IoTwins
BIG DATA OPTIMIZE INDUSTRY AND SERVICES

Grant Agreement N°857191

Distributed Digital Twins for industrial SMEs: a big-data platform

Deliverable 4.4 – Second Digital Twin Version Delivery for the Manufacturing Test-Beds

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 857191

Document Identification

Project	IoTwins
Project Full Title	Distributed Digital Twins for industrial SMEs: a big-data platform
Project Number	857191
Starting Date	September 1st, 2019
Duration	3 years
H2020 Programme	H2020-EU.2.1.1. - INDUSTRIAL LEADERSHIP - Leadership in enabling and industrial technologies - Information and Communication Technologies (ICT)
Topic	ICT-11-2018-2019 - HPC and Big Data enabled Large-scale Testbeds and Applications
Call for proposal	H2020-ICT-2018-3
Type of Action	IA-Innovation Action
Website	iotwins.eu
Work Package	WP4
WP Leader	GCL
Responsible Partner	GCL
Contributing Partners	FILL, BRI, BSC, ETXE
Author	Bassem Hichri (GCL)
Contributors	Edoardo Massano (GCL), Francesco Giovannini (GCL), Rizwan Qadeer (GCL), Verena Stanzl (FILL), Roberto Lambruschi (BRI), Javier Diaz (ETXE), Filippo Mantovani (BSC)
Reviewers	Giuseppe Di Modica (UNIBO), Morgan Cameron (ESI)
File Name	D 4.4 – SECOND DIGITAL TWIN VERSION DELIVERY FOR THE MANUFACTURING TEST-BEDS
Contractual delivery date	M30 – 28 February 2022
Actual delivery date	M31 – 29 March 2022
Version	1
Status	Final
Type	DEM: Demonstrator, pilot, prototype
Dissemination level	Public
Contact details of the coordinator	Francesco Millo, francesco.millo@bonfiglioli.com

Table of Contents

1 Introduction 4
2 Testbed 1: Wind-farm predictive maintenance system, BRI 4
2.1 Objectives of the testbed 4
2.2 Presentation of the digital twin of the testbed 4
2.3 Compliance and integration with the IoTwins platform architecture 5
2.4 Validation performed so far 7
3 Testbed 2: Machine tool spindle predictive behaviour, FILL 8
3.1 Objectives of the testbed 8
3.2 Presentation of the digital twin of the testbed 9
3.3 Compliance and integration with the IoTwins platform architecture 10
3.4 Validation performed so far 11
4 Testbed 3: Crankshaft manufacturing system predictive maintenance, ETXE 11
4.1 Objectives of the testbed 11
4.2 Presentation of the digital twin of the testbed 12
4.3 Compliance and integration with the IoTwins platform architecture 14
4.4 Validation performed so far 15
5 Testbed 4: Predictive maintenance and production optimization for closure manufacturing, GCL 18
5.1 Objectives of the testbed 18
5.2 Presentation of the digital twin of the testbed 18
5.3 Compliance and integration with the IoTwins platform architecture 22
5.4 Validation performed so far 23
6 Conclusion 25

1 Introduction

This deliverable provides the latest digital Twin version for the manufacturing testbeds in WP4 in which four large-scale industrial testbeds in the manufacturing sector were developed. The achievements and results in the following sections have been made according to the architecture for distributed and hybrid digital twins defined within WP2 and WP3. The objectives and Methodology of the work package are detailed in the previous deliverables of WP4 (D4.1, D4.2 and D4.3).

2 Testbed 1: Wind-farm predictive maintenance system, BRI

2.1 Objectives of the testbed

A Wind Turbine Generator (WTG) generally operates 24/7 with a constant requirement of increasing Annual Energy Productivity (AEP) and reducing unexpected downtimes, (due to adverse weather conditions that cause excessive wear and tear of wind turbine components), that can cause important losses of energy production.

Although the yaw gearmotor represents just 1.25% of the entire WTG costs, the related costs to the unproductive or damaged WTG are high. With corrective actions (after a failure occurs) the estimated costs of intervention/repair vary from 60k€ to 200k€ per turbine; in addition, the cost of production losses during the intervention can vary from 2k€ to 4k€ per day.

Test Bed 1 value proposition to be addressed to Customer Needs is finalized i) to have a higher annual energy production, ii) to reduce Maintenance & operational costs, iii) to have a reliable components' lifetime estimation and lower stocks of spare parts.

Our business model can offer additional services like: Smart yaw drive products, Predictive Maintenance services, Automatic spare parts ordering service (in future), Pre-installation/Run Time set-up simulation and parameter setting service

The benefits to the end user will then be: reduction of maintenance costs, improvement of the WTG set-up, provisioning of smart IoT-ized drives, increase of AEP, reduction/optimization of downtimes.

2.2 Presentation of the digital twin of the testbed

An initial version of a physics-based "virtual" twin of a wind turbine yaw drive system has been developed.

The physics-based model has been developed using SimulationX, I.e., the ESI proprietary system simulation tool, whose strength in multi-domain physics-based modelling lends itself very well to simulating complex cyber-physical systems such as WTGs. The model represents the control system, motors, and rotor of the yaw system of a turbine with a similar architecture to the target one, for which historical and manufacturer's data such as controller algorithms were provided by BRI and KKWind.

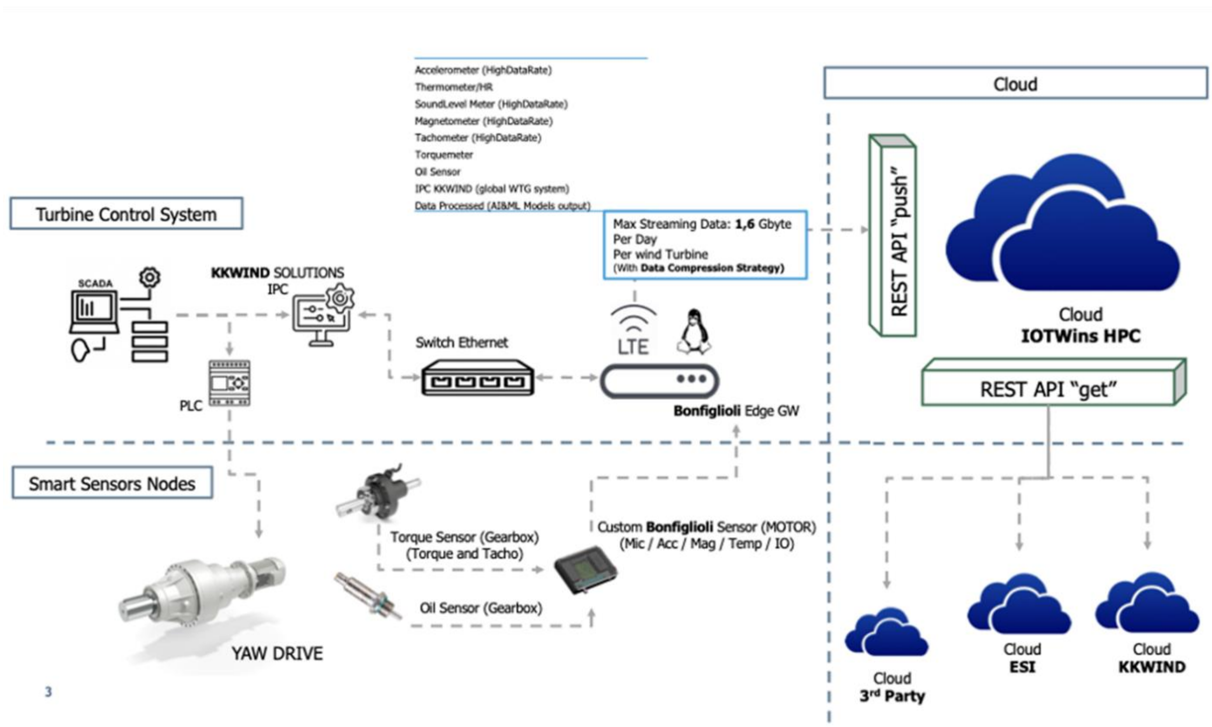
The initial studies have focused on comparing the real and simulated data streams about speeds and torques of yaw motors. So far, the paucity of available environmental data (wind speed, wind direction, etc) for the test turbine have limited the fidelity of the model that was achievable. These limitations will ultimately be alleviated by the high-frequency data that will be generated by the target WTG. Furthermore, ongoing work using data-driven methods for time-series data (by ENSAM) will be applied to the yaw acceleration data as a precursor to classifying the health of the system. The goal is to integrate the two models (physics-based and data-driven) to construct a "Hybrid Twin" model that can be applied to predictive maintenance applications.

The linked video shows a simulation performed using the current version of the physics-based model, with comparison of the actual wind speed direction and the computed yaw direction of the nacelle model.

[Yaw system simulation]

2.3 Compliance and integration with the IoTwins platform architecture

The aim is to merge the turbine control system (SCADA) and the purpose-built multi-sensor platform system through an Edge Computer being capable of gathering data from these systems.



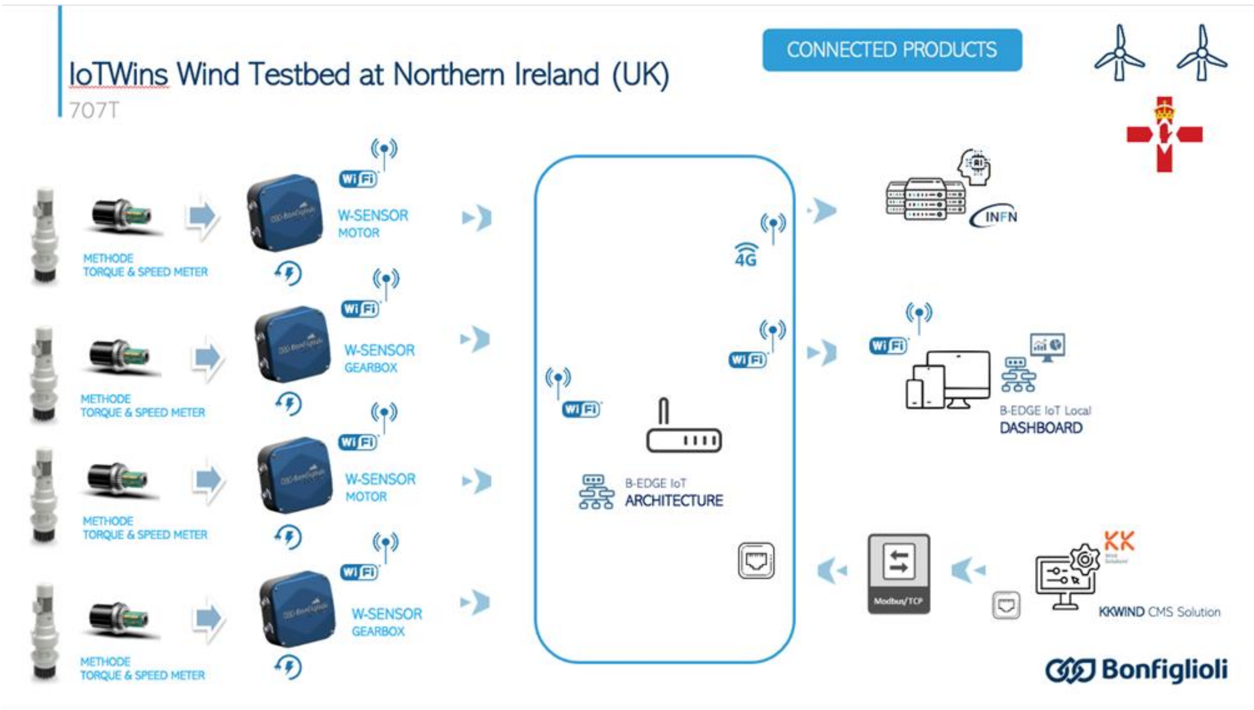


Figure 1: Hardware architecture overview

The two systems consist of smart sensors nodes on one side (high frequency data) and -the turbine control system plus the SCADA (low frequency data) of the WTG on the other one. The data stream that will be generated, based on team experience and know-how, will be based on both sensors applied on gearmotors and data coming from turbine control systems and possibly ecosystems. The data storage on the IoTwins cloud tier is using the following services: InfluxDB (for low-frequency data) and MinIO (for high-frequency data).

Further IoTwins services are still under evaluation together with TB1 partners, Cineca and UNIBO colleagues. Diagnostic and prognostic services could be particularly interesting for TB1.

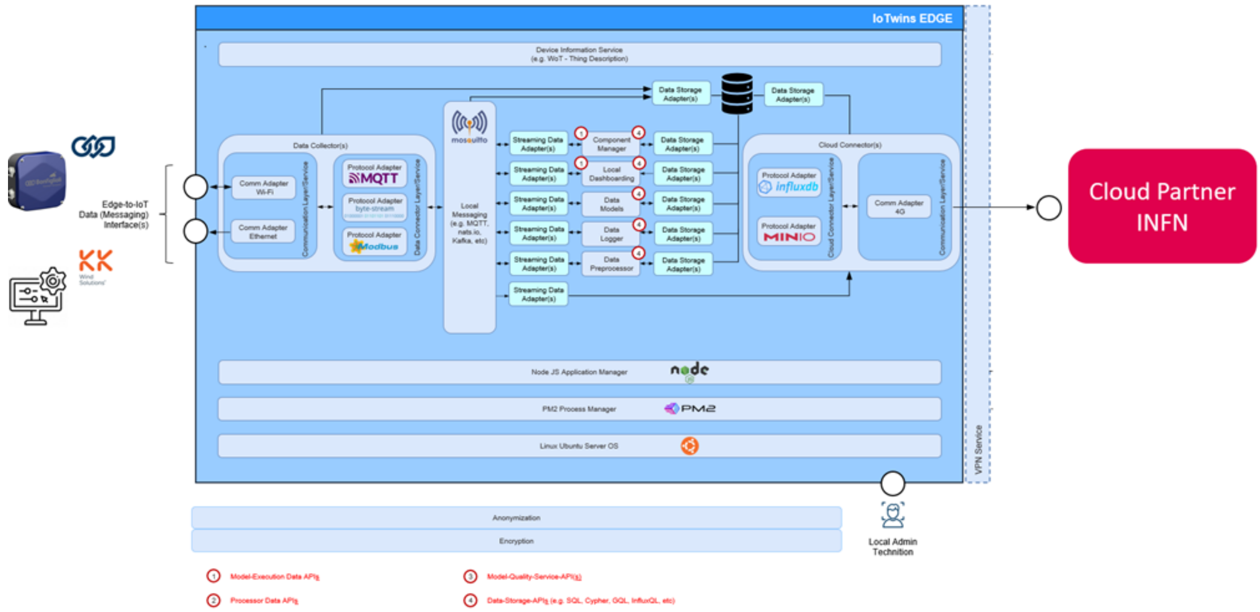


Figure 2: Architecture Overview

2.4 Validation performed so far

Initial results show that the implemented model of the nacelle of the WTG correctly tracks the changing direction of the wind. Figure 3 shows the low-frequency (every 15 minutes) environmental data that was available.

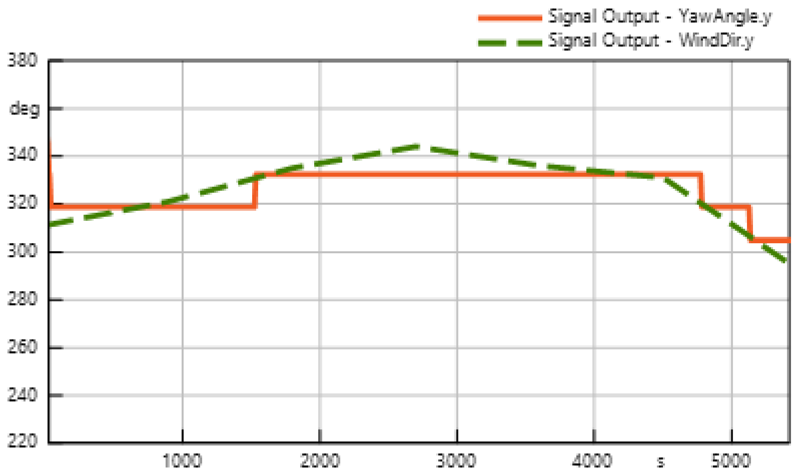


Figure 3: Simulated yaw angle (direction of WTG nacelle) vs actual wind direction

Ongoing work is focused on adapting this model to the target WTG. A first validation step aimed at verifying that the simulation model correctly predicts the yaw angle of the nacelle is ongoing. At the same time, aerodynamic model components have been developed and integrated in the overall system model using a combination of physics-based models and data collected by the Edge Twin (instrumented yaw drive and SCADA systems), with the aim of improving the correlation between simulated and measured torque values.

During WTG operation we identified some connectivity issues, which affected the data acquisition streaming. We suspect this issue is related to power supply problems: actions are in progress with WTG owner to analyse the issue and possibly connect the Edge to the uninterruptible power supply in the turbine or install its own backup.

Currently, one of the two Edge system recovered from a previous fault and started collecting and transmitting data. In the following figure, we report a snapshot of the local dashboard at the edge that displays the data collection and processing.

Figure 4: Local Edge Dashboard with collected and processed values

Where system simulation approaches are not able to accurately capture the physical phenomena, e.g., in the high-frequency accelerometer data generated by the instrumented yaw system, complementary data-driven methods such as Topological Data Analysis will be employed. Currently, the approach is being tested using historical acceleration data provided by Bonfiglioli from an instrumented yaw drive of a WTG similar to the target one.

Both data-driven and system simulation models can be embedded into lightweight executable components, thus enabling them to be deployed on the Edge machine, taking as input the necessary WTG sensor data to provide the operator with up-to-date estimates of the health of the system.

3 Testbed 2: Machine tool spindle predictive behaviour, FILL

3.1 Objectives of the testbed

This test-bed is aimed at creating multiple target-oriented digital twins of machine tools (Syncromill) in a shop floor (esp. automotive series production). By deploying simulation and ML models of machine tools, drives and spindles for detecting the condition and behaviour of the spindle manufacturing-relevant and quality-influencing parameters (load, forces, vibrations etc.) are predicted. This enables the reduction of unexpected rejects, breakdowns and downtime by optimizing load and performance indices.

A machine tool was set-up for this testbed in the Fill Future Zone, which is illustrated in Figure 5.

Figure 5: Syncromill for Testbed2 set-up in Fill Future Zone

3.2 Presentation of the digital twin of the testbed

The digital twin of Testbed 2 aims at creating multiple target-oriented digital twins of machine tools in a shop floor by deploying machine learning models for detecting the condition and behaviour of the machine tool, drives and spindles.

This is done in four steps:

1. Iotizing data from the machine and storing and processing them on edge
2. Data transfer to the cloud
3. Data Analytics and training Machine Learning models
4. Extension of the testbed

The architecture on edge has been designed according to the IoTwins reference architecture (discussed in deliverable D2.2), with a local messaging bus as central component. Experiments were carried out to collect data and perform analysis aimed at identifying important features. The edge device itself is managed by the Nerve platform developed by Testbed2 partner TIAG.

The current version of the digital twin collects all relevant data from the controllers of the machine and extracts the important features to distinguish the milling inside the material from milling outside the material, so to figure out the behaviour of the spindle. As part of the IoTwins project, the complete data recording pipeline was set up, the most important features for predicting the behaviour of the spindle were identified and a first version of the digital twin was developed. Furthermore, the data is transferred to the cloud.

The data was prepared and labelled manually. A neural network has been trained to perform the prediction of the timepoint where the tool enters the material. Several architectures and parameters have been evaluated. The dataset consists of 5181 samples, split into 80% training data and 20% for validation. The training was stopped after 38 epochs with around 1600 updates resulting in a validation loss (binary cross entropy) of 0.108 and an accuracy of 95.6%.

The loss-accuracy curve is visualized in Figure 6.

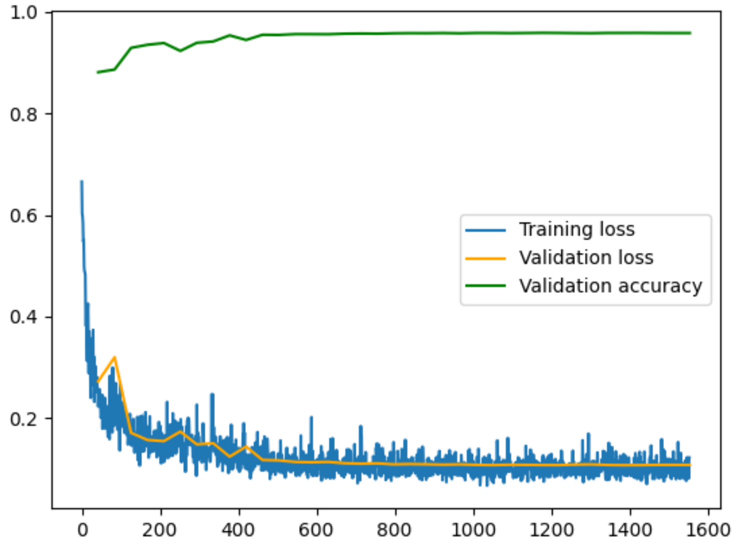


Figure 6: Loss-Accuracy Curve for Modeltraining in testbed 2

This model is containerized as docker container and embedded within the testbed 2 architecture. It will be further enhanced and enriched with other collected metadata to a digital product to reduce cycletime.

3.3 Compliance and integration with the IoTwins platform architecture

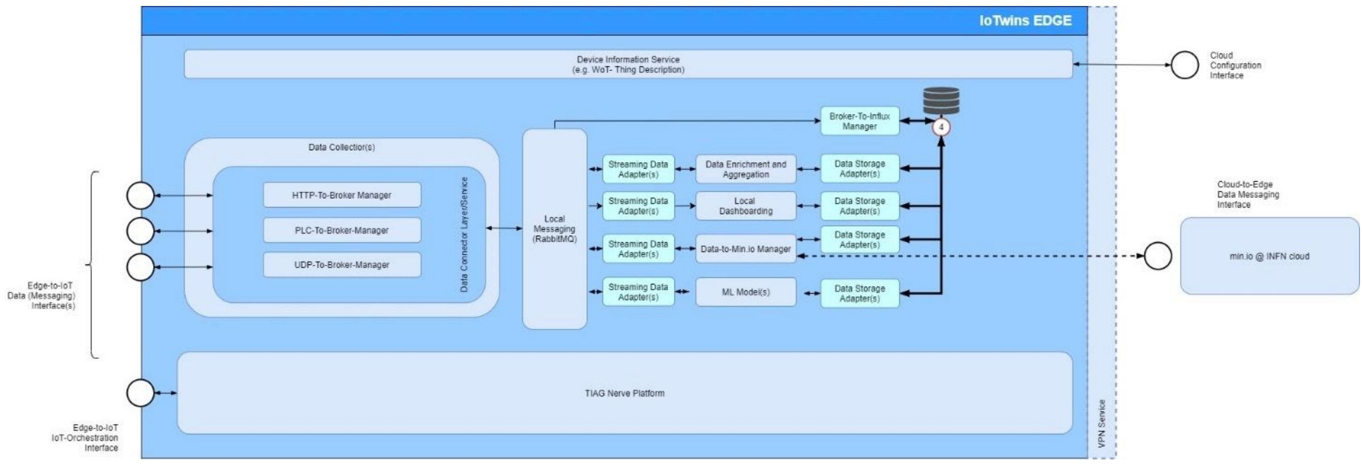


Figure 7: Architecture Overview

The architecture in Testbed 2 is compliant with the IoTwins platform architecture. Data Collectors collect the data from the different sensors and controllers and prepare them accordingly for the local message broker acting as the central backbone on the edge. From there, the data is transferred to containers for pre-processing the data as well as for model prediction. The data afterwards is stored in an instance of MongoDB. The raw data gets stored directly from the message broker using the Broker-To-Influx Manager. Furthermore, the data gets transferred batch-wise to the cloud using the MinIO service. At the current state, the cloud acts as central data storage using notebooks and manual scripts for data exploration and preparation for ML training. The focus is currently set to deployment on the edge. Summarizing, the components of the platform used in Testbed 2 are:

- Time-Series DB (InfluxDB)
- Message Broker (RabbitMQ)
- MQTT->InfluxDB Connector
- MQTT -> MinIO Connector
- Customized data analytics containers containing of Streaming Data Adaptors, data analysis and Data Storage Adapters.
- MinIO

3.4 Validation performed so far

To validate the work developed so far in Testbed 2, the system, the data and the machine learning model were considered.

Intensive test runs were made to successfully test the system for its robustness with regard to possible machine malfunctions, and the plausibility of the data was checked. The identified features for the predictions of the spindles were clarified with the domain experts and the results of the machine learning models were checked. All these tests were performed in-house and will be enriched with data coming from production as soon as the testbed is enlarged.

As part of Task 4.2.3 "Testing reliability and expandability", preparatory steps for Task 4.2.4 "Enlarging the testbed" have been performed. Suitable machines with similar mechanical structure have been identified to proceed with the extension of the testbed. In a first step, a small dataset of these machines has been collected to test the model and assess the expandability of the testbed. At the time of writing this deliverable, this process was still ongoing. First visual checks look very promising, however the process to quantify these results and conclude this activity is currently in progress.

4 Testbed 3: Crankshaft manufacturing system predictive maintenance, ETXE

4.1 Objectives of the testbed

As described in preliminary deliverables, the spindle head frontal ball-bearing is primarily focus for the testbed as a use case demonstration of potential applications towards maintenance. A spindle head is shown in Figure 8, placed in one of the versions of the testbed, where ball-bearing isolation for testing can be done.

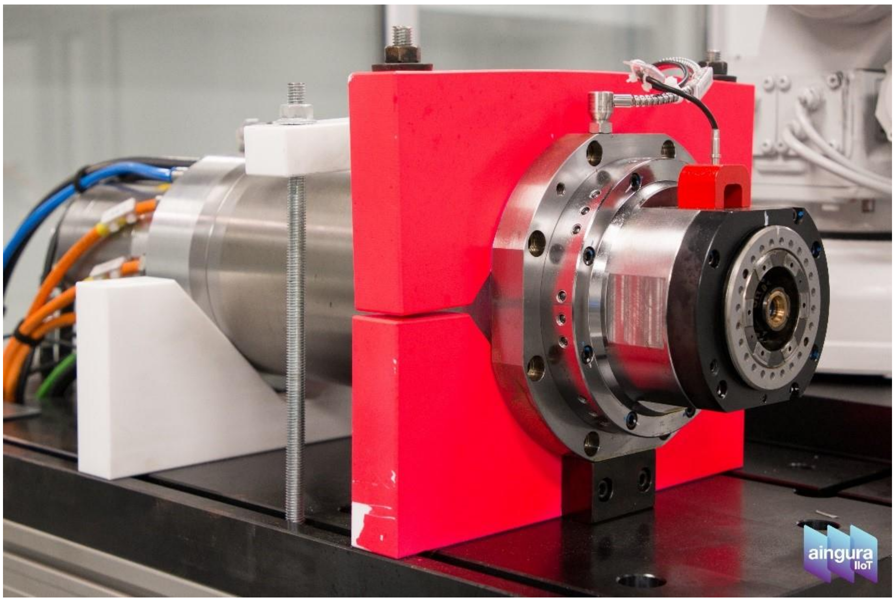


Figure 8: Crankshaft manufacturing spindle head (lab environment)

4.2 Presentation of the digital twin of the testbed

This testbed is certified by Industrial Internet Consortium[14] (IIC). The Industrial Internet Consortium brings together the organizations and technologies necessary to accelerate the growth of the industrial internet by identifying, assembling, testing and promoting best practices. Members work collaboratively to speed up the commercial use of advanced technologies. The testbed provides part of the infrastructure needed at three different deploying levels:

1. Lab: a synthetic setup with an infrastructure similar to the one deployed on the commercial products;
2. Factory-lab: a setup deployed on a real product when still in the factory (before deployment);
3. Production line: the actual final setup on the field and at scale.

These three scenarios will help to communicate, isolate (the ball-bearing analysis), test and validate all the required infrastructure related to data acquisition, pre-processing, and processing.

Therefore, at different deployment scenarios, the Aingura Insights (AI) edge computing node, from our Linked Third Party, Aingura IIoT, S.L. will be used to extract different variables related to the frontal ball-bearing. Additionally, the AI will be used to run feature subset selection over those variables to send only the require variables, avoiding redundant and not related that will compromise posterior data analysis algorithms.

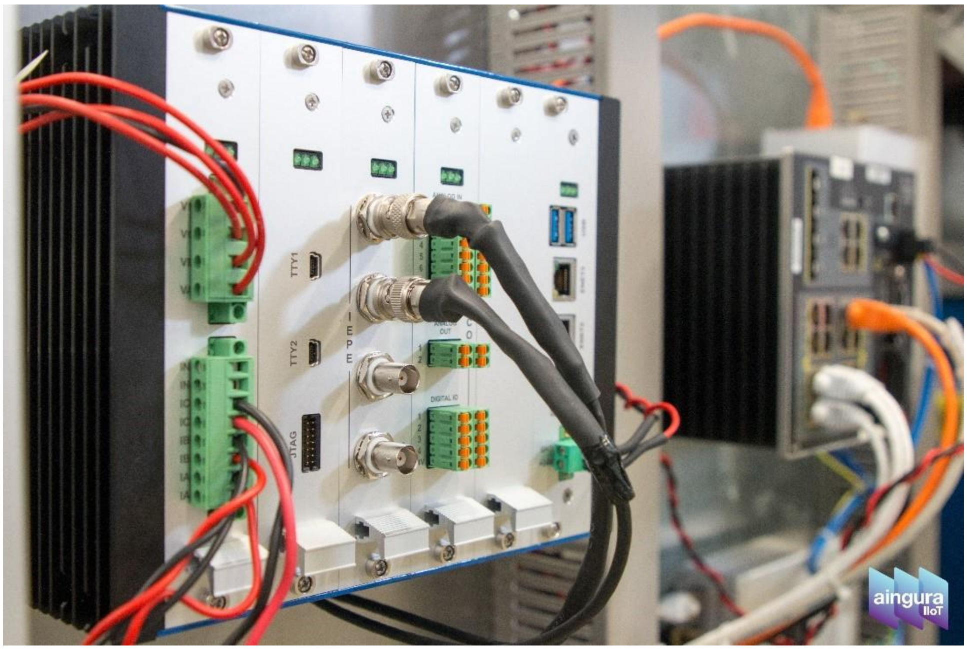


Figure 9: Aingura Insights (AI) edge computing node

Additionally, a specific testbench to extract data from ball-bearings has been built. From this testbench and the spindle head, the extracted data is mainly from accelerometers, thermocouples and CNC data (power, torque, current, winding temperature). The ball-bearing testbench is shown in Figure 10.

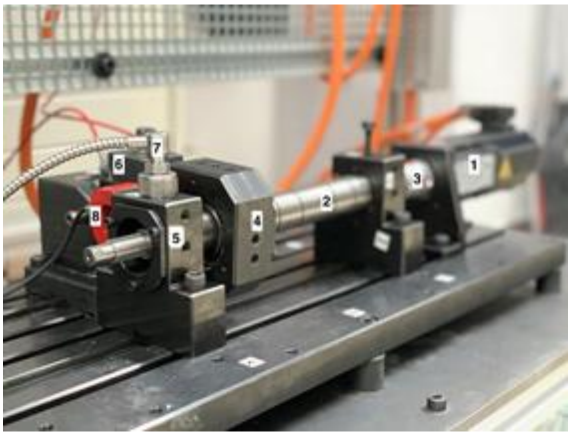


Figure 10: Ball-bearing testbench. (1) CNC servomotor, (2) rotary axis, (3) elastic coupling, (4) axial force heavy duty ball-bearing, (5) ball-bearing to be tested, (6) force actuator, (7) accelerometer, (8) thermocouple.

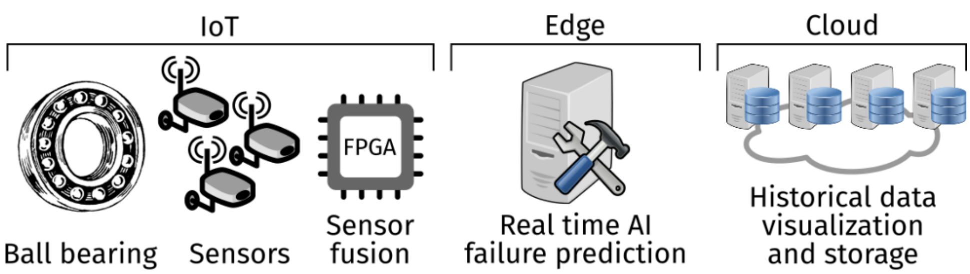


Figure 11: Dataflow of the testbed

The data collected by the sensors introduced in the previous sections are cleaned and fused by an FPGA-based IoT device (Aingura Insights). They are processed on the edge with AI techniques improving failure prediction. The testbed leverages the Cloud layer for storage and visualization of historical data. Figure 11 summarizes this dataflow.

4.3 Compliance and integration with the IoTwins platform architecture

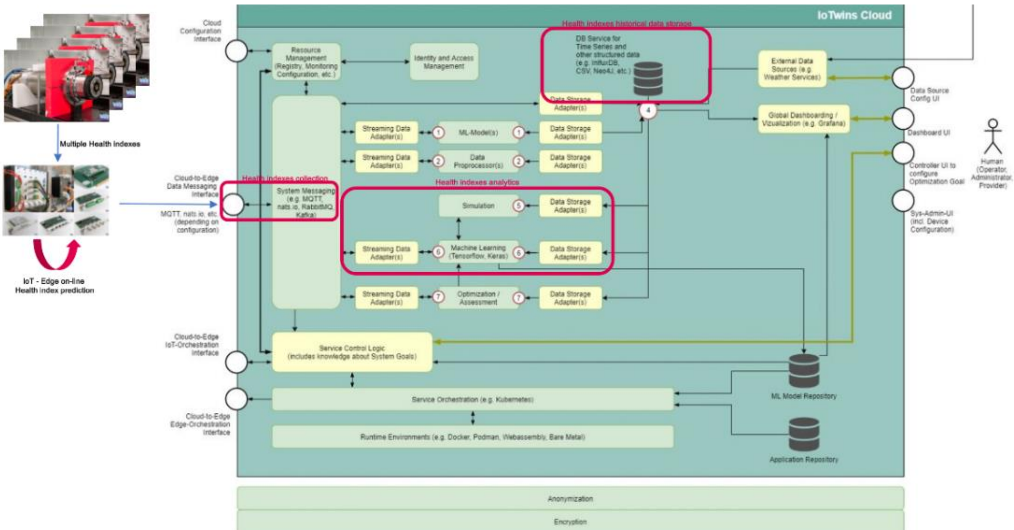


Figure 12: Integration with IoTwins cloud architecture

In Figure 12, the IoTwins platform architecture designed for the Cloud tier is depicted along with the architecture components exploited by TB3

The Aingura Insights module sends data from the process to the platform. This data are the insights from the edge ML-based and online computing algorithm, AR-As-LG-HMM. These insights are defined as Health Indexes related to each monitoring frequency and harmonics from the ball-bearing. The IoTwins platform's services will support the process of estimating the remaining useful life (RUL) based on these Health Indexes, with a posterior time series forecasting ML-based algorithm or similar.

Additionally, this integration creates the opportunity of having a centralized host gathering data from all the monitored locations. In this case, multiple machines with spindles will be monitored with one edge device for each of these spindles.

As a preliminary result, the initial step was to send the health index to a MQTT message broker deployed on the IoTwins platform.

First, in order to check this connection, a test was run with simulated data. The IoTwins platform hosts an influx2.0 server that expects data via a HTTP POST request to its port 443. As the platform is secured, a user and a password were provided in order to POST the information and access to web interface. The test was successful, and it was possible for the Aingura Insights to send the HTTP request with a negligible delay. The firstly connected device was a simple sensor reporting temperature and humidity of a laboratory.

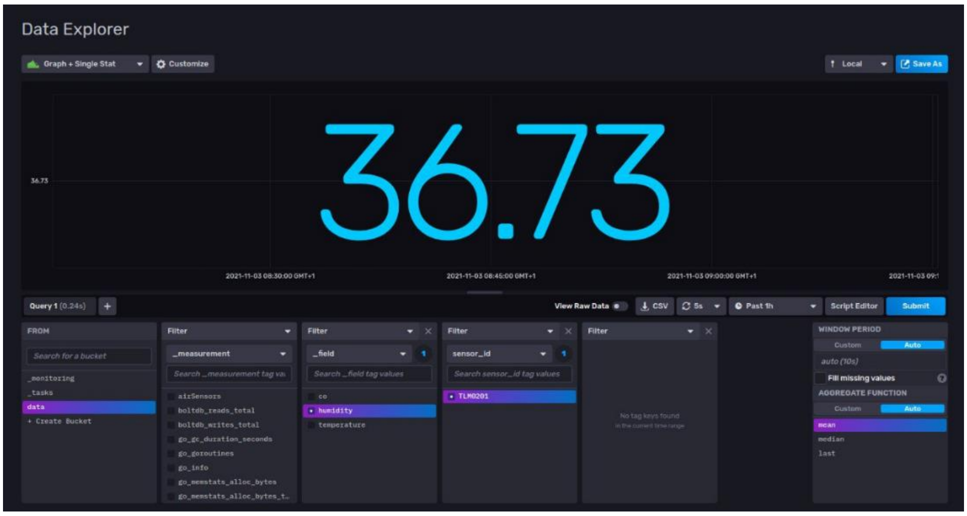


Figure 13: Results of first integration step in IoTwins platform dashboard with one value

As the connection to the platform was verified, health indexes were simulated and sent to the platform. Also, as the algorithm output formatting module was not fully implemented, the input values of the AR-AsLG-HMM was sent (namely, the amplitudes of the fundamental frequencies of the bearing.)

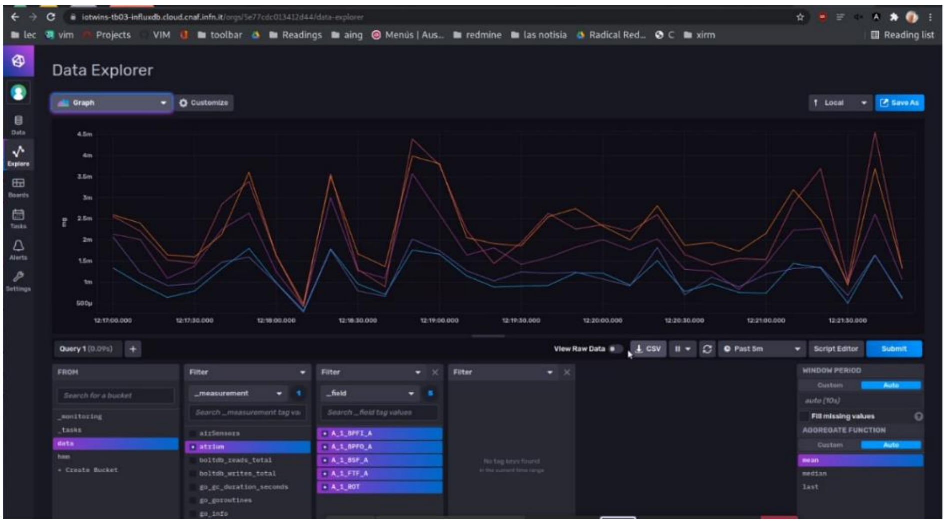


Figure 14: Results of first integration step in IoTwins platform dashboard with multivariate input

Once the data is stored in the platform, the dashboards showing the received data can be created and alarms can be configured to trigger when certain conditions are met.

As next steps, RUL estimation tests were carried out by sending to the IoTwins platform real health index coming from the lab version of the testbed (one fully operational spindle head and one ball-bearing run-to-fail system) every 15 seconds. Then, with the alerting system integrated in the Influxdb system, thresholds can be defined to alert an operator about remaining useful hours for the specific ball-bearing.

4.4 Validation performed so far

Testbed 3 is illustrated in this video (link). The testbed is divided into three different scenarios: (1) lab, (2) factory, and (3) production facility. The video describes just the first two scenarios, as the last one is performed at the customer premises (automotive OEM), where video recording is forbidden.

The first scenario is the lab, where a specific ball-bearing run-to-fail testbench is working 24/7 producing different types of datasets where the ball-bearing useful life is accelerated under different conditions. These datasets are used to simulate data streams from ball-bearings, in order to validate the Edge algorithms ideas, where no training is needed to estimate the remaining useful life. To this end, different failures modes are tested. Different forces (variable forces, temperatures, speed, …) are used with the aim of creating broad scenarios of validation. This dataset will also be used when the full integration with the complete IoTwins platform will be done, in order to validate ML based estimations done on the RUL. This testbench was purposefully built for the IoTwins project.

Figure 15: Accelerometer signals configuration on the Aingura IIoT computing node

Form this scenario, up to 8 run-to-fail tests have been done, where edge processing validation is carried out. Some examples of failures are shown in Figure 16.

Figure 16: Examples of failed ball-bearings

As explained in previous deliverables, this validation is done for the edge part, where data acquisition from the accelerometer, the filtering to extract the ball-bearing monitoring frequencies and the Machine Learning-based (AR-AsLG-HMM algorithm) Health Index estimation is running. It is important to highlight that this method is working online, consuming form the data stream (without data storage) and learning while its operation; embedded into an industrial/edge computing device. The validation of Health Indexes is the next step towards RUL estimation deployed at the IoTwins platform (as described in section 4.3). The complete deployment scenario is shown in Figure 17.

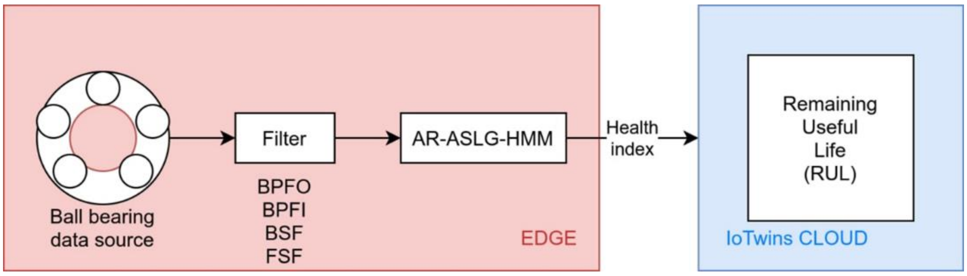


Figure 17: Deployment/validation scenario

This lab scenario also includes a spindle test facility, i.e., a larger machine, with all the systems and subsystems that make spindle heads run. This machine has all the lubrication, refrigeration, control systems, and power to operate a spindle head in dry cycle (no machining) when needed. This machine mimics the spindle head behaviour found in real machines, giving the testbed to extract real datasets with all the contextual variables found in CNC production machines. Although performing a run-to-fail test on the spindle heads is not feasible (these are really expensive elements and the designed useful life is around 2 years under working conditions), real-time monitoring can be done to showcase real operation and insight production in controlled and 24/7-available environment.

At this level, for the scenario and the run-to-fail described before, a demonstration/validation dashboard has been developed in order to capture singularities that can help to improve the performance. A screenshot of this dashboard is shown in Figure 18.

Figure 18: Validation/demonstration dashboard for edge insights (Health Indexes)

The second scenario shown is the video is the factory. When Etxetar produces CNC machines for its customers, those machines must be tested and validated prior to delivery. In fact, these machines need to perform a "production capacity" test, where around 200 real workpieces (real crankshafts) are manufactured. During this pre-production tests, Etxetar connects Aingura IIoT computing nodes to extract, test and validate different elements. Since this is a real production environment, this is the most controlled environment that is available to finally test different applications. As these types of data-based monitoring systems are highly sensitive to noise (huge amount of pre-processing is oriented to clean this noise), the scenario is highly useful to test the robustness of the solutions.

At the time of writing this document, there is no associated code already distributed on the IoTwins gitlab repository, because the integration with the IoTwins platform has only recently started. TB3 will upload selected datasets once the testing process is finished. It is important to mention that only datasets coming from run-to-fail testbench can be uploaded.

5 Testbed 4: Predictive maintenance and production optimization for closure manufacturing, GCL

5.1 Objectives of the testbed

As stated in the previous deliverables of WP4, the overall objective of this testbed is to predict the stoppage of an injection moulding machine caused by the failure of certain critical components (mainly injection bearings). The entire machine health is impacted by repeated failure several times which leads to a complete breakdown. This incident has an economic impact due to repairing costs and production loss.

5.2 Presentation of the digital twin of the testbed

The aim of this testbed (cf. Figure 19) is to aggregate big data from a selected machine and to analyse them with state-of-the art algorithms that will be created in WP3, with the final goal to establish an overall production management optimisation framework.

Within the scope of the IoTwins project, it has been added some critical ball bearings sensors on an injection moulding machine, measuring its vibration speed, acceleration and temperature. Between all variables, it has been identified which one are the most responsible for indicating the overall health of the machine. Then, following the ISO 10816-3 regulation, it has been defined thresholds for these measured values, where each threshold corresponds to one of four bearing statuses – normal, pre-warning, warning, alarm.

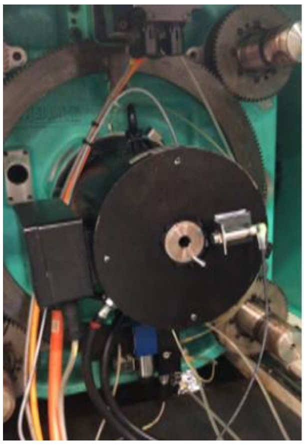


Figure 19: Moulding Machine testbench

The sensors, those already built in and those customize added, basing on the needs and the data type, communicate externally with specific protocols chosen by their manufacturer. For this specific testbed different communication standards were used, in particular:

- OPC / UA for data coming natively from the machine.
- CODESYS and MODBUS for consumption.
- OPC / DA for bearing data.

Due to the heterogeneity of the communication methodologies, emerged the necessity of a middleware communicator, that would be capable of integrating the information exposed by the machines and analysing the data. The technology chosen for this goal was KepServerEX, a platform of PTC, a world leader in the world of the Internet of Things. This tool incorporates hundreds of different communication drivers and can connect directly to the machine, exposing the normalized data to the outside.

Another PTC software called Thingwork was connected downstream to this tool, capable of processing the information received and storing it in any type of database, the one chosen for the project was InfluxDb, an open-source time series database (TSDB) developed by InfluxData. It is optimized for fast, high-availability storage and retrieval of time series data in fields such as operations monitoring, application metrics and Internet of Things sensor data.

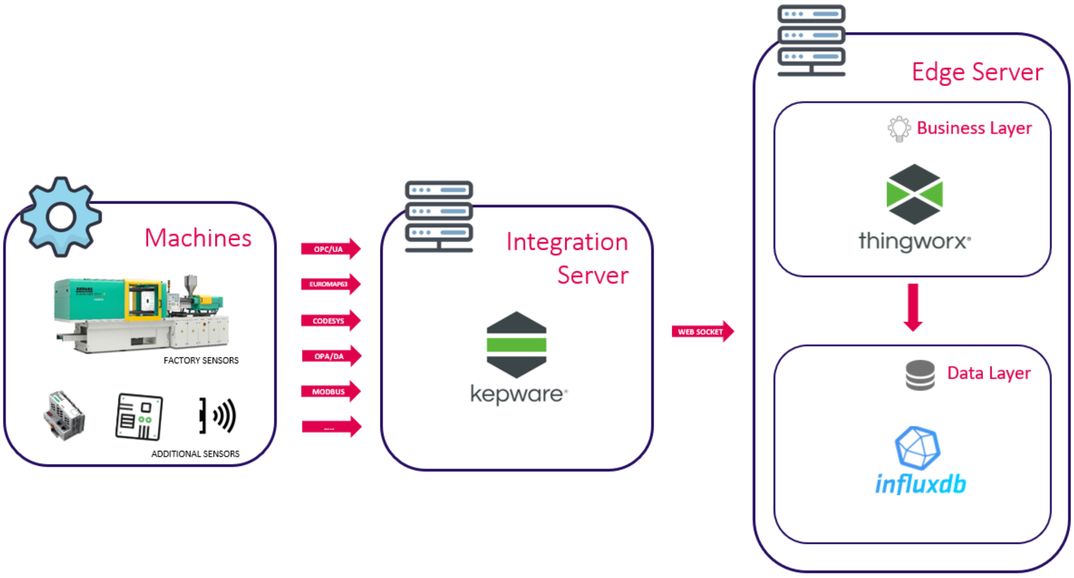


Figure 20: Software architecture

The chosen software architecture (cf. Figure 20) decided to approach this project was Software as a Service (Saas) which, in our case and to facilitate the comprehension of this document, we divided in front-end and back-end, the first is the client-side rendering that is responsible for the presentation layer, the aggregation and representation of the data; the back end oversees data acquisition and data processing.

The acquisition of data by the server side has as input our edge server (Thing Worx) and our database (MsSql), once the data is gathered it will be processed by numerous microservices, each one with a specific purpose and specification, and then be sent to the client side of our application that will receive this well-structured data and rearrange the information in a comprehensible way.

Our microservices were built using JAVA and JavaScript as languages and Spring and Express Js as its respective framework. There is one specific microservice in charge of handling the requests from our front-end and our edge server that will manage and redirect all of them to the respective microservice, it will also control the authorization and authentication for each request. Once the request of login is made, the application will validate that user, an access token (Json Web Token) will be created and used within every single further request.

The technology used by the front-end to render all data visualisable is Angular Framework that will make requests to the server side using a representational state transfer application programming interface (REST API), it is therefore possible for the user to view information from the machines, both real time and historical data. The platform allows users to have a corresponding digital twin of the machine with the advantage of obtaining aggregate information that can give added value to the data collected.

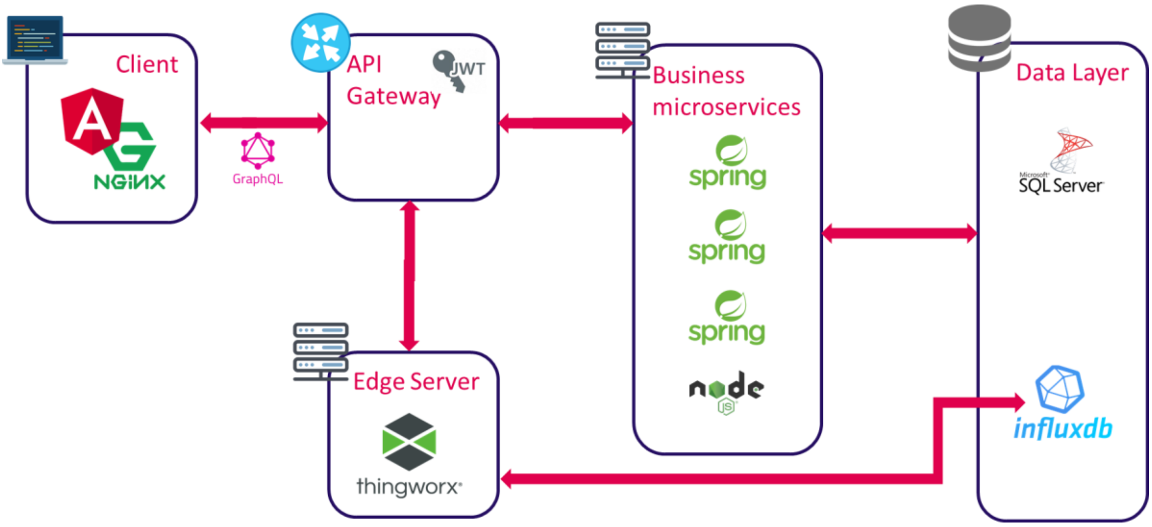


Figure 21: Data flow architecture

The goal of this TB is to predict the occurrence of the "alarm" status during normal production runs. This will be done by developing, with the help of our project partners in WP3, validating and testing a machine learning model, based on deep learning, capable of predicting the remaining useful life (RUL) of the ball bearing. The model outputs the RUL in the form of a survival probability (cf. Figure 22), with values ranging from 0 to 1, where 1 indicates that no failure will occur within now and a given time window, and 0 indicating the certainty that a failure will occur within now and a given time window.

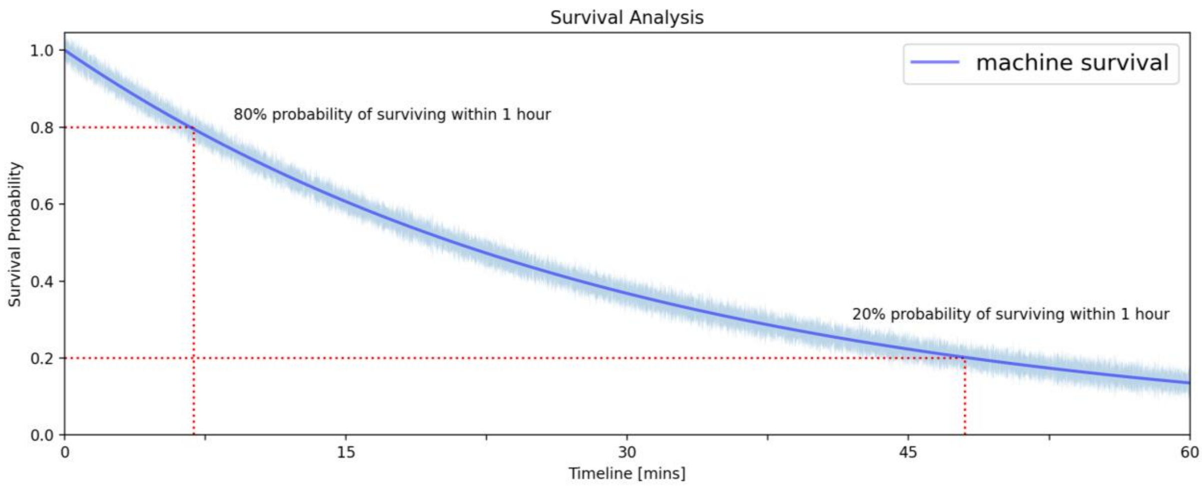


Figure 22: Survival Probability Model Outputs

For training the model we use the CINECA supercomputer. Every 15 minutes the data from our database, Influx DB, is collected, converted to a CSV file, and then sent to the HPC systems, in our case Marconi100.

Marconi100, the HPC system chosen for TB4, uses this information to train the model exploiting CINECA computing resources. The results of this activity are neural network models that are used to calculate RUL estimation of the injection machine component object of this TB.

Every 2 months it is scheduled a re-train activity with the up-to-date information in order to provide a more efficient and reliable model.

CINECA HPC also provides a specific REST API used to obtain in real time the information of the RUL expressed in hours.

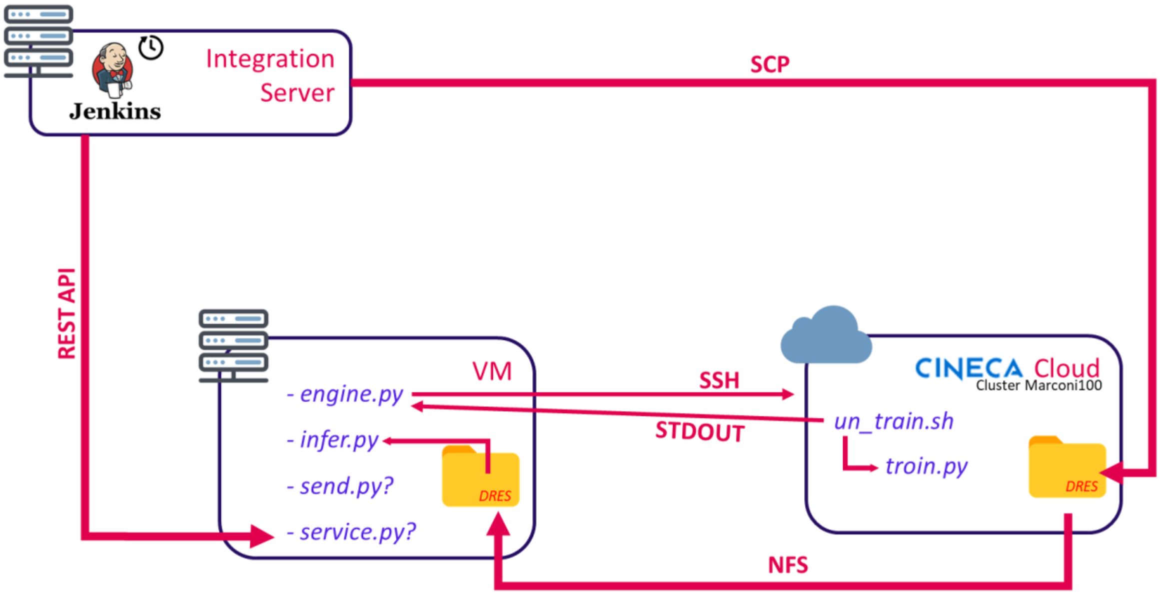


Figure 23: Integration Pipeline

5.3 Compliance and integration with the IoTwins platform architecture

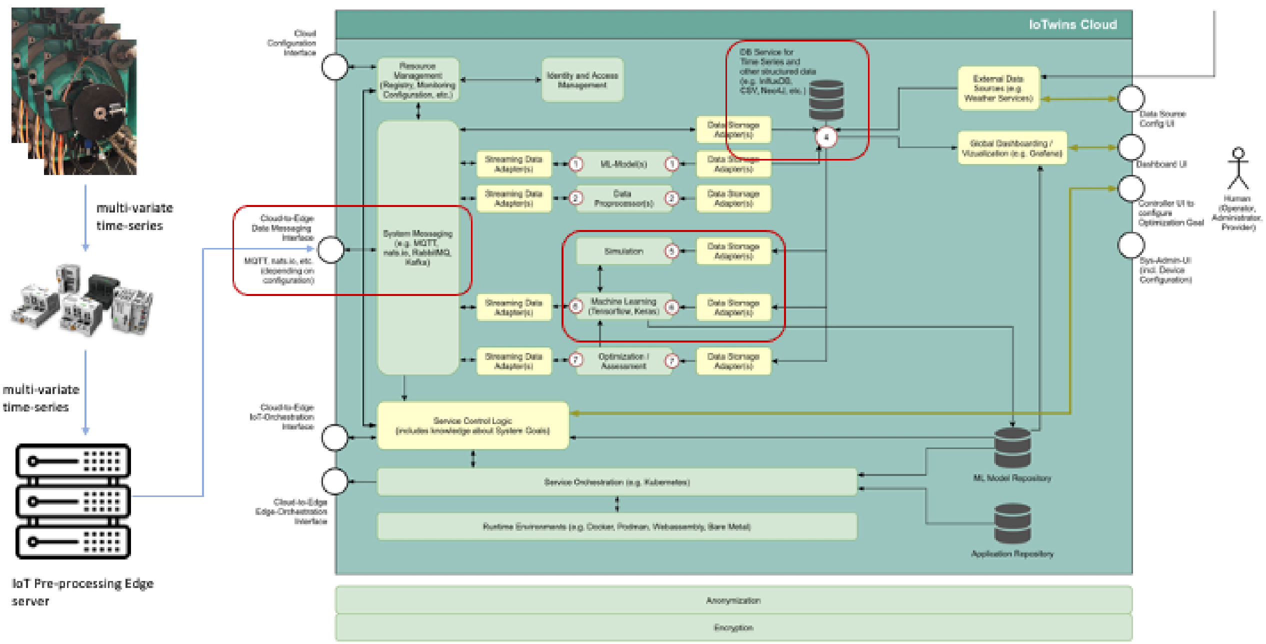


Figure 24: Integration with IoTwins cloud architecture

In figure 24, the overview IoTwins platform architecture designed is shown along with the architecture components exploited by TB4.

Using UPC/UA industrial communication protocol, machine sensors and the additional ones send data every 500 ms to the edge servers. The firstly connected sensors were the ones reporting electricity consumption and the temperature.

From the edge server1 (kepware) data were linked to the edge server2 (thingworx) using WebSockets. This integration gives the opportunity of having a centralized host edge server for all the monitored machines.

EDGE side it has been developed an architecture that uses components from an external partner to collect, normalize and store data from machine sensors. It has been developed a middleware platform based on microservices that will provide, among other things, connectivity with IoTwins cloud to consume its services to create, train, validate and run ML model.

Once data enters the edge server (ThingWorkx), it was sent immediately to the IoTwins Platform which hosts a time-series database (InfluxDB) that expects data via an HTTP POST request to its PORT 443. A password and username are mandatory to post the data to the platform.

In order to check the connection simulation data was sent to the platform. The test was successful and data flowed smoothly from the edge server mentioned above to the platform without any delay.

Figure 25: Results of the integration in IoTwins platform dashboard

Based on certain conditions, alarms can be configured to alert the operator.

RUL (remaining useful life) is calculated by a corresponding service in the IoTwins project and sent to the IoTwins platform. Using the same alerting system mentioned above thresholds can be set and defined to generate an alarm about the RUL for a specific machine.

The model is based on a recurrent neural network for time-series (CNN-LSTM). Interpretations of the model probability help the production managers to plan preventive maintenance to avoid any failures in the machine operation.

5.4 Validation performed so far

The validation aspect is done in one specific plant in Italy to reduce the complexity. The target is one injection moulding machine operating in a controlled environment but still producing regular parts to be assembled in final products. The validation step consists in comparing the physical data to the behaviour model output and correctly detecting a breakdown.

Input data for the model is a multivariate time-series sampled every 500ms with 49 input features. Data is then grouped in batches of 15 minutes for model training. The features include:

- the electric current consumption of the machine,
- the temperature in each of the cavities of the mould,
- the velocity of the vibration of the bearing
- the acceleration of the vibration of the bearing.

The figure below illustrates the sample input data.

Figure 26: Multivariate Features by time

Preliminary results, illustrated below, indicate that the model predictions are in accordance with the real data. Given a day in which the machine had one critical alarm at around 14:15, the model correctly begins predicting this failure at 13:30. This is illustrated in the figure below, showing a gradual decrease in model output value p from 13:30 (p = 0.8642) until 14:30 (p = 0.0353). Once the failure is resolved and the machine status goes back to Normal, the output value computed by the model increases and stabilises itself around approximately 1.

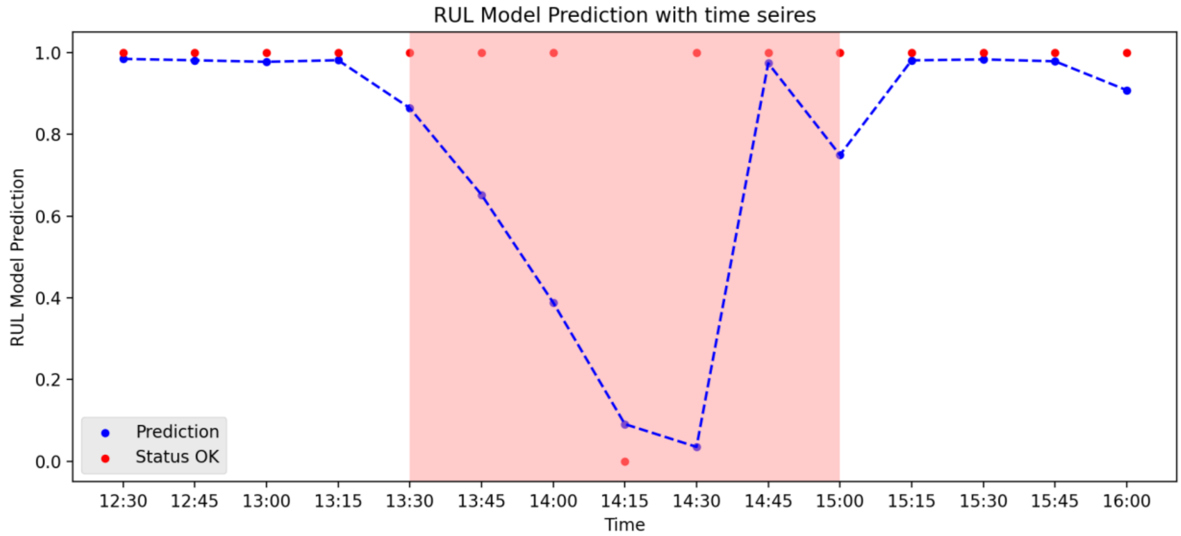


Figure 27: Model Prediction following Bearing Status

6 Conclusion

This deliverable reports the outcome of the achieved work within WP4 for 30 months to smartify or sensorize the systems related to the testbeds and data acquisition. In addition, D4.4 concentrates on presenting the second digital Twin version delivery for all the manufacturing testbeds.

To summarize the achieved results testbed per testbed:

Testbed 1 (BRI)

The **i**nitial version of a "virtual twin” (physics-based) model of a wind turbine yaw drive system has been improved with further content. Limitations have been observed in the simulation model due to the paucity of environmental data (wind speed, wind direction etc) available. In progress is an evaluation of predictive maintenance models to analyse WTG acceleration labelled data (ENSAM), Machine Learning predictors (classifiers and/or regressor as needed) based on topological timeseries descriptors will be developed to address this objective.

The data storage on the IoTwins cloud tier is using the following services: low-frequency (InfluxDB) and high-frequency (MinIO).

Additional predictive maintenance services from IoTwins platform (based on ML / AI algorithms) are under evaluation jointly with TB1 partners (ESI / ENSAM / KKWS).

Testbed 2 (FILL)

The initial version of the digital twin with the implementation of data processing and data collection of the machine tool has been extended with a machine learning model. The edge architecture to perform these steps is based on the IoTwins reference architecture and the data was transferred to the cloud (min.io instance) provided by IoTwins partner INFN. The first version of this model has been changed and extended to a neural network model. The initial dataset has been used to train the model using the IoTwins cloud resources and predict the timepoint a tool in the spindle enters the material. The model was containerized and integrated in the edge architecture. This set-up (IoTwins microservice architecture and IoTwins cloud resources) has been tested and validated for our machine in-house. In the remaining time of the project, the testbed will be enlarged. A test data set has been created on suitable machines to test the model. During the extension of the testbed, the results will be quantified, and adaptions will be done, where necessary.

Testbed 3 (ETXE)

The Edge analytics, from the sensor, the pre-processing, the online ML-based processing and insights is already working on the industrial embedded electronics called Aingura Insights. This opens the range of application in the next scenarios of the testbed: machine-tool into the Etxetar’s production facilities and operation in a complete manufacturing environment in an OEM`s power train manufacturing line. The system is now being tested and validated. However, as spindle head’s ball-bearing has a useful life outside the length of this project, a run-to-fail testbed is being used to estimate RUL under accelerated conditions. Computational bugs are being detected and solved by Aingura and BSC team.

The integration with IoTwins platform is on the first steps, we connectivity has been developed and tested. TB3 is now working to use IoTwins ML services to provide estimation on ball-bearing RUL from the Health Indexes coming from the Edge part in form of data stream each 15 seconds.

Tested 4 (GCL)

In this testbed we connected an injection moulding machine to a custom-made data-gathering system. The collected data sources include both sensors embedded on the machine by the manufacturer, as well as other sensing equipment installed and validated by GCL. We used various communication standards – OPC/UA, CODESYS, MODBUS, OPC/DA – as well as a dedicated middleware to integrate the information exposed by the machines and analyse the data. This tool incorporates hundreds of different communication drivers and can connect directly to the machine, exposing the normalized data to the outside. In addition, we devised a SaaS architecture to handle the complete data pipeline, from acquisition to storage. Intermediate processes include validation and pre-treatment.

The combination with IoTwins platform was completely tested and validated for our first machine. The collected data was used to train a Machine Learning model to estimate the remaining useful life (RUL) of a critical bearing in a Plastic injection moulding machine. This model will help to perform required preventing maintenance in advance to avoid any failures during production on the machine. The model runs on the Cineca HPC infrastructure. Every 15 minutes it receives fresh data from the machine and returns its prediction in the form of the Remaining Useful Life of the machine, expressed in hours.