MQTT client library. The Bluetooth Bluno beetle cannot publish data directly to the MQTT broker. The Bluno beetle sends instead data to a module installed on the gateway and then uses the MQTT client library installed on the gateway to publish data to the MQTT broker.

---

[1] http://mqtt.org/

Gateway domain: The gateway module consists in the demonstrator setup of the following components:

1. A broker - responsible for managing data sent from the sensors. We use the Mosquitto MQTT broker. The MQTT broker manages messages communicated between different components in the system, for instance, between the gateway and the application module or between the gateway and the sensors.
2. A sensor data verifier - responsible for validating the sensor data and detecting any anomaly that may happen in the system, i.e. detecting erroneous sensor data or devices malfunction. For example, the ambient temperature sensor, that supposes to have the temp value of up to 80 degree C, gives 100 degree C instead; or sensor suddenly stops sending data. The sensor data verifier will make sure that only qualitative data is sent to the application for further analysis.
3. A MQTT publisher - responsible for publishing messages to the MQTT broker.
4. A MQTT subscriber - responsible for subscribing message from the MQTT broker.
5. The BLE data communication - responsible for communicating with the BLE sensor and retrieving data before publishing it to the MQTT broker.

Application domain: With in the demonstrator the application module consists of the following components, each building upon the results of the previous ones:

1. MQTT subscriber - responsible for receiving the data from the gateway environment.
2. Data combiner - responsible for combining the values from the individual sensors, as received from the gateway environment, into a consolidated format appropriate for further processing and analysis (e.g. a time series with the last known values of the different sensors as columns).
3. Data pre-processing - responsible for cleaning the combined data stream using pre-defined pre-processing features (e.g. outlier removal, data imputation).
4. Anomaly detector - responsible for analyzing the pre-processed stream of data in search for predefined anomalies that may happen in the system, such as device malfunction.
5. Data visualizer - responsible for providing a dashboard depicting the current status of the system and displaying anomalies when they are detected. In addition, this component provides information about correlated and possible redundant measurements, about measurements highly varying over time, etc.

It is worth noting that in the applied architecture, every gateway has its own broker. The application module can connect to different gateways (with different brokers) for acquiring sensor data. The idea is that we want to embed everything in the gateway (see generic architecture in Figure 2) so that users do not need to install a broker in a separate machine/hardware in the network. This is also good in case we use different messaging protocol (other than MQTT) for different gateway. Another advantage of having a separate broker is to reduce congestion in case large number of sensors is used with very high sampling rate.

The examples shows the applied strategy in the project. For each generic architecture proposal we have shown different instantiations, in different demonstrators. The idea was to show the industrial partners typical task they have to take into account. The detailed deployment of the tasks is highly application specific and we tried to motivated different solutions.

Results achieved by Hahn-Schickard

The work at Hahn-Schickard focused in WP4 on supporting the development of a generic system architecture and specialize this architecture for the Hahn-Schickard clean-room-demonstrator (see WP5) as well as designing wireless sensor node and provide other sensors that could be used as versatile technological building blocks in the demonstrators.

Camera readout for analog gauge

Many older machines already provide a lot of data, but in an analog way. To be able to read old analog displays digitally and to retrofit older machines into a system, a camera-based readout for analog gauges was designed in the project.

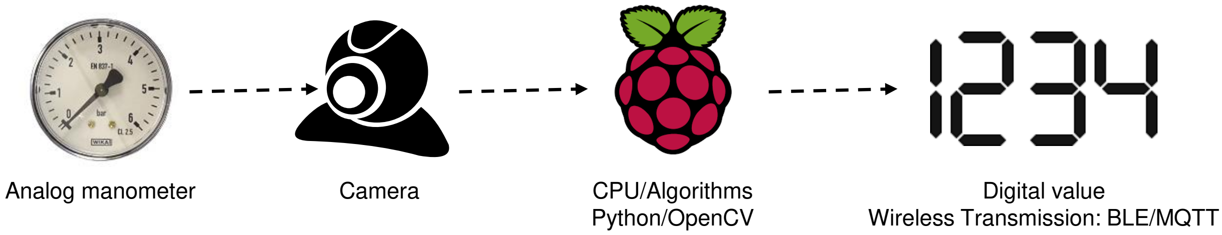

Figure 5. Schematic representation of the steps involved in reading out the analog indicators.

When evaluating the image taken with a USB webcam, various filters (blur, gray scale, sharpen) are first applied to optimize the image quality for the subsequent image processing algorithms. Afterwards, the actual instrument in the image is detected via a circle detection. Further, the individual scale lines are identified and the center and rotation point of the pointer are determined. As a final step, the current measured value is determined via the position of the pointer relative to the scale lines and digitally transmitted (wireless) to the higher-level system.

IR Temperature Sensor

In order to be able to measure the temperature on an object not only at one point, but to view an entire surface from a distance, an IR camera based system was designed and realized.

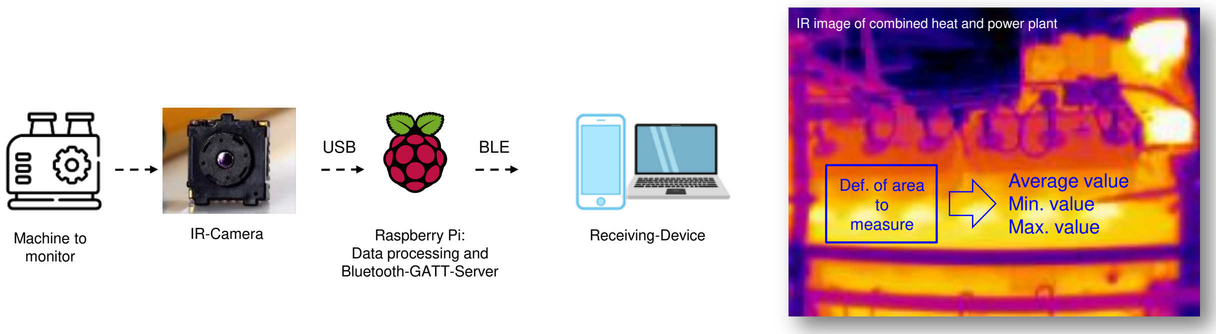

Figure 6. Schematic representation of the steps involved in IR-camera readout and transmission of the resulting discrete values to a device.

A Lepton Flir 3.5 IR camera with a resolution of 160x120 pixels, capable of capturing a temperature range from -10°C to 450°C, acts as the heart of the setup. The camera is connected via USB to a Raspberry Pi, which reads and analyzes the image data and sends the obtained pre-processed data via Bluetooth to a receiving device. It is possible to define an image range for the evaluation, in which average temperature as well as minimum and maximum value are determined. The sensor was also provided to be used in the SPICY demonstrator.

InsightProducts Sensor

The *InsightProducts Condition Monitoring sensor* was developed during this project based on the Ameli 4.0 CM sensor developed in a former BMBF-project (see section 3.5). It represents a universally applicable wireless sensor node and combines sensors for measuring acceleration/vibration/structure-borne sound, a microphone for measuring noise as well as environmental sensors for measuring temperature, humidity, air pressure and light intensity. The sensor is powered by battery or with cable. Various modifications were made to eliminate the sensor's teething problems as well as problems with long-term stability. The previous wireless processor (TI CC2650) was replaced by the Cypress CYBLE-416045-02 to increase the stability of the Bluetooth data transfer, as there were problems with the old setup to keep the data connection stable for many days. The ST IIS3DWB was introduced as the new low-noise accelerometer with ultra-high bandwidth. Optionally, the use of the previous accelerometer Bosch SMB470, which is unfortunately not commercially available as a single device to the public, was made possible.

Figure 7. InsightProducts Condition Monitoring Sensor with 3D-printed housing and sensor PCB (all images). Sensor mounted by means of a mounting plate on a vacuum pump (top right). The PCB has been manufactured and assembled by the User Committee Member Binder Elektronik GmbH.

For debugging the old Ameli-Sensor and isolate the problems, developing the new sensor hardware and the corresponding firmware as well as constructing and manufacturing of the 3D-printed housing and the robust aluminum ground plate, extensive work has been necessary.

A stiff connection of the acceleration sensors to the aluminum ground plate is very important to record structure-born sound properly. For this reason, a tray was provided in the aluminum ground plate so that the acceleration sensors could be potted.

Embedded pre-processing

Besides the wireless processor dedicated to manage the wireless data communication, the InsightProducts sensor includes a more powerful ST ARM Cortex M4 processor for pre-processing the recorded time-based sensor data already on the sensor. This data reduction is important, as not all time signals can be transmitted via the wireless interface in real time
For the InsightProducts sensor we implemented a pre-processing of data based on a random forest algorithm. The random forest is taught with features derived from the time signal of the accelerometer. The features are also calculated for different frequency ranges.
After the sensor has been trained with data recorded in the "good" or "bad" state of the machine, the algorithm can subsequently decide independently in which state the machine currently is (also the states in between). This makes it possible to detect in advance when the machine is approaching the "bad" state. Timely maintenance can be initiated and a costly unexpected standstill of the machine can be avoided.
The 3D light intensity sensor (see section 3.5) has also been equipped with pre-processing algorithms to reduce the amount of data so that it fits the transmission bandwidth of the wireless connection. The time signal of the single light sensor element is not transferred directly to the higher-level system. Instead, the minimum, maximum and average sensor value is calculated within a predefined time interval and these resulting features are transmitted. Furthermore, the Fast Fourier Transform (FFT) is calculated for each time interval and the frequency of the highest intensity peak is sent. Thus, the frequency with which the light source fluctuates can be determined.

Other sensors

Some other wireless sensors from previous projects served as building blocks for the cleanroom demonstrator (see section 3.5).

Results achieved by FZI

The FZI supported the general architecture, especially with the dedicated view to support an explicit semantic annotation and enrichment of the measurement data. The extended, instantiated architecture is shown in Figure 8. From an architectural view, the whole setup conforms basically to the architectural design (Sensor – Gateway – Application) as described in the general work. A difference in this setup is that the MQTT broker does not reside on the gateway itself but on a dedicated server instead. This flexibility was taken into account during the design of the general architecture, and the actual selection is carried out during instantiation and deployment. The *RDF Produce & Consume* components include another server, which hosts an application acting as MQTT subscriber and publisher in conjunction with transforming measurement data to conform to RDF while at the same time enriching the measurement data with explicit semantics. Part of this server is also a knowledge base (KB) respectively triple store with a SPARQL interface in which generated RDF triples are stored. The KB offers the possibility to perform semantic queries against it for example to consider past measurements. This application will utilize reasoning to derive facts from the semantically enriched measurement data.

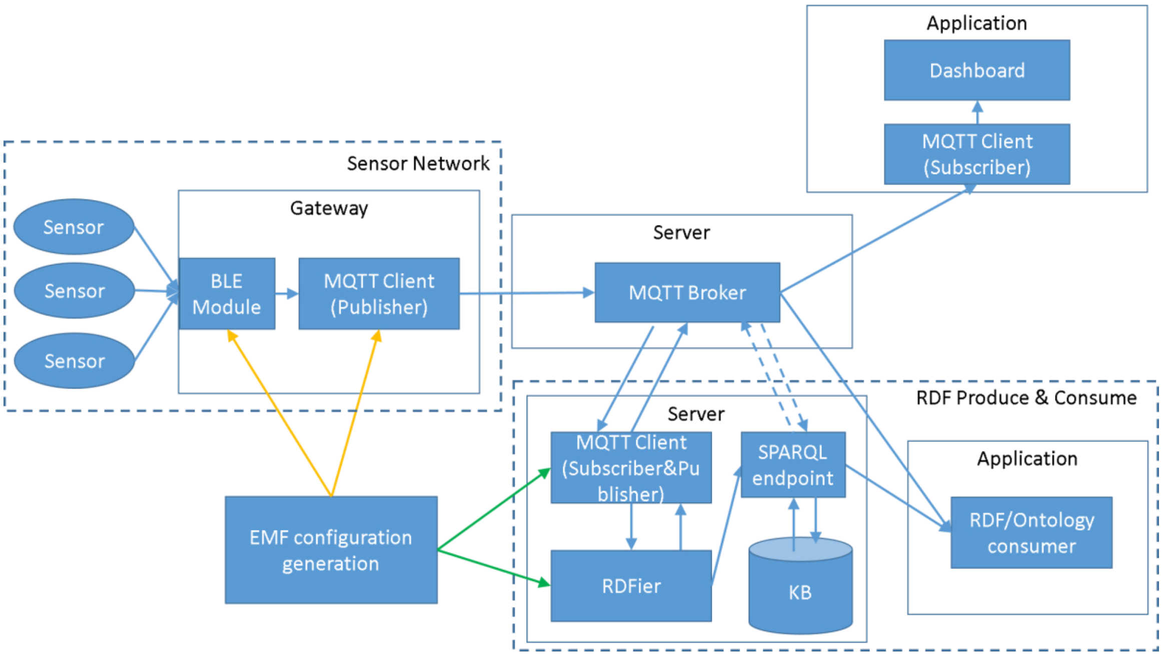

Figure 8. Semantic enrichment component architectural view.

Tooling – Within the project tooling support was created, which was meant to ease and accelerate wireless sensor network setup and deployment. The model-based approach provides an easy-to-use eclipse-based wireless sensor network editor to the user (Figure 9).

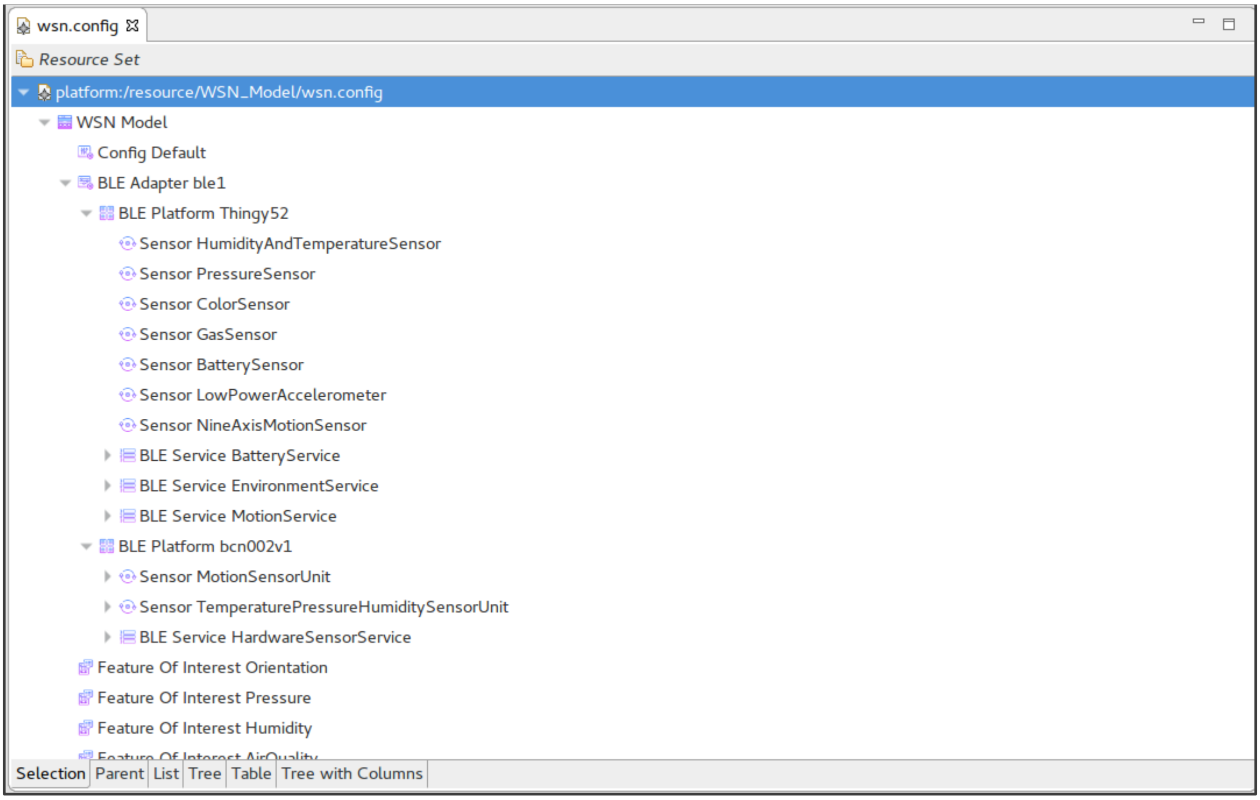

Figure 9. Wireless sensor network editor.

This editor can be used to create a model of the targeted wireless sensor network including gateway, sensor nodes and MQTT broker configuration. Such a model can be considered as a single configuration of a wireless sensor network. Thus, it is possible to create multiple configurations for the same physical hardware, which can be stored and loaded as needed.

Once the wireless sensor network model has been created using this editor, model-based code generation can be used to generate a fully functional executable for the wireless sensor network gateway(s). This gateway executable is a stand-alone application, which has only a single dependency: Java Runtime Environment (JRE). The model editor as well as the code-generation framework have been implemented using the Eclipse Modeling Framework[2] (EMF) and the Epsilon Generation Language[3] (EGL).

The generated gateway code considers communication with an MQTT broker and also the semantic enrichment of the sensor measurement data. This was achieved by integrating a specialized domain specific language (DSL) with the wireless sensor network editor. This DSL has the purpose of simplifying the process of raw sensor measurement data interpretation and was implemented using Xtext[4]. Figure 10 shows an example of a raw data interpretation specification.

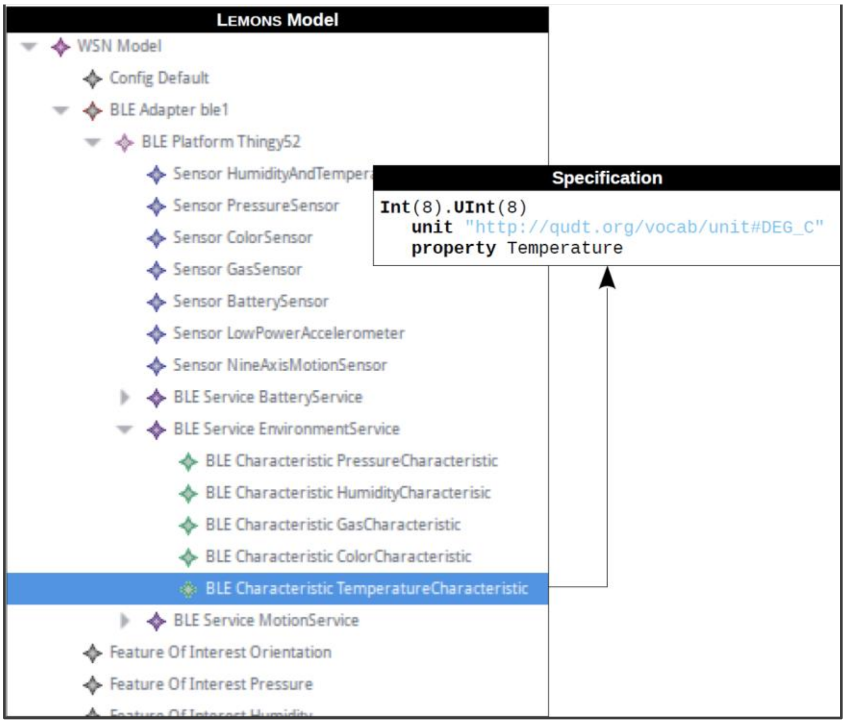

Figure 10. Language for raw sensor measurement data interpretation

In this example, the sensor data, which is expected to be represented by an array of bytes, carries the measured numeric temperature value constituted by two bytes. The first byte will be interpreted as an *Integer* while the second byte will be interpreted as an *unsigned Integer*, together representing the temperature value where the former is the integer part and the latter the fractional part of the numeric temperature value. The additional *unit* and *property*

[2] See https://www.eclipse.org/modeling/emf/

[3] See https://www.eclipse.org/epsilon/doc/egl/

[4] See https://www.eclipse.org/Xtext/

expressions are used to state information about the unit of the sensor measurement value respectively its kind. This information will then be part of the semantically enriched sensor measurement RDF data and is also used to relate sensor measurement types to corresponding MQTT topics (In this case to relate the BLE *TemperatureCharacteristic* to the *Temperature* topic).

The semantic enrichment component producing RDF data respectively micro-ontologies can also be considered as part of the prototype toolchain. The generation of machine-readable descriptions of measurements equipped with explicit semantics as a post-processing step leads to measurement data, which can easily be interpreted. Raw measurement data storage for example in a CSV file typically contains only the raw numeric measurement values usually along with a timestamp. In contrast to that, we employ micro-ontologies utilizing OWL[5] to express and describe measurement data and its relevant context, see Figure 11 for an example. This example depicts a single temperature sensor measurement, which is called *observation*.

```
@prefix qudt:  <http://qudt.org/1.1/schema/qudt#> .

@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix owl:   <http://www.w3.org/2002/07/owl#> .

@prefix xsd:   <http://www.w3.org/2001/XMLSchema/> .

@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .

@prefix fziWsn: <http://sim-narsil.fzi.de:8888/wsn/> .

@prefix fziWsnObs: <http://sim-narsil.fzi.de:8888/wsn/observation/> .

@prefix qudt-unit: <http://qudt.org/vocab/unit#> .

@prefix sosa:  <http://www.w3.org/ns/sosa/> .



fziWsnObs:  a        owl:Ontology ;

        owl:imports  fziWsn: .
```

Figure 11. Single temperature sensor measurement expressed as OWL micro-ontology in Turtle syntax.

As shown in the example, the OWL micro-ontology contains not only the raw numeric temperature sensor measurement value and a timestamp, but also other references to for example another ontological description of the sensor, which carried out the measurement and an ontological description of the feature of interest. Also important is the fact that the *result* of each observation does not only consist of the raw numeric value but also of an explicitly defined unit. To rely on (quasi-)standards with regard to the ontology vocabulary each micro-ontology utilizes the Semantic Sensor Network[6] (SSN) ontology to express facts about the sensor

[5] See https://www.w3.org/OWL/

[6] See https://www.w3.org/TR/vocab-ssn/

network itself as well as the QUDT[7] ontology to refer to explicit unit and data type definitions. Reusing these vocabularies fosters general interoperability.

In addition to the micro-ontologies, which are *dynamically* generated for each observation of a sensor, the aforementioned model-based code generation framework also generates a *static* ontology, which contains descriptions of sensors, platforms and deployments as well as other contextual information. Each micro-ontology then also contains references to this static ontology. Dividing ontological descriptions into a static ontology and dynamic micro-ontologies has the advantage that the total size of each micro-ontology is greatly reduced because not all information needs to be transmitted. The wireless sensor network configuration is unlikely to change often, which is why it can be considered static in opposite to the sensor measurements (observations). Of course, it is also possible to change the static ontology afterward, which be might necessary for example when an additional sensor is added to the wireless sensor network or if any other sensor network configuration needs to be changed. In that case, the wireless sensor network editor as described before, can be used to edit the wireless sensor network configuration. Once the configuration has been updated, the gateway executable as well as the corresponding static ontology can simply be regenerated in a single run of the generation framework, thus potentially saving a lot of time even in the operating phase of the wireless sensor network life cycle.

The semantic enrichment component has been implemented using the Apache Jena[8] framework for the construction of ontology models and the serialization thereof. Apart from transforming MQTT messages having plain numeric sensor measurement values as payload to micro-ontologies which are then published via the MQTT broker, the semantic enrichment component is also capable of connecting to any standard SPARQL[9] endpoint. Supporting the SPARQL UPDATE[10] language, the application can store generated micro-ontologies in a triplestore. We used the Apache Jena Fuseki[11] SPARQL server as a triplestore.

Depending on the frequency at which observations are made by the wireless sensor network and the corresponding micro-ontologies generation we found that the triplestore storage processing time can quickly become a bottleneck. Especially at relatively high frequencies such as 30 Hz[12] as in the case of the FZI prototype (see WP5) storage processing time of the triplestore is problematic. This is due to the design of the triplestore, which is typically a graph database. Such databases usually use highly specialized data structures, e.g. a B+ tree in the case of Apache Jena Fuseki. Inserting data into such a data structure typically is a costly operation, which has a high computational complexity. Nevertheless, we found that this problem can effectively be mitigated. Instead of updating the SPARQL endpoint each time when an observation micro-ontology is generated, multiple micro-ontologies can be grouped and then merged into a single ontology, which is then stored in the triplestore in a single operation. At the frequency of 30 Hz in the case of the FZI prototype for example we had good results with a

---

[7] See http://www.qudt.org/

[8] See https://jena.apache.org/

[9] See https://www.w3.org/TR/rdf-sparql-query/

[10] See https://www.w3.org/TR/sparql11-update/

[11] See https://jena.apache.org/documentation/fuseki2/

[12] Consider that this frequency applies to the overall system. Thus it can affect not only a single but multiple characteristics: Accelerometer measurement values and also of the gyroscope. If more characteristics are used, there will also be more data to transmit. In the FZI prototype characteristics can operate at different frequencies.

group size of 100 ontologies. Of course, at lower frequencies this method introduces a delay between observation micro-ontology generation and its storage in the triplestore. Thus, it is necessary either to adapt the group size according to the sensor system frequency or to omit it completely in case the frequency is low enough for the triplestore to maintain a good performance while updates are applied.

With regard to the industrial steering committee requirements, a dashboard, instantiated in the application domain was created. The dashboard visualizes the sensor data. The application is realized using HTML, CSS and JavaScript and is therefore able to run fully in a user's browser. The web application connects to a MQTT broker and automatically subscribes to predefined topics. These MQTT topics correspond to specific sensor measurement streams. For demonstration purpose, we implemented charts (compare Figure 12) for the ambient temperature, air pressure, humidity and CO²/TVOC (Total Volatile Organic Compounds) values. In addition to that, we visualize the ambient light color and the altitude via WebGL and plot the measured vibration, the gyroscope values and the battery level of the sensor device.

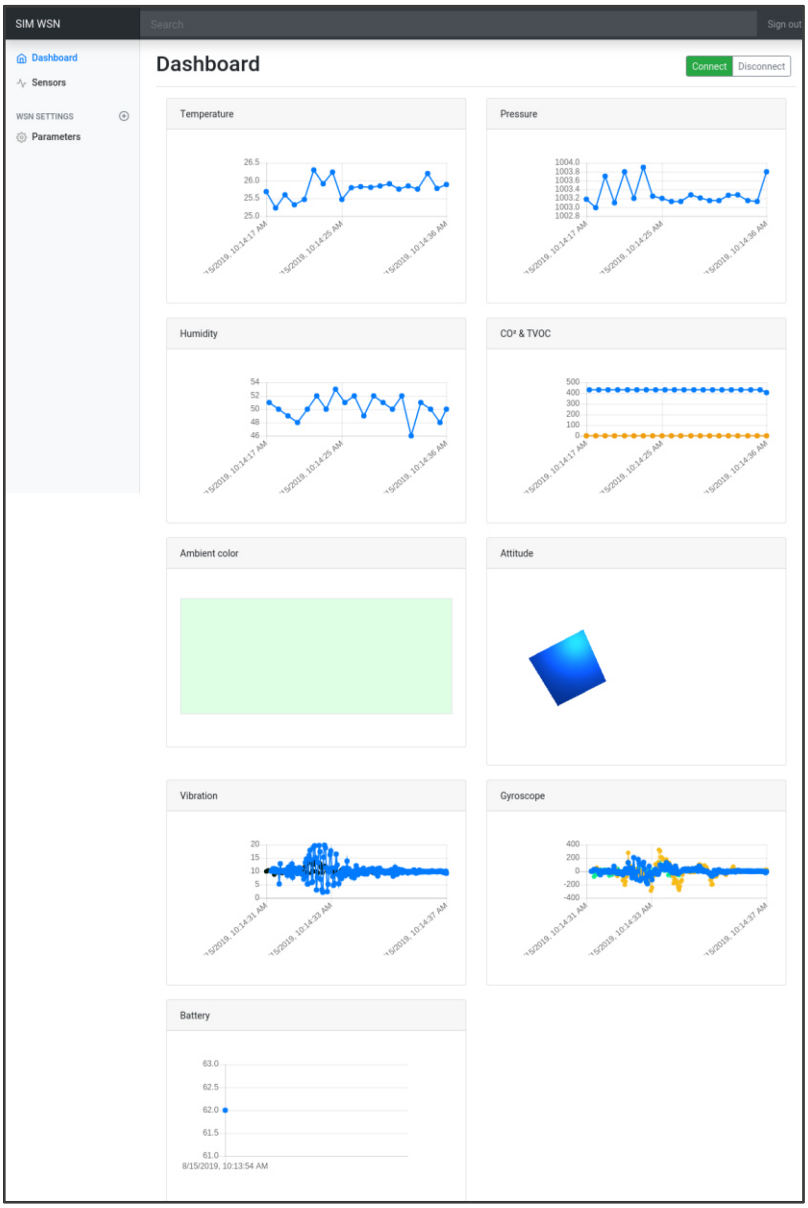

Figure 12. Data visualization dashboard.

All the demonstrator or more precise use case specific architecture instantiations were used to derive the general architecture, and sketch the available technology building blocks, that where combined in a kind of library/guideline to support the industrial partners.

3.5 WP5: Demonstrator developments

Work performed

In this work package, different demonstrators were realized, addressing the requirements of the industrial partners. Multiple demonstrators were implemented within the project, each with a dedicated focus, orientated at the needs of a member of the industrial steering committee. A number of five demonstrators were realized in direct cooperation with companies from the steering committee. Due to confidentiality reasons, we cannot discuss them here in more detail.

The SPICY demonstrator was developed in cooperation with all research partners under the leadership of Sirris and should summarise the findings of the project.

Results achieved by Sirris

The idea behind the SPICY setup, a steam engine system demonstrator, is to monitor and detect anomalies during system operation. It allows users to not only monitor remotely the state of different system components, but also alert users in case of any anomalies caused by component failure. The main component of the SPICY system is the electric water boiler tank, which generates high-pressure steam that converts into force to power other system components such as dynamos for generating electricity to power a lamp, a punching machine, a grinding machine, a carousel, a virunator (man punching the corona virus). In Figure 13 we have attached two transmissions to the steam engine in order to create several different measuring scenarios and also use the power of the steam engine as optimal as possible. The following connections were realized:

1. One dynamo connected directly to the steam engine so as to receive maximum power (Figure 15).
2. Corner transmission with two dynamos on each side (Figure 14). Because of the corner connection of the transmission and the resulting friction, we lose quite some power on the 3rd dynamo. This way we can measure how much power is lost between the 3 dynamos comparing the (i) direct connection, (ii) the intermediate transmission connection and (iii) the added friction on the intermediate transmission connection.
3. Straight transmission driving the hammer mechanism (pattern matching) (Figure 17), milling machine (Figure 16), virunator (Figure 18) and carousel (vertical/horizontal movement of planes) (Figure 19).

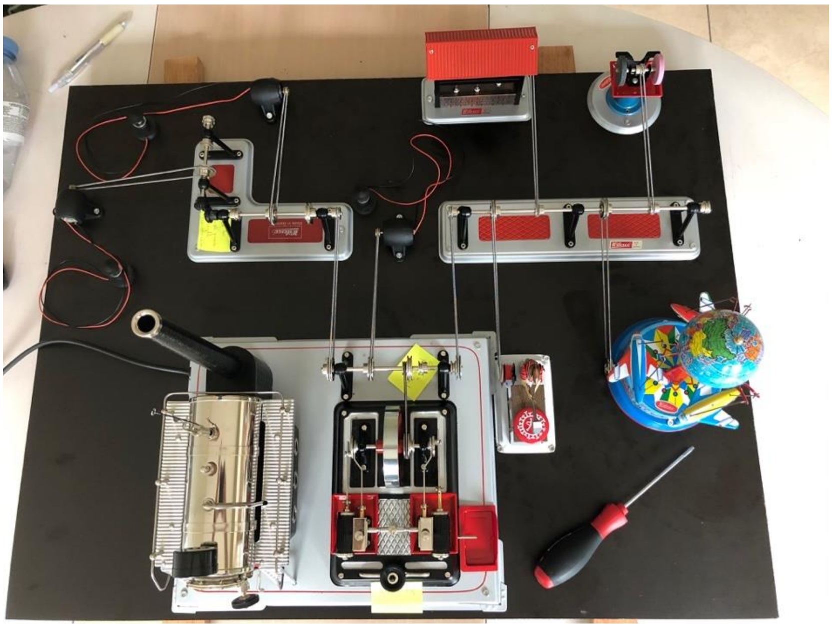

Figure 13. SPICY setup including the steam engine powering two transmissions with system components.

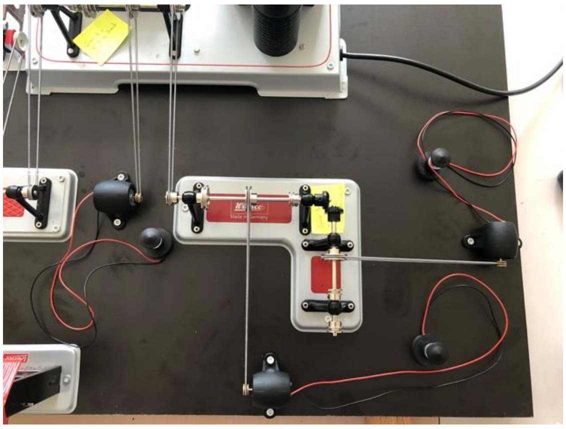

Figure 14. Corner transmission with 3 dynamo's.

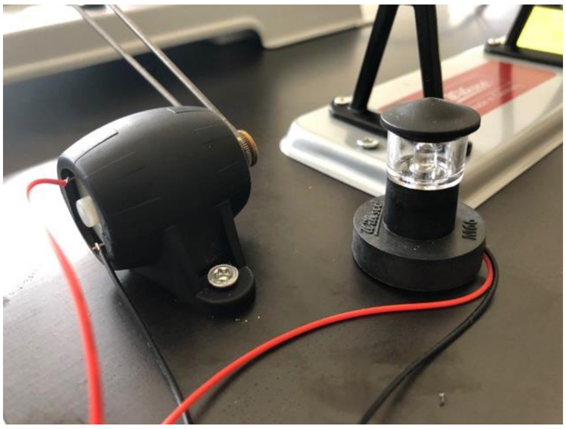

Figure 15. Single dynamo + lamp.

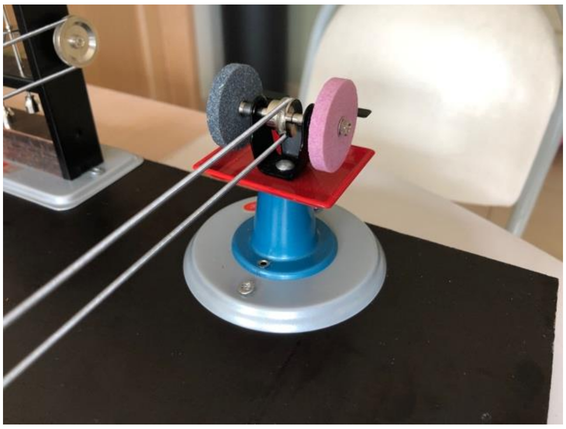

Figure 16. Milling machine.

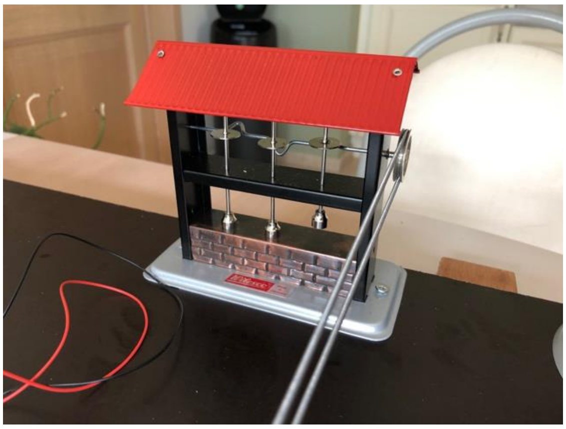

Figure 17. Hammer mechanism.

Figure 18. Virunator in our fight against COVID19.

Figure 19. Carousel.

Each SPICY component is attached with sensor(s) to monitor its state. From the measured sensor data of the components (e.g. temperature, vibration/acceleration, light, sound, movement and power consumption) in combination with data analytic tools, the current state of the components and upcoming maintenance as well as information about overall state of the system operation could be derived.

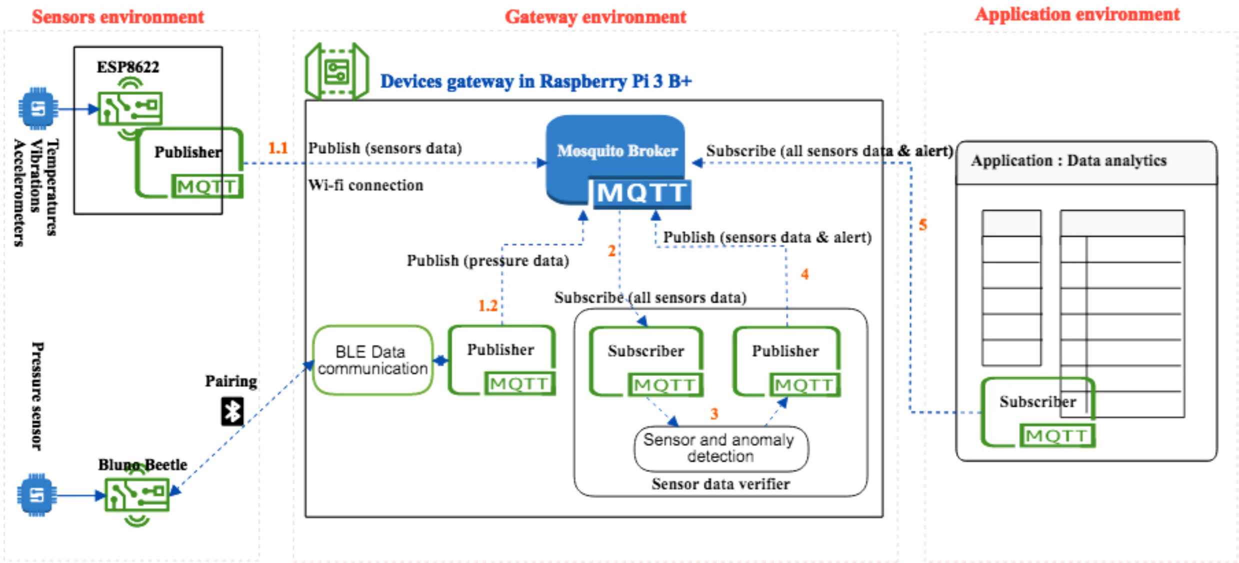

Figure 20. SPICY General architecture.

Regarding the general architecture of the system (see Figure 20), SPICY is designed following the modular approach where the entire system is divided into three independent modules: (1) sensor domain, (2) gateway domain and (3) application domain.

Concerning sensor domain, various sensors and different interface protocols are used in the SPICY setup (see Figure 20). The sensor information is sent to the gateway where it is pre-processed before forwarding to the application environment where it is further processed and stored. From the application domain, the sensor data can be accessed, analysed and displayed. The selection of an appropriate micro-controller (MC) depends on the actual requirements, such as type of sensors we plan to use and the communication technology. There are micro-controllers supporting WiFi, BLE, LoRa, Zigbee, ZWave, Sigfox and so on. The choice of sensors also affects the selection of the micro-controller since it can be connected to a sensor if and only if it has a software library allowing a micro-controller and a sensor to communicate. Due to the above-mentioned reason, it is not possible to generalize the selection of a micro-controller. It is a case-by-case study. For example, for SPICY four types of micro-controllers are used: (1) ESP8622 for WiFi communication, (2) Arduino bluno beetle for BLE communication, (3) Wipy for BLE communication and Arduino MKR for WiFi communication (see Table 1 and also Figure 21). This is because the required system setup and sensors can operate with those micro-controllers. The selected micro-controllers are selected from the previously constructed building block library.

Regarding sensor selection, it also depends on the use cases and the actual needs of the individual system setup and what environment information is needed for monitoring the system operation. Another factor that also influences the selection of the sensors is the location where they are positioned, especially in terms of inference, standardized and resilient sensors that can operate in such harsh environment are required. In the case of the SPICY setup, we use different types of sensors depending on where they are placed and what environment information we need to gather. We use both the standard and ambient sensors (e.g. ambient temperature sensor) and professional pressure sensors that operate in very harsh environments such as inside the boiler tank. For the detailed sensor selection for the SPICY setup, one can find them in Table 2.

Table 1. List of micro-controllers used in SPICY setup

Type of micro-controller	**Description**
ESP8622 feather wing	This micro-controller has built-in WiFi connection. It can also be set as a wireless access point. This MC can be connected to different types of sensors with ready-to-use libraries. In the context of the SPICY setup, we use ESP8622 feather wing for communicating ambient temperature, pressure and humidity to the application gateway. A BME280 sensor is connected to the ESP8622 feather wing through a digital pin. The MC collects sensor data and forwards those to the gateway for further processing.
ESP8622 node MCU	Node MCU is a MC that uses an ESP8622 chip and has also built-in WiFi. Similar to the feather wing, node MCU can be connected to different types of sensors with ready-to-use libraries. In the SPICY setup, node MCU is used for communicating accelerometer and gyroscope sensor data to the gateway. It is also used for the communication of the contactless infrared sensor to the gateway.
ESP8622 IoT board AI-Thinker	ESP8622 IoT board is a MC that has built-in WiFi. This MC is used in the SPICY setup for communicating the ambient temperature, pressure and humidity to the application gateway. Many other IoT application sensors can be connected to this MC with ready-to-use libraries.
Wipy 2.0	Wipy 2.0 by Pycom is a MC that has built-in WiFi and BLE. For the SPICY setup, this MC is used for communicating the ambient BMP280 temperature sensor to the gateway.
Arduino bluno beetle	Bluno beetle is an Arduino board MC that has built-in BLE. For the SPICY setup, this MC is used for communicating inside-boiler-tank pressure to the application gateway.
Arduino MKR Wifi-1010	This MC has the build-in WiFi. For the SPICY setup, this MC is used for communicating sound sensor data to gateway.

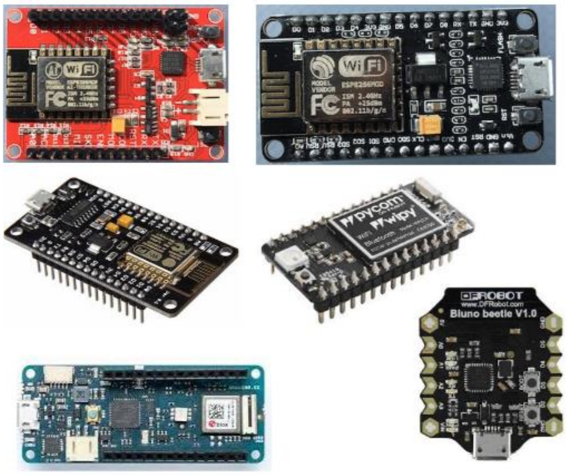

Figure 21. Micro-controllers used in SPICY demonstrator.

Table 2. List of micro-controllers and their associate sensors used in SPICY setup

Micro-controllers		**Sensors**
Types	**Firmware MC board**	**Types**
ESP8622, feather Wing Huzzah	ESP8622 feather-wing	Ambient sensors, BME280
ESP8622 node MCU	ESP8622-generic-node MCU	MPU-6050 accelerometer and gyroscope
ESP8622 node MCU	ESP8622-generic-node MCU	Infrared thermometer MLX90614
ESP8622, IoT Board, AI-thinker	ESP8622-generic	Ambient sensors, BMP280
Wipy 2.0	Wipy 2.0	Ambient sensors, BME280
Bluno beetle	Arduino	MEAS Pressure sensor
Arduino MKR WiFi-1010	Arduino	Sparkfun sound sensor

The choice of wireless communication technologies depends on the actual sensor placement, data reliability requirements and the placement of the application gateway that acts as the central hub collecting data from different sensors. In the wide area system setup, long range wireless technologies such as LoRa or Sigfox can be used while in medium range system setup, WiFi can be a choice. Zigbee, Z-Wave and Bluetooth BLE are good choice in a setup where sensors and application gateway are in a close distance.

Another key factor in wireless communication technologies selection is the data reliability requirements. For applications intolerant to data loss, we need to have a reliable wireless communication technology such as WiFi where TCP can operate while LoRa or Sigfox are suitable for data-lost tolerant applications.

For the SPICY setup, we choose two wireless communications, WiFi and BLE. This is because they are suitable for this type of system setup. The selection of a wireless communication is for portability purposes and for easy installation of sensors and establishing connection to the gateways.

Security - In the case of the SPICY setup, different security protocols are used for securing the transmission of data between different components of the system, such as between sensor(s) and gateway(s) as well as between gateway(s) and application(s). Taking into account the high security risk of the wireless sensor network, we also implemented the sensors' authentication, on top of the secure communication, that controls all sensors before onboarding the system. Each sensor is authenticated before allowing it to join the network.

The security implementation consists of the following components:

1. Authentication and authorization: At device level, passwords and credentials are used to authenticate sensors before allowing them to on board the system. Every sensor needs to authenticate itself at the MQTT broker with username and password. Only sensors with valid username and password are allowed to publish data through the broker. The same is applied for the gateway and the application modules that can subscribe to sensor data from the MQTT broker if and only if they have valid username and password. At the frontend application level, usernames and passwords are used for authenticating users in the system.

2. Storage and data processing security: All data collected from sensors are encrypted before storing at application level.
3. Communication security**:** TLS is used for securing the communication between the gateway and the frontend application, and for securing the communication between the gateway and the sensors (e.g. micro-controllers). In the context of the SPICY setup, we use different types of TLS supporting the micro-controllers we use. However, it is important to note that not all micro-controllers support the TLS protocol.

Messaging - In the SPICY demonstrator, we use a publish/subscribe protocol. This is because this protocol type is suitable for the SPICY system configuration and setting; a pub/sub protocol is more appropriate than other protocols such as COAP. Among the existing publish/subscribe protocols, we selected the MQTT protocol and in our implementation MOSQUITTO distribution is used for both the broker and the publish/subscribe client development. MQTT is used for communicating sensor data between the sensor(s) and the gateway(s) and also between the gateway(s) and the frontend application(s).

Results achieved by Hahn-Schickard

The idea behind the clean room demonstrator is to monitor machines and environmental conditions in the clean room of Hahn-Schickard (Figure 22) via various networked sensors. The sensors were partly developed in previous projects but also some commercially available sensors were used. From the measured sensor data of the machines (e.g. temperature, vibration/acceleration, light intensity, humidity, power consumption, …), we intended to obtain information about the current state of the machines and upcoming maintenance as well as information about the environmental air parameters in the clean room such as temperature, humidity, pressure and direction and strength of flow.

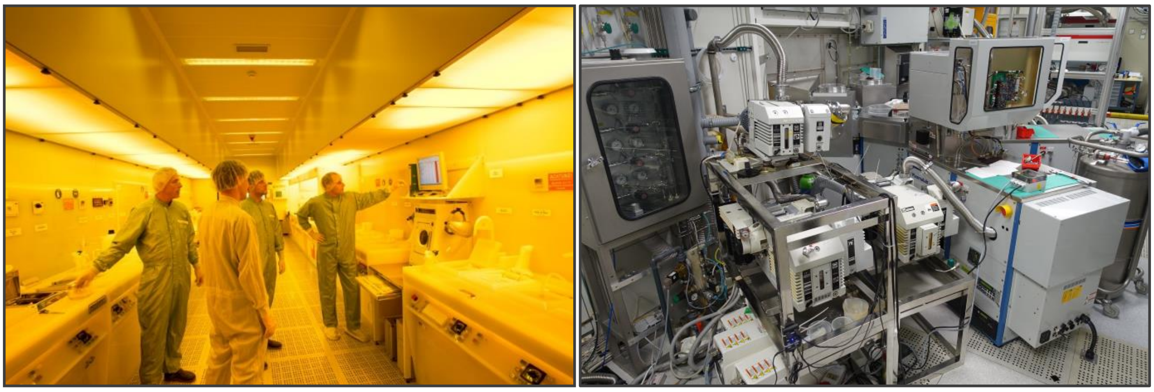

Figure 22. Clean room environment: High-quality white-room with operators and machine frontends (left), Mid-quality gray-room with machine backend (right).

Besides the sensors in the clean room, also the combined heat and power plant of Hahn-Schickard as well as our machine shop were partially equipped with sensors to monitor their proper operation. Figure 23 gives an overview about the used sensor types in the clean room demonstrator.

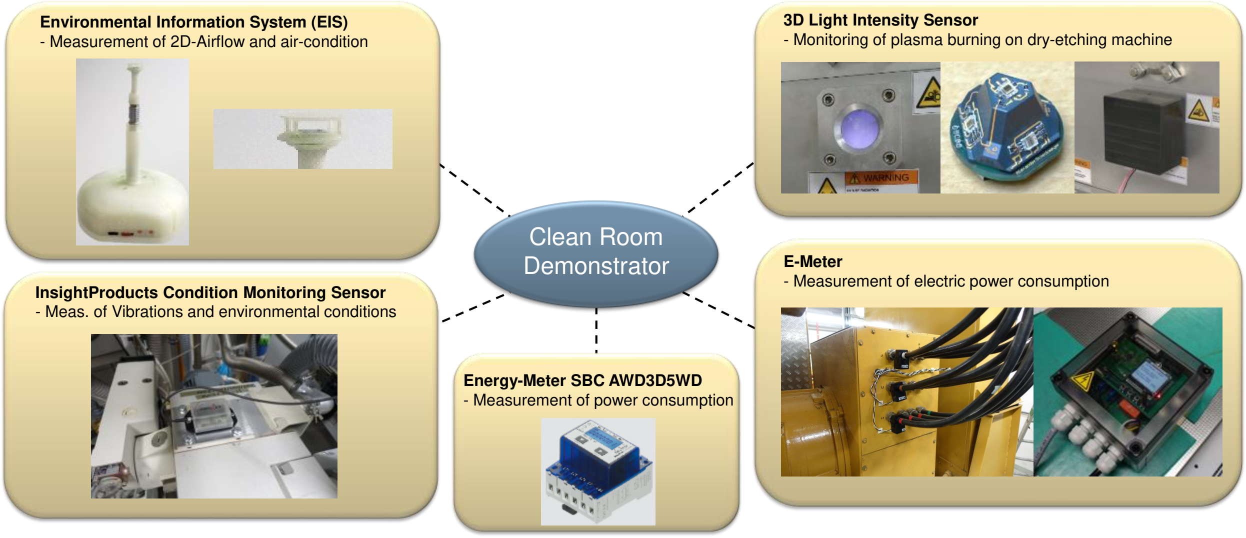

Figure 23. Overview of sensor types used in the clean room demonstrator

Regarding the general architecture of the system, the three main building blocks *Sensors environment*, *Gateway environment (sensor network + IoT Network)*, and *Application environment (Analysis)* defined for the SPICY-demonstrator can also be found in this demonstrator setup in a slightly extended variant:

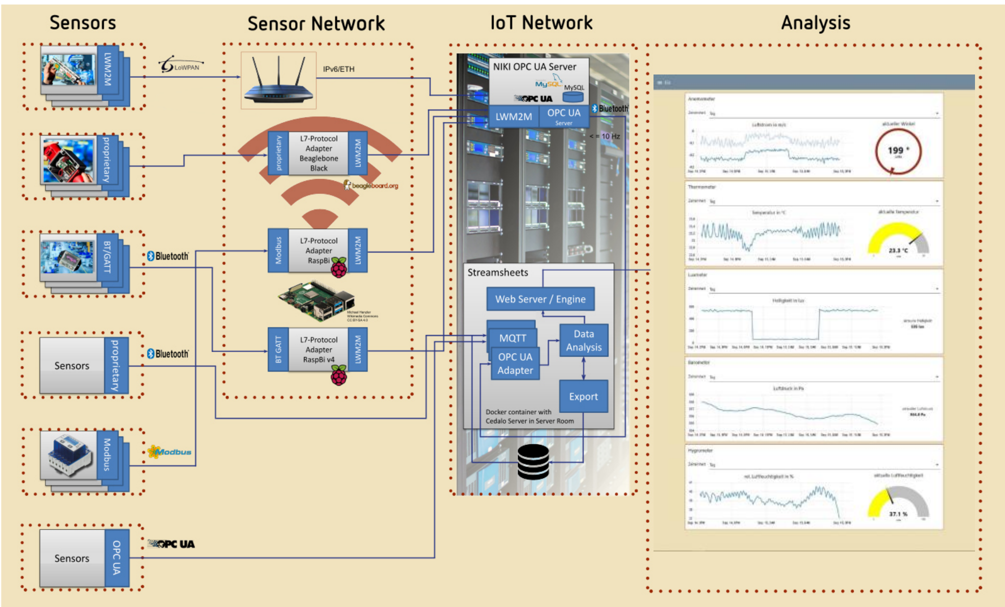

Figure 24. General architecture of the clean room demonstrator.

Regarding the sensor environment, various sensors and their different interface protocols are shown in Figure 24. The sensor information is sent to different gateways in the sensor network and the overall data is transferred to and stored on an OPC-UA-Server in the IoT Network. From the application environment, the data of the OPC-UA-Server and its corresponding database can be accessed, evaluated and visualized.

In the clean room demonstrator of Hahn-Schickard several sensor platforms from former projects and corresponding microcontrollers were deployed and partially enhanced. The table below gives a survey over the sensor platforms used.

Table 1: Overview of sensors used in the InsightProducts clean room demonstrator

Device Name	**Relevant Measurand(s)**	**Interface between Sensor and Gateway**	**Place / Application**
ESIMA E-Meter	• Electrical Power	Wireless via 6LoWPAN	• Oxidation oven in clean room • Vacuum pump in clean room • Combined heat and power plant • Milling machine in mechanical workshop
Energy Counter SBC	• Electrical Power	Cable based via Modbus	• Oven in clean room
InsightProducts Sensor as an further development of AMELI 4.0 Condition Monitoring Sensor	• Vibrations • Temperature • Humidity • Pressure • Light intensity	Wireless via Bluetooth Low Energy	• Vacuum pump in clean room • Combined heat and power plant
3D Light Sensor	• Light intensity	Wireless via Bluetooth Low Energy	• Plasma monitoring in clean room
Environmental Information System (EIS)	• Air flow • Temperature • Humidity	Wireless via 6LoWPAN	• Air monitoring in clean room (sputtering line and photolithography)

The *Ameli 4.0 Sensors* (from previous BMBF-Project Ameli 4.0) in Figure 25 were developed as a condition-monitoring sensor to measure vibration and acceleration as well as environmental parameters on machines. One wireless sensor and one cable based variant were realized.

Figure 25. Ameli 4.0 Sensors for measuring vibration/temperature/pressure/humidity/light intensity. Wireless Condition Monitoring Sensor (left) and CAN-based Process Monitoring Sensor (right)

The wireless *Condition Monitoring Sensor* measures acceleration, acoustic noise, temperature, pressure, humidity and light intensity. It is powered either by battery or by cable. Whereas the *Process Monitoring Sensor* only measures temperature and acceleration. It is supplied with power by cable via a CAN interface. Both sensors are equipped with a STM32L486 µC that allows the (pre-)evaluation of the collected data directly on the sensor. The wireless data transmission is performed via a separate µC (TI CC2650), which is capable of the wireless protocols Bluetooth and 6LoWPAN. This approach serves to distribute the processor load between two microcontrollers and to create a logical separation.

Preprocessed sensor data is transmitted to a gateway, which is based on the BeagleBone Black. The BeagleBone Black is a Linux-based one-board computer similar to the Raspberry Pi. It is equipped with an ARM Cortex A8 processor, 512MB RAM, 4GB flash storage, Micro-SD-slot, USB, Ethernet and HDMI. Beaglebone's TI AM3359 processor offers native CAN support, which allows high and stable system performance on the CAN bus. Its standard functionality was extended by a self-developed hardware interface for CAN and for Bluetooth Low Energy. Linux USB support and Linux Bluetooth Stack allows Bluetooth hardware version modifications to be more flexible. The high and stable performance of the measurements, in timing perspective, is supported by Linux OS's direct access to ARM General Purpose Input/Output (GPIOs). The Beaglebone's performance is sufficient to cover Bluetooth, CAN communication, GPIO based triggering and TCP/IP communication with the Data Storage system. The detailed documentation of the open-source hardware and operating system allows the system to be modified for industrial grade applications.

After the data has been received from the sensor, the data is dispatched on the gateway and afterwards delivered to the OPC-UA-Server via LwM2M-protocol, which provides the IoT-data for further processing to other applications.

The *InsightProducts Condition Monitoring sensor* was developed during the InsightProducts project based on the Ameli 4.0 CM sensor. For a description see section 3.4.

Two of the InsightProducts CM sensors were installed on vacuum pumps in the cleanroom. The 3D light intensity sensor is installed on a machine nearby. To read the sensor's data via Bluetooth Low Energy a gateway based on a Raspberry-Pi has been developed and mounted on a DIN-rail in a cabinet near the sensors (see Figure 26). The gateway is powered by a 5V power-supply and cooled by a radiator with two fans. For a better reception, the gateway is equipped with an external Bluetooth antenna. The received data is then transferred via LAN to the OPC-UA server.

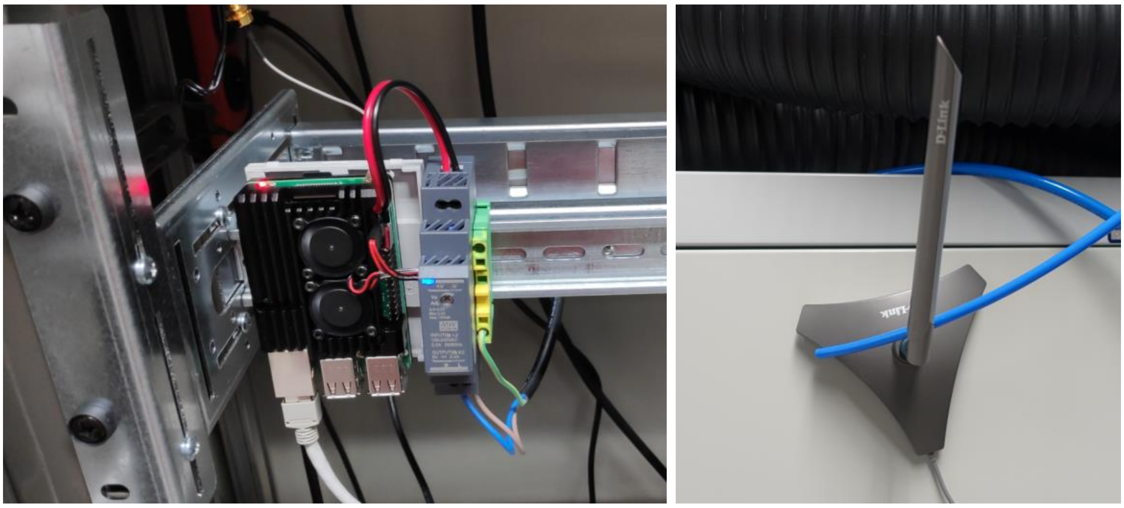

Figure 26.Left: Raspberry-Pi with radiator and power supply mounted on DIN-rail in a cabinet in the clean room. An external antenna as well as Ethernet is connected to the Raspberry-Pi. The Raspberry-Pi receives the signals of the InsightProducts sensors as well as the signal of the light sensor and forwards the data to the OPC-UA server. Right: External Bluetooth antenna connected to Raspberry-Pi.

The *Environmental Information System (EIS)* is a sensor, which monitors the room climate inside buildings. It consists of a 2D airflow sensor that measures the strength and direction of the flowing air. Additional air parameters such as absolute pressure, temperature, humidity are recorded by the system. In addition, the light intensity of the environment is measured. Figure 27 shows the device in its housing.

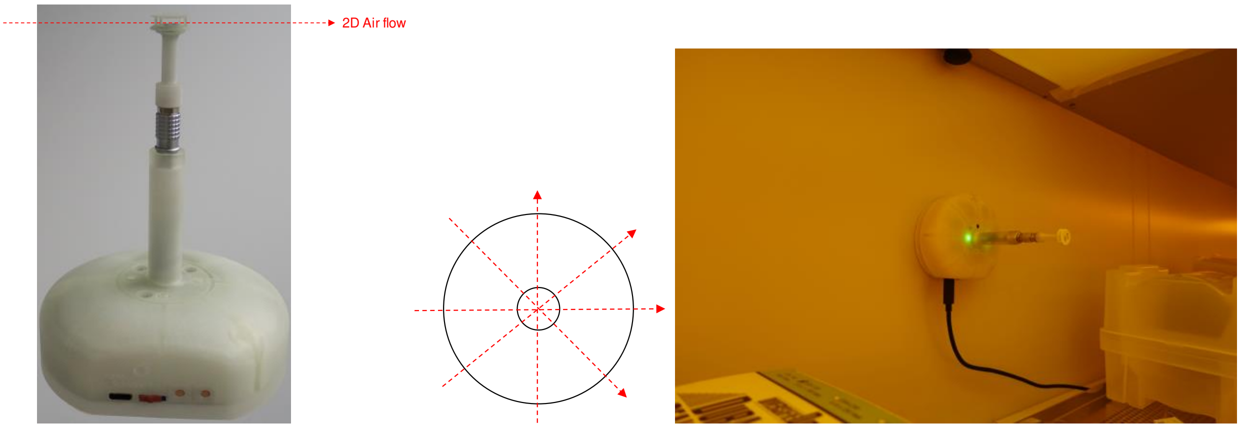

Figure 27. Left: Environmental Information System (EIS) measures airflow in two dimensions as well as temperature, pressure, humidity and light intensity. Right: EIS installation on a wall in the clean room.

The sensor is supplied with energy either by battery (chargeable over USB) or by cable with an external power supply. It offers an USB interface as well as wireless transmission via 6LoWPAN or Bluetooth Low Energy (BLE).

The *ESIMA E-Meter* (see Figure 28) is a sensor to measure the electric parameters current, voltage and power. It measures AC-currents with 1 to 3 phases at 50Hz contactless (galvanic isolation) via current transformers. The measurement range for currents is from 10A to 1000A, depending on the used current transformer. In contrast, the voltage measurement is contact based (galvanic connection) for 1 to 3 phases at 50Hz. The system is powered either via a rechargeable battery (charging via USB) or via an external power supply. The data can be read-out cable based over a serial connection (RS-232) or wireless via 6LoWPAN or Bluetooth Low Energy.

The microcontroller basis of the ESIMA E-Meter is a Cypress PSoC® 5 ARM3 processor, which is shipped with various analog and digital blocks like operational amplifiers and A/D-converters. This makes it especially suitable for connecting analog sensors. As gateway, either a Raspberry Pi or a BeagleBone based platform is used.

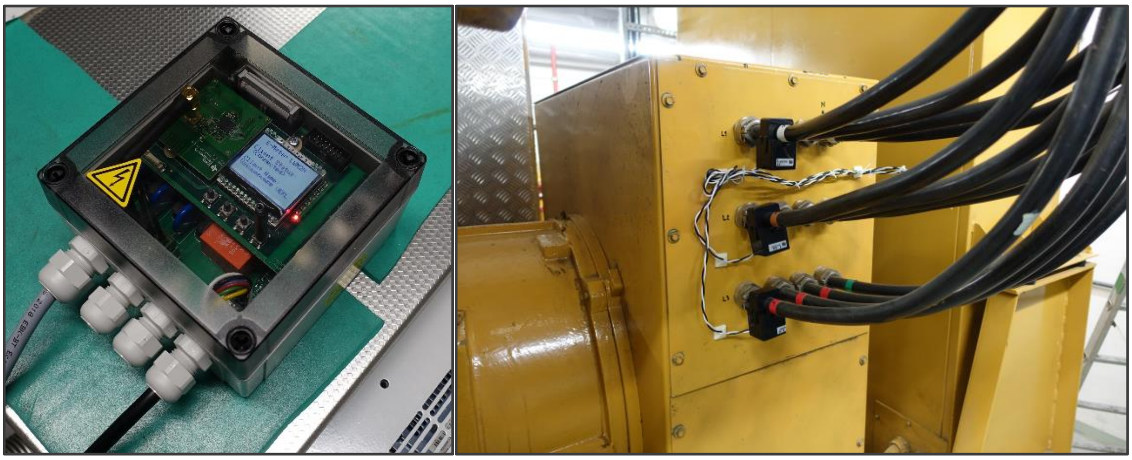

Figure 28. ESIMA E-Meter: Measurement of Current, Voltage and Power (left), E-Meter current transformers at power output of combined heat and power plant.

SBC Energy Meter

As a commercially available sensor we additionally use the Saia Burgess Controls (SBC) AWD3D5WD energy meter (Figure 29). The sensor measures 3-phase AC voltage and is used to read the meter reading, current, voltage and power. The voltage is measured galvanic connected while the current measurement is done galvanic isolated via an external current transformer. As cable based sensor interface serial Modbus is available.

Figure 29. Saia Burgess Controls AWD3D5WD energy meter[13].

The *Hahn-Schickard 3D light intensity sensor* (Figure 30) measures light intensity (VIS and IR) in different directions of space. This was achieved by applying four light sensors to a specially shaped MID substrate. The light sensor element is commercially available, while the substrate and electronics were developed at Hahn-Schickard. As µC a Cypress PSoC®4 is used, which offers a Bluetooth Low Energy interface. Power is supplied via a button cell or via an external power supply.

[13] Image Source: https://www.saia-pcd.com/en-gb/products/electrical-energy-meters/bus-capable-energy-meters/three-phase-bus-capable-energy-meters

In the project, the sensor is used to monitor the plasma burning in a plasma chamber of a dry etching machine.

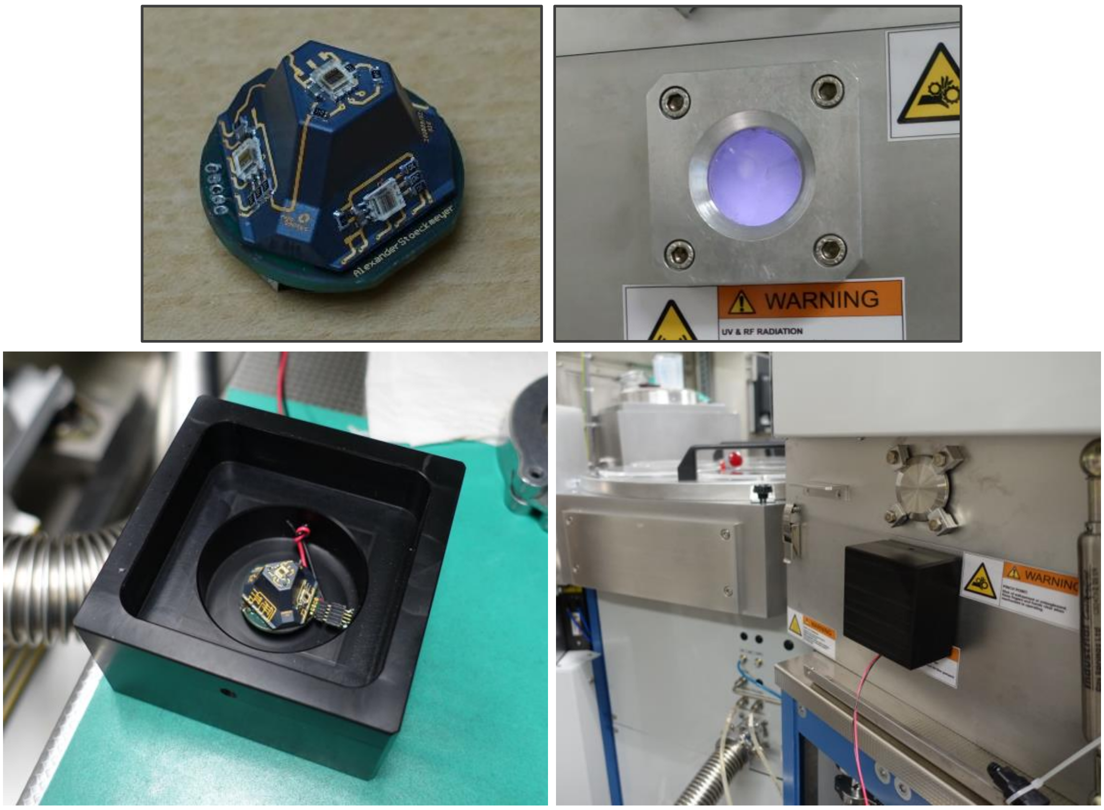

Figure 30. Hahn-Schickard 3D light intensity sensor in MID-technology (top left), window to plasma chamber of dry etching-machine (top right), mounted sensor on dry etch machine with black housing (bottom).

Interfaces

Insight Products Sensors: For the Condition Monitoring Sensor Bluetooth Low-Energy (BLE) was chosen for wireless transmission of the data within approximately 20m. The primary reason to choose BLE is the low power consumption, which is considered essential for a battery-powered sensor. In general, the Cypress CYBLE-416045-02 allows also the use of other wireless protocols. The 3D-printed plastic housing of the sensor easily allows transmitting wireless data to the gateway. For the use of the original Ameli 4.0 Condition Monitoring Sensor, the use of 6LoWPAN was initially planned, but was discarded after the presence of some incompatibilities with the mobile processor.

The Ameli 4.0 Process Monitoring Sensor was designed for use in harsh environments. Therefore, it comes with a solid and hermetic metal housing that would make the transmission of wireless data difficult. For that reason, the sensor was equipped with a cable-based CAN-interface to provide a robust data connection and power supply.

For the other sensors, similar considerations were made. The wireless interface was implemented for easy installation of a device and establishing a connection to a gateway within a range of about 20 meters. The disadvantage of wireless connections is that they are easily disturbed by other devices or too many devices sharing the same frequency range. This depends very much on the location where the system is to work. A cable-based solution and interface offers in most cases a more reliable connection. The disadvantage is the need of the cable and more effort for installation and maintenance.

Security

In the sensor platforms different security protocols are used for the secure transmission of data between sensor and gateway as well as between gateway and application.

For the 6LowPAN communication protocol the proprietary stack of the former project partner Stackforce GmbH is used and runs as a closed source system on both sensor and gateway. The encryption of the data is possible.

For the Bluetooth connections in our demonstrator, we used no security features, as it was just a test system after the replacement of 6LoWPAN. The Bluetooth Standard in general comes with an AES encryption, which could be used to secure the system's data transmission.

In the communication between gateway and server, the TCP/IP network protocol is used. LAN and Wi-Fi communication can be extended by additional participants, which increases the possibility to compromise the system overall wired connection length. For this reason, the use of encryption and authentication is highly recommended and can be achieved by the use of ssl/tls.

LwM2M is based on UDP, thus communication can be encrypted by using Datagram Transport Layer Security (DTLS), which is also suitable for streaming applications. After the communication protocol operates a handshake and establishes trust based on X.509 PKI Infrastructure, CoAP messages are encrypted as DTLS application data, allowing end-to-end system encryption.

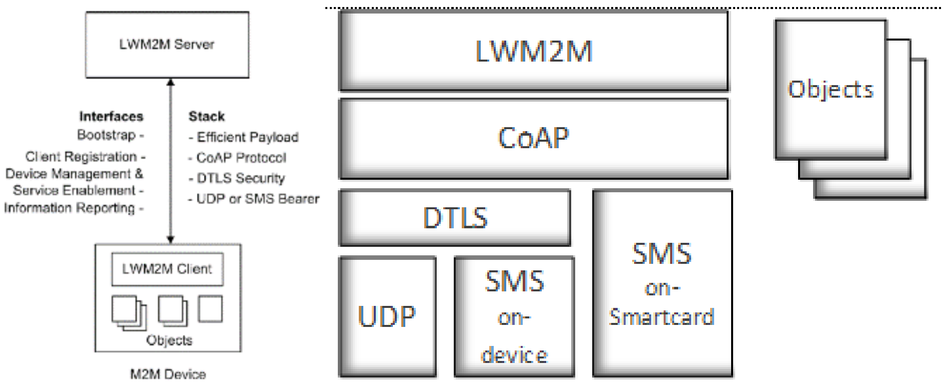

Figure 31. LwM2M general architecture and protocol stack [connect2.io]

The data sent over the CAN bus in our Ameli 4.0 demonstrator is currently not encrypted, but the encryption of the data on the sensor and gateway before/after transmission independent from the CAN-protocol would be feasible.

Messaging

The clean room demonstrator implements multiple transmission protocols and messaging technologies. A survey of the used protocols and the corresponding devices is given in Figure 32.

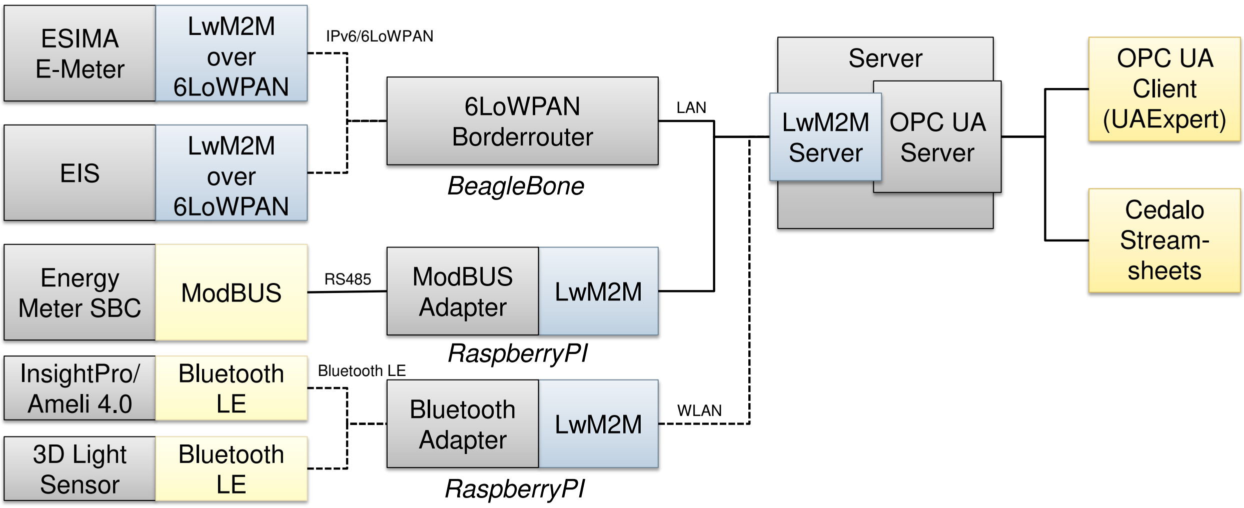

Figure 32. Survey of used transmission technologies for the clean room demonstrator.

The data of the InsightProducts Condition Monitoring Sensor and the Process Monitoring Sensor is transmitted via the *Lightweight Machine-to-Machine (LwM2M)* protocol. It is an object-based transmission protocol that comes with a well-defined data and communication model and predefined ready-to-use objects (OMNA, IPSO, GSMA). It transmits for example besides the pure sensor value, also additional information such as the name of the sensor and the unit of the sensor value (description of objects). It is a CoAP-based protocol that has been especially intended and designed for device management in the IoT. Furthermore, it offers the possibility of connectivity monitoring, remote device actions and structured firmware over-the-air updates.

Open Platform Communications - Unified Architecture (OPC-UA) is a cross-platform and manufacturer independent standard for data exchange. It has a client-server-based communication model, which we use for the backend communication. The OPC-UA server can be accessed by an OPC-UA client like UAExpert, Stream-sheets or Node-RED to access single data values or visualize the data as a plot. The data models of LwM2M is mapped to OPC-UA's data model. OPC-UA offers platform independent secure communication and supports also asynchronous messaging. OPC-UA includes an information model as well as limited semantics.

For the connection of the sensors to their gateway/adapter BLE and 6LoWPAN is used for wireless data transmission. For the only wire-based sensor, RS485 is used as simple serial protocol.

Results achieved by FZI

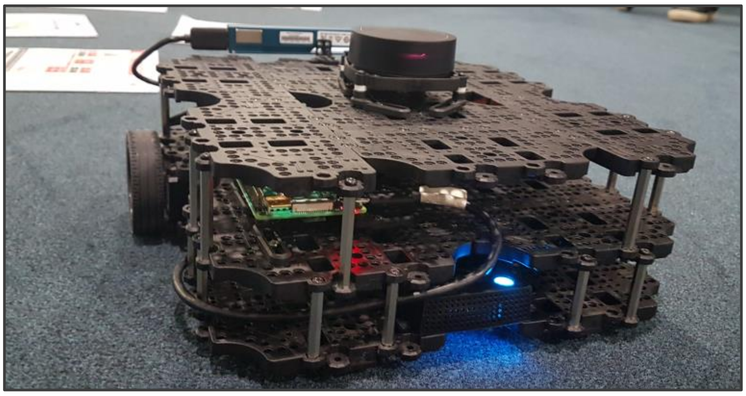

Figure 33. Mobile robot equipped with an IoT sensor platform.

As a demonstrator setup, a mobile robot was equipped with an IoT sensor platform. A major goal was to monitor the robot device's condition. Using measurement data, we detected when the robot hits an obstacle or a rough surface. For this purpose, the IoT sensor platform *Nordic Thingy:52* was attached to the lower back side of a *TurtleBot Waffle* (see Figure 33).

From an architectural point of view, the overall demonstrator setup is in alignment with the generic architecture, as described earlier. The three main building blocks *Sensors domain*, *Gateway domain*, and A*pplication* domain are also present in this demonstrator setup.

With regard to the sensors and gateway environment, the Bluetooth LE wireless sensor platform was connected to a Raspberry Pi gateway. At the next higher network layer MQTT was used as messaging protocol. A special case of the instantiation of the generic architecture is that the MQTT broker does not reside directly on the gateway but on a dedicated machine instead. The intention behind this decision is primarily to reduce overall load on the gateway. A second intention is to provide more flexibility with regard to the possibility of integrating additional gateways.

Regarding the application domain, the demonstrator setup contains a software component, which acts as an MQTT subscriber and publisher. This software, subscribed to the acceleration and rotation measurement data, publishes and transforms the raw sensor measurement data values to semantically enriched representations using micro-ontologies, which are then again published via the MQTT broker.

The Thingy:52 comes with a couple of sensors, such as for measuring the ambient air temperature, pressure, humidity, ambient light colour, etc. In this setup, we were particularly interested in vibration, which is why we used the sensor platform's accelerometer for measuring acceleration in X, Y and Z-axis. The vibration is then calculated via the norm of each acceleration vector. Moreover, we are measuring rotations of the device also in the aforementioned axis via the sensor platform's gyroscope. In order to obtain useful results in terms of detecting peaks in vibrations, the system operates at a frequency of 30 Hz.

For the mobile robot platform setup, we rely on a Raspberry Pi as gateway, see Figure 34. This is because the Raspberry Pi is a proven and well-known platform. There exists a large ecosystem for it with many applications and libraries readily available. Moreover, it has a small footprint and is a low-cost device. The goal is to place the device on or embedded it within the interior of a machine. This makes it highly suitable of IoT applications. Additionally, the

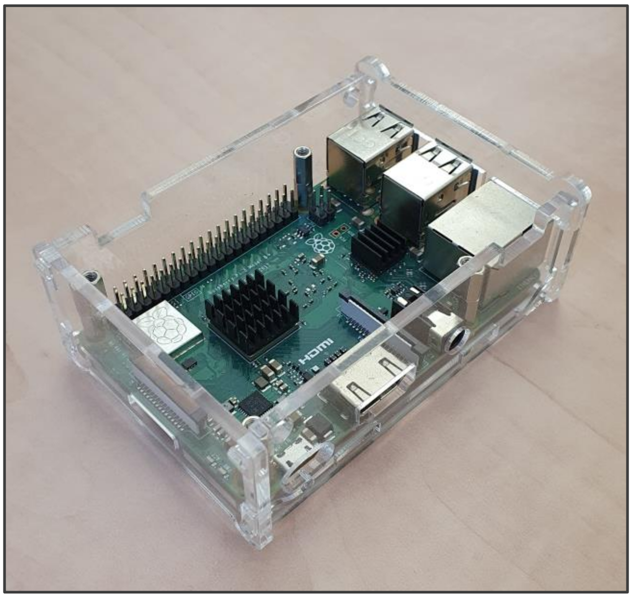

Figure 34. Raspberry Pi gateway.

Raspberry Pi has Bluetooth LE and WiFi connectivity, turning it into a great gateway.

With regard to the sensor platform, we chose the Nordic Semiconductor Thingy:52™ multi-sensor prototyping platform (Figure 35). This platform is equipped with several environment sensors. Amongst these sensors are an accelerometer and a gyroscope, which were required for the prototype setup. The Thingy:52 has Bluetooth LE connectivity and can also be extended with additional peripheral devices via GPIOs and/or the I²C bus. An important point is also, that the source code of the firmware of the Thingy:52 is available, which eases development of applications that require a custom firmware. The Thingy:52 is also a low-cost device, which in sum makes it highly suitable for IoT (prototype) applications.

Also important in the decision for a sensor platform was the possibility to mount the sensor platform on the target system, which is to be monitored, easily. We clearly favoured a non-intrusive approach to the sensor deployment over an intrusive one due to the implications of each approach.

Figure 35. Nordic Thingy:52TM multi-sensor prototyping platform.

In the course of the development of the demonstrator, we also experimented with an additional sensor platform: The STMicroelectronics STEVAL-BCN002V1 multi-sensor board (Figure 36). This board is much smaller than the Thingy:52TM and can therefore be more easily attached to any mounting point. With regard to sensors, the BCN002V1 also comes with a Time-of-Flight sensor for measuring distances within a close range. While we managed to integrate this additional sensor in our system connecting it with the gateway, a drawback of this sensor node compared to the Thingy:52TM was, that the firmware was not well documented.

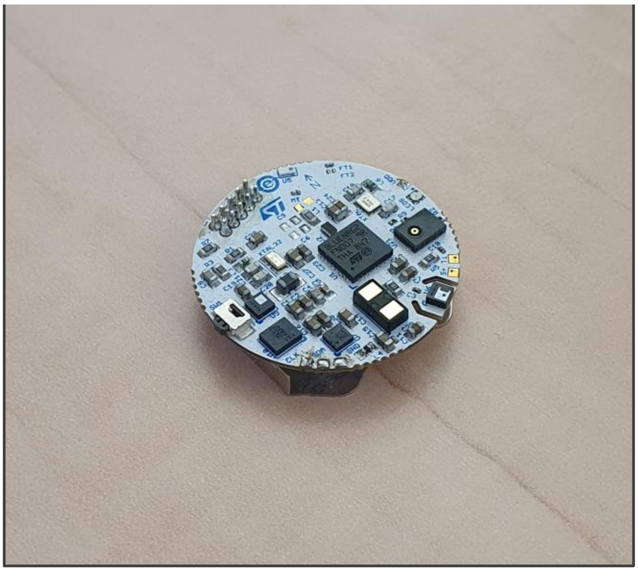

Figure 36. STMicroelectronics STEVAL-BCN002V1.

As the demonstrator consists of a mobile robot platform, it is suitable to use a wireless communication technology; the selection strategy was documented in the guidelines for sensor selection. The prototype system is meant to be operated indoors where the distance between the sensor node and the gateway is relatively low. While the Raspberry Pi gateway supports Bluetooth LE and WiFi connectivity, we have a demand for a low power solution. Thus, the close-range wireless communication technology Bluetooth LE was chosen over WiFi due to its lower power requirements. The need for a low power solution stemmed from the fact that we want to mount an additional sensor platform on the target robot platform, which should be independent from the monitored systems in terms of power and communication. This kind of isolation is on the one hand required to avoid an intrusive approach to the sensor deployment on the target robot platform and on the other hand to provide a certain degree of redundancy between the two systems with regard to system condition monitoring.

Security - Considering the wireless sensor network application and collected data, we decided to omit device level authentication and authorization compared to the generic architecture. While this might seem risky at first, it was in fact a well-considered decision, which we will explain in the following.

As already mentioned, the generic architecture proposes to use authentication and authorization at device level. Sensor nodes need to authenticate at the gateway via username and password to board the system. This means that the gateway is a *passive* component, which primarily receives sensor data and forwards it to a broker from where on it will be distributed to the subscribers. Sensor nodes actively send sensor measurement data to the gateway in this case. In contrast to that, in this demonstrator the gateway is an *active* component, which actively connects to the sensor nodes to set them up to send data to the gateway. Thus, the gateway opens and controls connections to the sensor nodes. This mechanism leverages Bluetooth LE Generic Attribute Profile (GATT) characteristic notifications. The main advantage of using this approach is that the gateway decides to which sensor nodes connections will be established instead of leaving control to the sensor nodes. Additionally, it eases initial wireless sensor network setup because it does not require authentication and authorization mechanisms at device level. In many cases, commercial off-the-shelf products can be used completely unmodified, saving a lot of effort. Moreover, this approach avoids the problem of storing network credentials on the sensor nodes.

However, a drawback of this approach compared to the generic architecture is, that it is theoretically possible to substitute sensor nodes with malicious ones. This could be done by spoofing the MAC address of a Bluetooth LE sensor node and mimicking exactly the behaviour of a proper sensor node. Even though we were aware of this issue, this was not considered as a real threat in the actual demonstrator setup since only the mobile robot's acceleration and rotation was transmitted.

The broker was configured to use authentication for publishing and subscribing and authorization in terms of allowed topics to publish/subscribe to. This is primarily to ensure that only white-listed gateways and applications are able to publish/subscribe data via the broker. The broker implementation allows very fine-grained access control. It is for example possible to restrict a particular client's access to certain topics and/or only publishing respectively subscribing.

Final sensor measurement data storage is not encrypted in our setup due to the insensible nature of the collected data. Protection of this data can be considered as a post-processing step. It is possible to extend the demonstrator with this kind of protection.

Messaging - For the demonstrator, MQTT was used as messaging protocol. This is in alignment with the generic architecture as well as with the translation of the generic architecture to the InsightProducts architecture. The gateway acts as a client to the MQTT broker and only publishes received sensor measurement data as MQTT messages. In addition to the dashboard example application as described earlier, we introduced an additional component, which was responsible for the semantic enrichment of the sensor measurement data. This component also connects as a client to the MQTT broker and behaves as subscriber and publisher. Subscriptions of this component are to topics, which correspond to the sensor measurement data. Therefore, the semantic enrichment component subscribes to sensor measurement data topics, produces corresponding RDF data from this sensor measurement data and finally publishes the produced RDF data via the broker. The RDF data consists of OWL micro-ontologies describing each sensor observation and is serialized to the compact and still human and machine-readable Turtle[14] format.

[14] See https://w3.org/TR/Turtle

3.6 WP6: Evaluation, Dissemination & Exploitation activities

Work performed

In this work package, the demonstrators were evaluated with respect to the metrics defined in WP2 and the requirements, respectively the feedback of the industry partners. Furthermore, all dissemination and exploitation activities were coordinated in this work package. For a tabular presentation of these activities, compare Section 7.

Results achieved by Sirris

The developed SPICY demonstrator provided an evaluation of the designed architecture during the project. In addition, it was used to validate several security metrics in terms of data security in device, in transit between the devices and the gateway and between the gateway and the backend, and secure storage.

In addition, several project developments were validated in cooperation with individual user group members, such as: cost estimation of cloud infrastructure for monitoring of a large scale of connected products, process of data analysis: from preliminary data exploration, to labelling, to creation of qualitative data set, to extraction of product and environmental insights based on collected data.

Finally, the project resulted in several follow initiatives on a national level such as the Smart Product Exploration[15] project.

Results achieved by Hahn-Schickard

Regarding the evaluation of the clean room demonstrator here two sets of sample data of two sensors are presented exemplarily. The Environmental Information System (EIS) recorded air flow, illuminance, temperature, pressure and humidity data. In Figure 37 a section of about one month is plotted. In the air-flow data and the illuminance data the single week days can be easily recognized. Temperature and humidity are controlled in the clean room and for this reason very stable. The measured pressure depends on the weather situation. During two days, temperature and humidity show a big deviation from their normal values (green square). This was caused by maintenance work of the air conditioning in the cleanroom. Such an anomaly can also be easily detected via algorithms. There are also two subsections where sensor data is missing. This indicates that the sensors may sometimes hang or data transmission may otherwise be disturbed.

[15] https://www.sirris.be/orientation-and-decision-making-smart-product-exploration.

Figure 37: Evaluation of environmental data in the clean room recorded by Environmental Information System (EIS).

The Illuminance recorded by the 3D-Light-Sensor is shown in Figure 38. In the zoomed section it can be seen very easy the activity of the machine. For “no activity” the illuminance is almost zero while during an etching process the illuminance is over 50 lux. It is even possible to distinguish between several types of activity. A flat activity curve indicates a pure etching process, while an oscillating curve indicates a combination of etching and passivation process. Between the three states “no activity”, “etching” and “etching+passivation” can be easily differentiated also by an algorithmic evaluation.

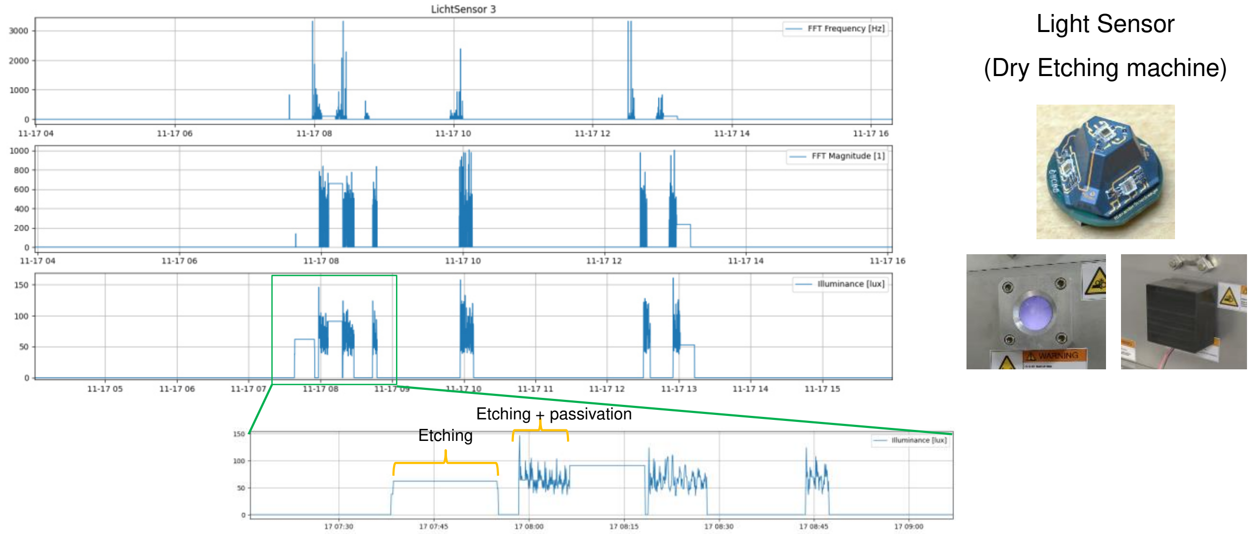

Figure 38: Evaluation of dry-etching data recorded by 3D-Light-Sensor.

Regarding the predefined metrics, the sensors (especially InsightProducts Sensor) can be mounted easily on different machines and can be powered by battery or wire-based. The data rate and latency values were found in the expected range, depending on the transmission protocol used. The reliability of the transmission for wireless protocols turned out to be critical in some situations and the connection of the sensor to the gateway was lost sometimes. Strategies were defined to overcome this problem for example resetting the sensor/gateway in predefined time intervals as well as trying to reconnect as soon as the connection is detected as lost. In terms of stability of the connection, clear advantages of a wire-based connection towards a wireless connection could be recognized (as expected before). The OPC-UA protocol turned out to be suited to transmit and save the gained data in an organized and structured way and serve as source for easy visualization of gained data and results. The detection of special events and anomaly based on the recorded could already be realized by using simple algorithms. The data on the OPC-UA server served as a central source for successful monitoring the current state of the connected sensors and machines.

A detailed list of the dissemination and exploitation activities of the project partners is listed in section 7.

Results achieved by FZI

The metrics evaluated by the FZI focus mainly on the efficiency of the introduced explicit semantic. That was not only an interest to the partners that originated the idea for such a semantic enrichment, but also to partners that saw the potential to adapt the technology for different use cases, based on enrichment and logging.

For the evaluation of the metrics as an example use case based on the Thingy:52 IoT sensor platform was defined. Within the use case the following data was measured:

- Ambient air temperature
- Ambient air pressure
- Ambient air humidity
- Ambient air CO_2 concentration
- Ambient air TVOC concentration
- Ambient light color
- Acceleration in three axis
- Rotation in three axis
- Sensor device battery level

All measurements were carried out using the same configuration of the sensor platform, the gateway and the broker. We used 30 Hz as update frequency for the sensors related to motion. All other sensors operated at 1 Hz, except for the ambient light colour sensor, which operated at 0.67 Hz. The overall WSN architecture and a corresponding component for the semantic enrichment were conforming to the SPICY demonstrator architecture, with the sole difference of using a dedicated machine for hosting the MQTT broker due to the reasons explained in the former section. In all measurements, we used an Intel® Core™ i5-750 CPU at 2.67 GHz with 8 GB of RAM for running the semantic enrichment component.

Computational overhead analyses in case of semantic enrichment

In order to measure the computational overhead of the micro-ontology generation from raw sensor measurement data, we instrumented the application source code. We collected the measurement data from each of the aforementioned sensors for 4 minutes, collecting a total of over 17.000 data points. Then, we compared the processing time to the produced sizes of

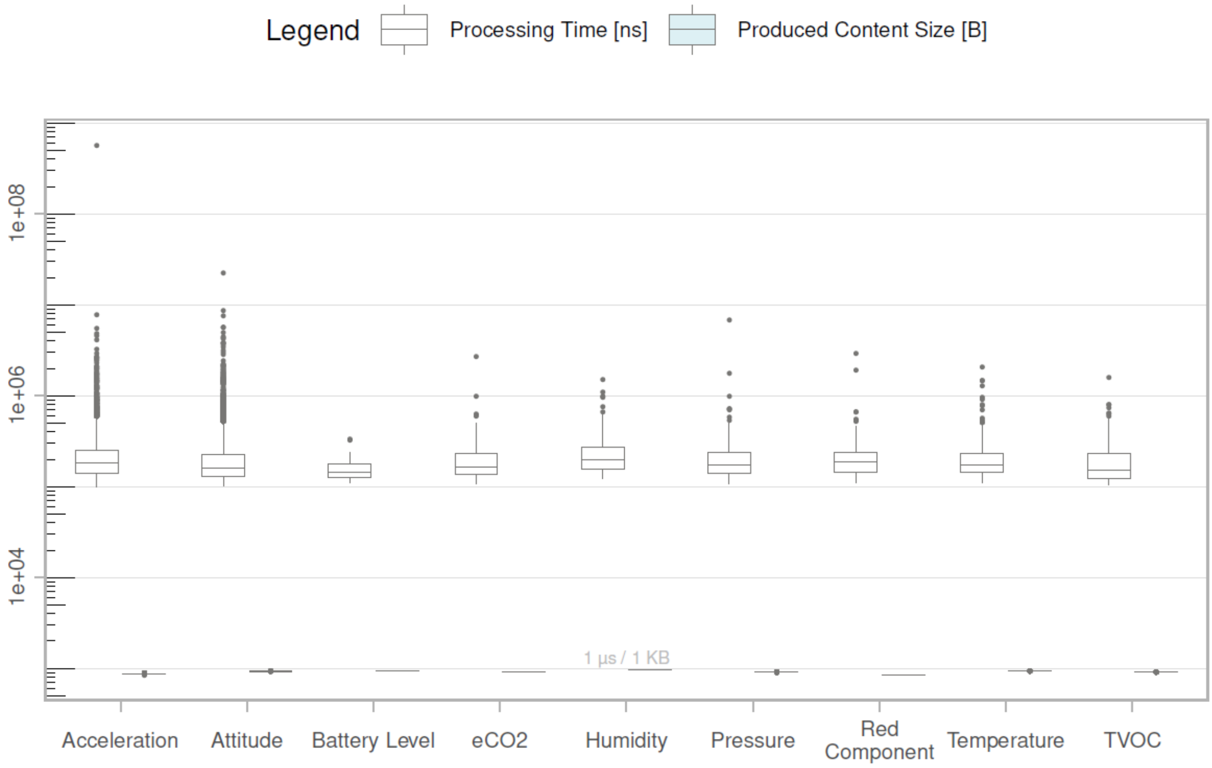

Figure 39. Computational overhead of the micro-ontology generation.

generated micro-ontologies for each data point. Please refer to Figure 39 for the results of this comparison.

We can see from these results that the majority of the processing time is performed in less than 500 µs with some outliers above 1 ms. This can be explained with the side-effects of the garbage collection of the Java Virtual Machine (JVM). The produced content sizes are below 1 KB. Please find more information on the produced content sizes in the next section.

From these results, we draw the conclusion that in a typical setup the overhead by our approach with regard to latency will mainly consist of the network transmission delay instead of the processing time. Therefore, when implementing such a system the system designer needs to take the network more carefully into account.

While the former measurement was a good starting point, we wanted to investigate more on the amount of time, which the different phases of the micro-ontology generation required while also taking compression of generated micro-ontologies into account. Therefore, we instrumented the source code of the semantic enrichment component so that we could measure the time it took for:

1. **Translation** of received MQTT messages to an internal data structure used for further processing
2. **Generation** of a corresponding micro-ontology
3. **Serialization** of the generated micro-ontology to a serialization format of OWL
4. **Compression** of the generated, serialized micro-ontology

We collected measurement data for 5 minutes, which resulted in over 179.000 data points. Figure 40 shows the results of this measurement.

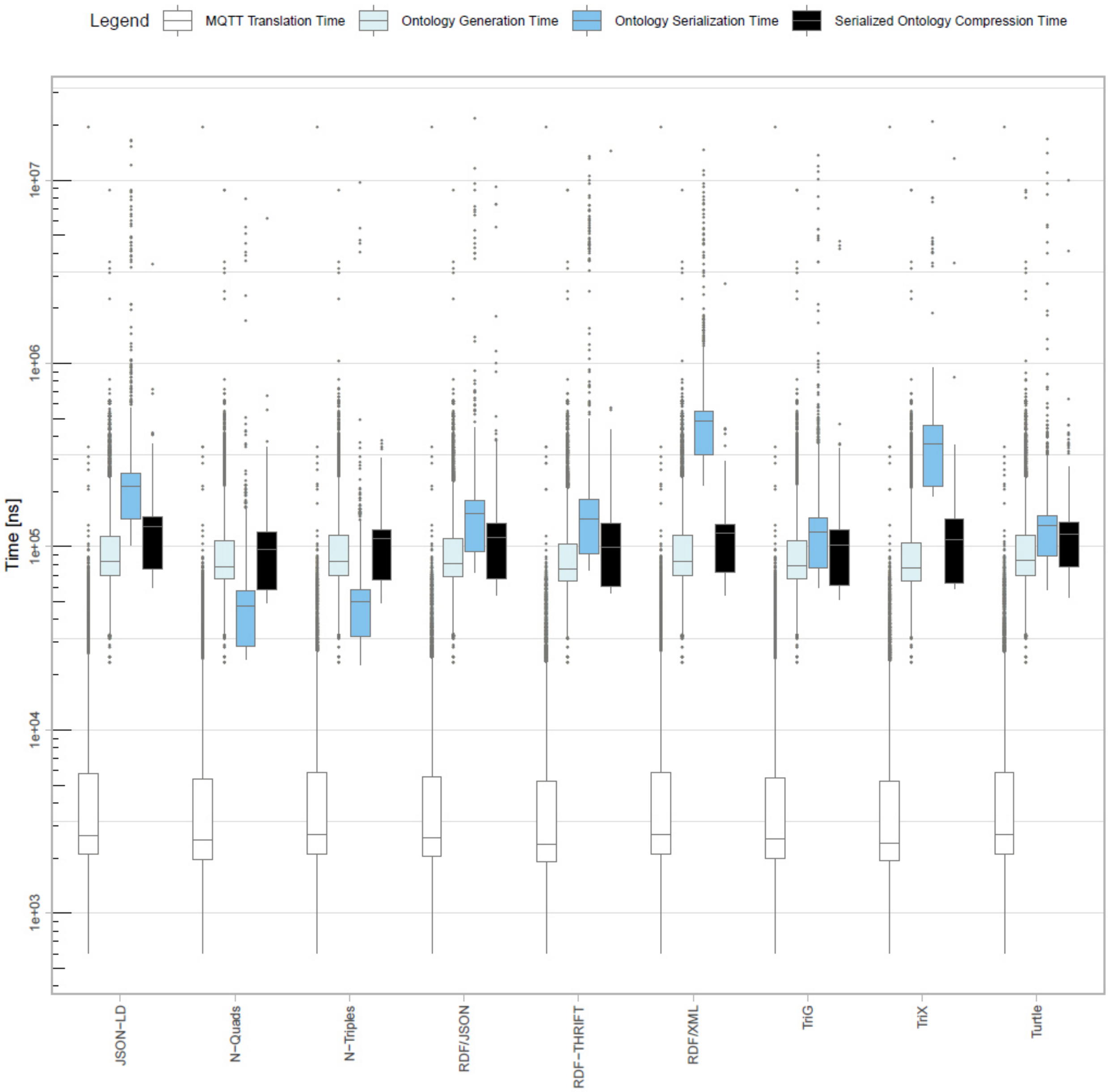

Figure 40. Processing times of the micro-ontology generation phases.

These results show that there are only minor differences in terms of processing times even between different serialization formats of OWL. This fact confirms the previous finding, that the processing time for the generation of micro-ontologies can rather be neglected in comparison to the introduced network transmission latency.

Data overhead analyses in case of semantic enrichment

Regarding the produced content sizes, we compared different common serialization formats of OWL:

- Turtle
- RDF/XML
- N-Triples
- JSON-LD
- RDF/JSON
- TriG

- N-Quads
- TriX
- RDF Binary

We collected measurement data from each sensor for 1 hour, collecting over 250.000 data points. Each micro-ontology was then serialized to each of the aforementioned formats. In addition to that, we also implemented GZIP compression of each of the resulting serialized micro-ontologies. Please refer to Figure 41 for the results of the comparison.

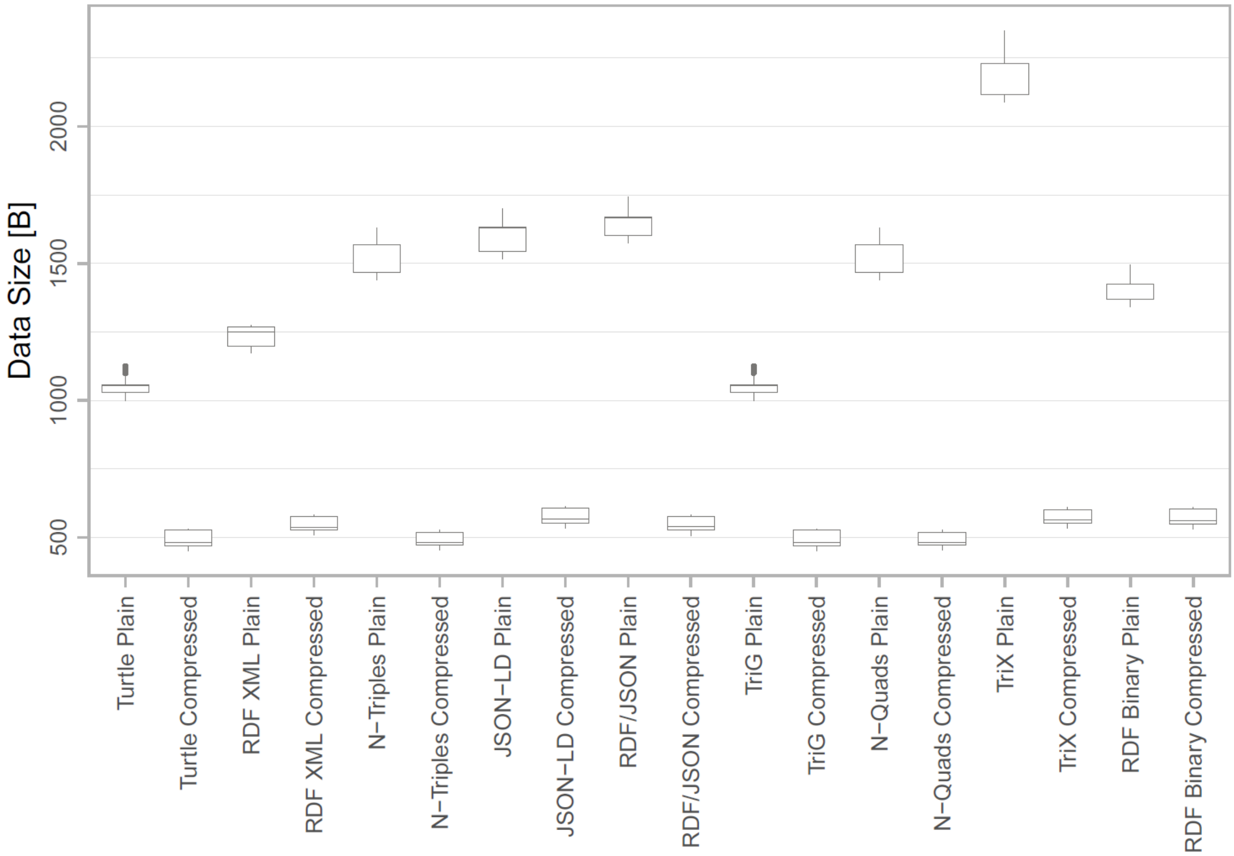

Figure 41. Comparison of micro-ontology serialization formats and compression.

These results show that there are significant differences between the OWL serialization formats with regard to produced micro-ontology content size. For example, micro-ontologies serialized in TriX format have almost twice the size than the same ontologies serialized in Turtle format. This is important for the design for the WSN system design especially when selecting and configuring the network infrastructure.

We can also see from the results, that compression can be used to effectively reduce the micro-ontology size. Therefore, compression can be used to reduce latency and required bandwidth for micro-ontology transmission in a network.

Moreover, the results show that when looking at a single serialization format the generated micro-ontologies have pretty much the same size. There is only little variance of the micro-ontology size even though different types of sensors were used. This can be explained by the fact that the implementation of the semantic enrichment component generates micro-ontologies, which have a quite similar structure and content. Referring to Figure 11, which shows a generated micro-ontology, one can see that firstly, only specific parts will change between micro-ontology instances (sensor value, unit, timestamp and contextual references such as to the sensor description). Secondly, differences of the actual content size of the parts, which change between two micro-ontology instances, will be insignificant. Thus, the fact that micro-ontology sizes have insignificant variance for a serialization format eases the required bandwidth analysis as the one we describe in the following section.

Of course, it is important to mention at this point, that the differences between generated micro-ontology sizes strongly depends on the WSN application and the implementation of the semantic enrichment component. In general, these differences can be more apparent. Nevertheless, the probability of designing a WSN system with a semantic-enrichment component generating micro-ontologies, which has insignificant size differences, is quite high because when following our approach of separating the ontological information in static/dynamic parts there will typically be only a small number of statements in each micro-ontology.

Required Bandwidth Analyses in case of semantic enrichment

The latter sections on the analysis of computational overhead and on the data overhead analysis of the semantic enrichment of sensor measurement data. WSN system designers should focus more on the network infrastructure as computational overhead can mostly be neglected.

An important aspect on this matter is the network bandwidth required for the WSN system to work properly. Considering the results presented in the latter sections and the concrete WSN application one can calculate the minimum required bandwidth:

> *A sensor is operating at 30 Hz and the size of each generated micro-ontology will be 1 KB, then the minimal network connection bandwidth is 30 KB/s.*

Even though the variance of micro-ontology sizes can be significant in general, it is reasonable to assume that it will be insignificant for a typical WSN application. Even if the variance of the micro-ontology sizes is significant, it is still possible to do the calculation and for example expect the worst case.

In any case, the WSN system designer has to consider the actual application to choose proper sensor operating frequencies. Based on the selected operating frequencies and the set of sensors network requirements can be derived. Network requirements such as the bandwidth can be defined with an additional margin to provide more flexibility and to avoid unwanted drops in throughput or other forms of network congestion. There are many aspects which need to be considered when designing a WSN and the required network bandwidth strongly depends on the concrete application and also the on the concrete implementation of the semantic enrichment.

4 Information on expenditure financed from grants for PM

Hahn-Schickard

- Personnel (Item A of the financing plan)
  Based on the initial adjusted project plan, the following expenditures in person-months (PM) for personnel were realized during the project period

Item	Person-months
A.1 Scientific - technical staff	15,90 PM

- Equipment (Item B of the financing plan)
  No project-specific acquisitions in the reporting period

- Third-party services (Item C of the financing plan)
  No third-party services in the reporting period

FZI

- Personnel (Item A of the financing plan)
  Based on the initial adjusted project plan, the following expenditures in person-months (PM) for personnel were realized during the project period

Item	Person-months
A.1 Scientific - technical staff	24,75 PM

- Equipment (Item B of the financing plan)
  No project-specific acquisitions in the reporting period

- Third-party services (Item C of the financing plan)
  No third-party services in the reporting period

5 Necessity and suitability of the performed work

The work carried out in the period under review essentially followed the planning in the project application and in accordance with the earmarking. The formulated tasks were successfully completed. The type and scope of the work performed corresponded to the application and was thus necessary and appropriate for the successful implementation of the project.

6 Scientific-technical and economic benefit of the results achieved in the project, especially for SMEs

Sirris

A summarizing documentation of the projects findings can be found in the public blog https://www.sirris.be/insightproducts-road-relevant-product-data

Several industrial demonstrators were realised validating the developed building blocks around the following five areas:

- Optimal use of sensor and communication solutions supported by and supporting qualitative data
- Determining how to acquire qualitative data and extracting actionable product insights from it
- Enhancing the design and operations of a product based on qualitative data insights
- Enabling a product to be deployed as a service ('product-as-a-service')
- Gaining access to customer data

These included experimentation with industrial datasets in different real-life industrial settings, design and development of several industrial sensors such as the environmental monitoring InsightPro sensor, and exploration of the benefits of fast prototyping in view of initial product monitoring through the SPICY demonstrator.

Hahn-Schickard

The goal of the entire InsightProducts project and in particular of the economic transfer measures was the successful use and adaptation of the results, especially by SMEs. Through the work in the project together with our partners, new know-how has been built up, which has formed a valuable starting point for this project and future research and product developments.

The InsightProducts-Sensor and other results developed in the project can be easily integrated into existing machinery of SMEs and offers various advantages:

- Flexible overall system architecture supporting various transmission technologies and standards as well as sensor interfaces
- InsightProducts-Sensor/CPS to measure the most important physical quantities for machine monitoring (acceleration/vibration, acoustic sound, temperature, humidity, light intensity, ..) easy to mount on machines
- Building blocks enable companies to build up a condition monitoring of their machines in quick time with just minor adjustments of the hardware-setup.
- Easy adaptation to the specific needs of customers/companies
- Wireless and wire based communication via different IoT-protocols
- Battery operated and mains operation
- Preprocessing of data with “Random forest”-algorithm directly on the sensor possible

The results of the developed clean room demonstrator will be used in further projects at Hahn-Schickard (e.g. Mittelstand 4.0 - Kompetenzzentrum Textil vernetzt) to learn more detailed information from the recorded sensor signals. Here anomaly recognition and machine learning algorithms will be employed. The demonstrator is also presented to visitors from interested companies and organisations.

In all participating research centers, the work in InsightProducts has created know-how that complements existing competencies and expands the range of engineering and software technology services. At Hahn-Schickard, this goes in the areas of MEMS sensors and sensor nodes, wireless and wire based communication protocols, IoT platforms and network topology, as well as in teaching at Offenburg University of Applied Sciences and the University of Freiburg. The knowledge built up will also be important for the acquisition and realization of future public funded and industrial projects as well as product developments.

The InsightProducts-sensor has already been adapted by the company Softing for their IoT-demonstrator and has been presented on the virtual fair “SPS-Connect”. The presentation was also released on the MicrosoftBusiness-Youtube Channel: (https://www.youtube.com/watch?v=hJqlPWE03vk&t=10m35s)

FZI

The prototype developed by FZI and adapted to the SPICY, respectively the generic architecture was also used to develop and investigate a concept for semantic enrichment of measurement data from wireless sensor networks. Because of the coordination with one German company from the industrial steering committee and in response to requests from the Belgian User Committee, research was conducted in this framework to identify basic requirements for semantic enrichment of sensor measurement data. The results of these investigations quantify the necessary effort for semantic enrichment of measurement data and thus serve as a support for companies when making decisions regarding the use of semantic enrichment in wireless sensor networks. Furthermore, the results allow companies to assess the required resources as well as the performance of all serializations of ontologies commonly used at the time of the investigations for the representation of measurement data. In addition, the results can be used to investigate the minimum bandwidth requirements for semantic enrichment of companies in order to be able to carry out an adequate selection, structuring and dimensioning of the corresponding system infrastructure.

The concept for semantic enrichment of measurement data in wireless sensor networks, as well as the prototypical implementation of this, was also described in a scientific publication and the results of the investigations were thus made available to the public. In the paper, the concept of semantic enrichment in wireless sensor networks is described. The novel approach - in combination with techniques of model-based development enables an explicit representation of the configuration, as well as an acceleration of the (re-)configuration of such a network and considers both the development and the operational phase of the life cycle of wireless sensor networks. From a business perspective, the description of the approach, together with the results on the aforementioned research, can serve companies as a whole as a conceptual basis and cornerstone in the planning and development of tools and products for the deployment, provisioning, and management of wireless sensor networks.

The developed demonstrator will be used, for public events or bilateral discussions with industrial partners. A first version was already presented during the FZI Open House, to explain the methods developed within InsightProducts.

In cooperation with partners from the industrial steering committee, a topic for further research was identified that builds upon solutions achieved in InsightProducts. The basic idea is, to use similar techniques developed in the project to provide an explicit semantic to sensor data for data created within the system design, especially for simulation-based verification. The idea is to have beside the actual simulation tool, a tool for semantic enrichment that annotates the generated data, and helps to interpret the data in a later state.

7 Transfer of project results to economy

Tabular listing of the executed and planned transfer measures.

Transfer activity	Status
Project web page	Done Sept. 2018
Coordination with industry partners as well as regular User Committee Meetings (UCM) (German: Treffen des Projektbegleitenden Ausschusses)	
• 1st Belgian UCM	Done - Oct. 2018
• 1st German UCM	Done - Feb. 2019
• 2nd Belgian UCM	Done March 2019
• 3rd Belgian UCM	Done Oct. 2019
• 2nd German UCM	Done Nov. 2019
• 4th Belgian and 3rd German UCM	Done April 2020
• 5th Belgian and 4th German UCM (Final Meeting)	Done Nov. 2020
• Bi-literal communication with different industrial partners	Done 11 meetings between Sept. 2018 – Feb. 2020
Articles in industry-oriented and research publication / Collective dissemination actions	
• “Knowledge transfer and dissemination initiatives in the domain of product usage monitoring”- Launching event of its EluciDATA Lab	Done Sep. 2018
• “InsightProducts offers support to improve product and service offering with product insights” - https://www.sirris.be/insightproducts-offers-support-improve-product-and-service-offering-product-insights	Done January 2019

• InsightProducts – "Condition monitoring and predictive maintenance" – HannoverMesse 2019, Hannover	Done April 2019
• "InsightProducts - Actionable insights into product service delivery" - Brokerage event of the EluciDATA Community	Done April 2019
• "InsightProducts demonstrates how to acquire qualitative product data to improve product & service offering" - https://www.sirris.be/insightproducts-demonstrates-how-acquire-qualitative-product-data-improve-product-service-offering	Done June 2019
• InsightProducts – "Condition monitoring and predictive maintenance" - Sensor+Test fair, Nuremberg	Done June 2019
• "InsightProducts – objectives and demonstrators "- First public workshop for InsightProducts	Done Oct. 2019
• "Ontology-based Requirements Transformation" - IEEE ISSE 2019	Done Oct. 2019
• "InsightProducts - Actionable insights into product service delivery" - Flanders Make Symposium 2019	Done Nov. 2019
• "Delivering value through data science & AI" - Smart Products Unlocked event	Done Dec. 2019
• "InsightProducts - Actionable insights into product service delivery" – FZI OpenHouse	Done Feb. 2020
• Hannover Messe 2020	Cancelled (Covid-19) Apr. 2020
• Expert workshop, VDC TZ St. Georgen	Cancelled (Covid-19) May 2020
• Sensor+Test 2020	Cancelled (Covid-19) Jun. 2020
• "How data mining can improve your product and service offering" - (https://www.sirris.be/acquiring-qualitative-product-data-improve-your-product-and-service-offering)	Done July 2020
• "LEMONS: Leveraging Model-based Techniques to Enable Non-Intrusive Semantic Enrichment in Wireless Sensor Networks" - Software Engineering and Advanced Applications (SEAA)	Done Aug. 2020

• Presentation of InsightProducts at "Retrofit for the Internet of Things" event of "Mittelstand 4.0 Kompetenzzentrum Textil vernetzt"	Done Sep. 2020
• InsightProducts Sensor was presented on the SPS-Connect 2020 within a demonstrator of Softing AG. Video published on MicrosoftBusiness Channel: https://www.youtube.com/watch?v=hJqlPWE03vk&t=10m35s	Done Nov. 2020
• "Cost-effective use of relevant data - the InsightProducts case" - htatps://www.sirris.be/cost-effective-use-relevant-data-insightproducts-case	Done January 2021
• InsightProducts at the Smart Service seminar organized by BEMAS, the Belgian Maintenance Association	Done January 2021
Internal dissemination events and demonstrator setup	
• Sirris – internal cooperation in the context "smart connected products"	Done Q4/2018
• Sirris – internal cooperation in the context "smart connected products"	Done Q4/2019
• FZI – Cooperation with the internal project HoLL-Therm (demonstrator setup)	Done Q1/2020
• HS – Setup of the clean room demonstrator show case	Done Q2/2020
• HS – Cooperation with the project "Kompetenzzentrum Textil vernetzt"	Done Q2/2019 - Q4/2020
Exploitation activities (after the end of the project)	
• Project reconciliation using research findings of InsightProducts	Ongoing - Started Q3/2020
• Publications in journals and periodicals	tbd
• Use of the project results in academic teaching ○ University of Freiburg ○ University of Applied Sciences Offenburg ○ Karlsruher Institut of Technology	Ongoing
• Utilization of the project results in further research projects	Ongoing
• Presentation of the demonstrators and results to interested companies and visitor groups in place	Ongoing

• Acquisition of new industrial projects on the basis of the gained results
    o Bilateral Projects with SMEs
    o Feasibility studies for SMEs developing smart applications
    o Support for SMEs in the development of smart products
	Ongoing