D5.1

Architecture and Data framework

745625

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 745625

Deliverable No.	D5.1	Work Package No.	WP5	Task/s No.	Task 5.1 and T5.2
Work Package Title		Data Acquisition & Analytics Ecosystem			
Linked Task/s Title		T5.1 Architecture for Data Acquisition and Analytics O&M Ecosystem T5.2 Information Model/ Data Interoperability			
Status		Draft Final	(Draft/Draft Final/Final)		
Dissemination level		PU	(PU-Public, PP, RE-Restricted, CO-Confidential) (https://www.iprhelpdesk.eu/kb/522-which-are-different-levels-confidentiality)		
Due date deliverable		2018-09-28	Submission date		2018-09-26
Deliverable version		D5.1_ROMEO_Architecture&DataFramework_v5.3_PU.docx			

Document Contributors

Deliverable responsible	INDRA		
Contributors	**Organization**	**Reviewers**	**Organization**
Paloma Verdejo Herreras Catherine Murphy-O'Connor Daniel Rodríguez Alvarez Francisco José Gómez Miñambre Alfredo Pérez Jiménez	INDRA	Leonardo Casado Javier de la Peña	ZABALA
Metin Feridun Ioana Giurgiu	IBM	Igor Reinares	LAULAGUN
Beatriz Diaz Carles Viciano Marta Fernández	IBERDROLA	Robert Jones	IBERDROLA
Tim Fisher Ursula Smolka	RAMBOLL		
Maj-Brit Stroemdahl Flaskager	SIEMENS		
Pierre Bousseau Pierre Stéphan Rayhana Baghli Alexios Koltsidopoulos	EDF		
Adolf Burgstaller Herbert Gsenger	UPTIME		
Cristian Rodenas	ADWEN		
Ulrich Oertel	BACHMANN		

Document History

Version	Date	Comment
0.1	2017-10-24	TOC
0.2	2018-06-14	Updated TOC
1.0	2018-06-22	Teesside architecture section 3.3 updated
2.0	2018-07-16	Wikinger (section 3.1) and East Anglia I (section 3.2) updated
3.0	2018-08-07	Updated section 4
3.1	2018-08-10	Iberdrola comments taken into account, transformation document into public, updates according to meetings conducted week 6th-10th August
3.2	2018-08-23	Node#1 update in East Anglia I, updated section 3, Iberdrola's feedback and comments included (RJ 23/08/2018)
4.0	2018-08-23	Teesside Mechanical components update
4.1	2018-08-24	Updates according to meetings conducted week 20th-24th August
5.1	2018-09-10	Updates according to meetings conducted week 3th-7th September
5.2	2018-09-11	Minor changes in the format
5.3	2018-09-24	Laulagun and Zabala revision plus addition of conclusions section

Table of contents

List of Abbreviations

Abbreviation	Description
API	Application Programming Interface
CMS	Condition Monitoring System
CSV	Comma Separated Values
DDS	Data Distribution Service
DMZ	Demilitarized Zone
DTLS	Datagram Transport Layer Security
FTP	File Transfer Protocol
GMS	Grid Measuring Station
HMI	Human Machine Interface
HPPP	High Performance Park Pilot
HTTPS	Hypertext Transport Protocol Secure
MQ	Message Queue
ODBC	Open Data Base Connectivity
OMG	Object Management Group
O&M	Operating and Maintenance
OS	Operation System
PLC	Programmable Logic Controller
PLS	Position Lateral Security
SCADA	Supervisory Control And Data Acquisition
SFTP	Secure File Transfer Protocol
SHM	Structural Health Monitoring
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCM	Turbine Condition Monitoring
TP	Transition Piece
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UCC	Universal CMM Controller
UDP	User Datagram Protocol
WP	Work Package
WPS	Wind Power Supervisor
WS	Web Service
WTC	Wind Turbine Controller
WTG	Wind Turbine Generator

1. Executive Summary

This deliverable presents the ICT architectures of the 3 ROMEO demonstrators: Wikinger, East Anglia I and Teesside as well as the detail of each integrated component. These architectures must meet the requirements and specifications of the project [1]: optimization of the maintenance of wind power facilities, extension in life of turbines and reduction of the cost of power generation. For each architecture, component descriptions have been defined; from field hardware devices deployed in the field to high level end-user applications integrated in the demonstrator. Cybersecurity mechanisms have been also considered to be incorporated in different layers of the data acquisition and analytics infrastructure. Due to cybersecurity reasons, not all detail of protocols used in the low level interfaces of the pilots as well as system configuration has been included in this document. Nevertheless, the consortium has a more detailed D5.1 confidential deliverable where this information is available.

The data framework describes the data that will be collected from each of the wind farm demonstrators; the strategies for storing them in the IBM Cloud ecosystem; and the flow of data between the components within the ecosystem. The goal of this activity is to ensure the availability of data from partners to enable the computation of physical fault models and the predictive models in the IBM Cloud ecosystem.

2. Introduction

2.1. Purpose

The objective of this deliverable is to provide the results of the work conducted in task 5.1 Architecture for data acquisition and analytics O&M ecosystem and task 5.2 Information model data interoperability.

On the one hand, this deliverable presents the overall system and communication architecture of the 3 ROMEO demonstrators: Wikinger, East Anglia I and Teesside in order to know how the different components will interact to fulfill the project objectives. For each component, the use case that they fulfil has also been included, a description of the functionality, the inputs and outputs of the component and together with the input and output dependencies with other components available in the architecture. In addition, a physical architecture overview of each pilot is presented identifying the different hardware that is needed to be connected for the data acquisition as well as to host the models, algorithms and high-level tools involved in each demonstrator. Cybersecurity mechanisms have been also considered to be incorporated in different layers of the data acquisition and analytics infrastructure. Moreover, all data exchange interfaces between components to be developed within WP5 were identified for each of the 3 demonstrators.

The data framework describes the data that will be collected from each of the wind farm demonstrators; the strategies for storing them in the IBM Cloud ecosystem; and the flow of data between the components within the ecosystem. The goal of this activity is to ensure the availability of data from partners to enable the computation of physical fault models and the predictive models in the IBM Cloud ecosystem.

2.2. Relation to other project documents

1. D1.1 Project requirements specification
2. D1.4 Threshold values to be used for WT monitoring purposes, based on existing fleet data for Siemens turbine
3. D3.1 Physical approach solutions as a DLL (Diagnosis +Prognosis)/ CM product based on running design

3. ROMEO architectures

The objective of this chapter is to provide the overall system and communication architecture of the 3 ROMEO demonstrators: Wikinger, East Anglia I and Teesside in order to know how the different components will interact to fulfill the project objectives and specifications [1]: optimization of the maintenance of wind power facilities, extension in life of turbines and reduction of the cost of power generation.

The components integrated in each architecture, that are not part of the AS-IS architecture of the demonstrator, are later developed in different work packages depending on the nature of the component: Diagnosis and Prognosis models in WP3, Structural Condition Monitoring models in WP4, Data acquisition and Analytics infrastructure in WP5, and O&M Information Management Platform in WP6.

The architecture of each demonstrator has been defined after the analysis of the systems, models and algorithms participating in each layer taking into account the integration mechanisms and protocols that were available for the components as well as the nature of the data to be integrated: real-time or historical data. The different layers considered in the architecture are as follows:

- Field Hardware Devices/ Communication Head-End layer: Sensor data acquisition layer which encompasses the various sensors and onshore data acquisition systems
- Utility Application Control and Management: In charge of the wind farm monitoring and maintenance at utility level
- Real Time Integration layer: which includes the smart gateway infrastructure, real-time communications manager and the open field message bus; allowing edge and distributed processing
- Cloud IoT platform layer: responsible for centrally managing all the interfaces within the ROMEO ecosystem. It is in charge of the data storage and the cloud analytics infrastructure
- Analytics: including WT offline failure models and structural condition monitoring
- Utility Application-End User layer: O&M Information management sytem and utility business applications

For each component, the use case that they fulfil has also been included, a description of the functionality, the inputs and outputs of the component and together with the input and output dependencies with other components available in the architecture. In addition, the last section of each demonstrator includes the physical architecture overview of the pilot identifying the different hardware that is needed to be connected for the data acquisition as well as to host the models, algorithms and high-level tools involved in each demonstrator.

In the cybersecurity section, several security mechanisms that can be incorporated in the data acquisition and analytics infrastructure, both on top of the real time platform and in the connections to the Cloud IoT infrastructure are presented.

Finally, all data exchange interfaces between components to be developed within WP5 were identified for each of the 3 demonstrators.

The architecture of the three demonstrators has been defined with the information that is already available from the pilots as well as from the analytics that are being defined in WP3 and WP4. During the progress of the project and in the development phase, some changes may be conducted for an optimal integration and performance of the whole ecosystem.

3.1. Wikinger architecture

The first site selected in ROMEO to prove the methodologies of the project is Wikinger Offshore wind farm.

The figure below shows Wikinger's architecture which contains the different systems that take part in this demonstrator from the offshore data acquisition from different sources, upload of the information to the onshore and back-office, the real time integration layer to handle large volumes of data in a secured, distributed and loosely coupled way, the IoT cloud analytics infrastructure that allows the data storage and the execution of the diagnosis and prognosis models up to the end-user layer to show and analyze the results.

This overall platform will provide desirable mechanisms to improve O&M strategies increasing prospects for life extension and ultimately lower the levelized cost of energy of offshore wind.

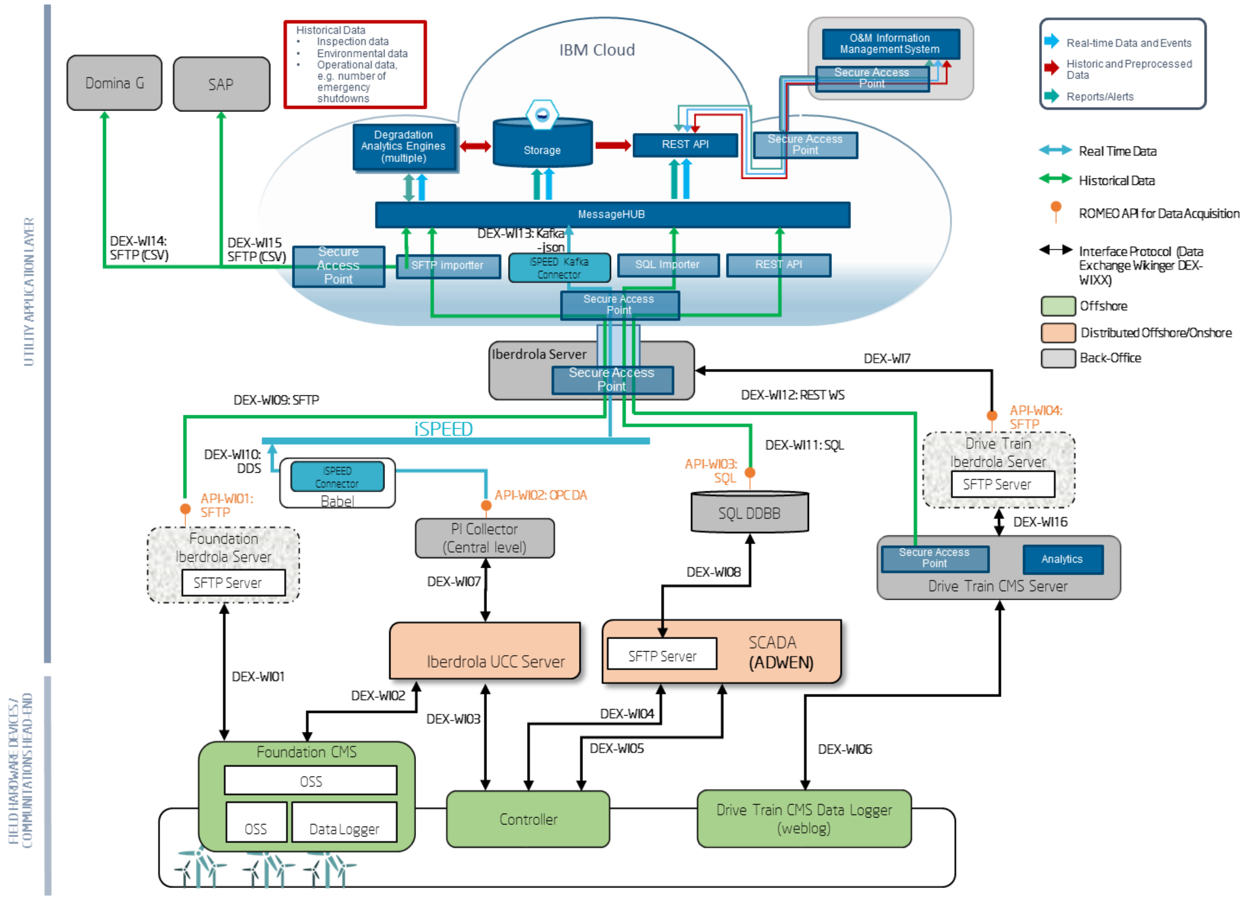


Figure 1: Wikinger architecture

The first layer we find in Wikinger's architecture is the **Field Hardware Devices/Communication Head End layer** which contains 3 head-end systems located offshore: Foundation CMS, Controller and Drive Train CMS Data Logger. These systems are in charge of monitoring and controlling the foundation vibrations, the components within the wind turbine and the WTGs drive trains respectively.

Connected through several protocols to the Field Hardware Devices/Communications Head-End layer there are five components at the **Utility Control and Management** layer, in charge of the windfarm control and management. These components are divided in two groups:

- Distributed offshore/onshore self-managed and self-balanced systems:
  - **Iberdrola UCC Server**, which acts as a communication gateway, connecting all devices in a wind farm such as wind turbines, meteorological masts, substations, regulators, etc. This server collects data from the Foundation CMS and retrieves data from the Controller.
  - **SCADA (ADWEN)**, is a software for supervising and monitoring all the devices in a wind farm such as wind turbines, meteorological masts, substations, etc. and it is responsible for the windfarm control and management. It is connected to the Controller.
- Back-office systems:
  - **Foundation Iberdrola Server**, which retrieves foundation vibration data from its mirror server at wind farm level.
  - **PI Collector**, based on a Data Archive in which time-series data are stored in tags and an Asset Framework in which the assets are organized in a hierarchy with all the relevant information regarding the asset. It is integrated with the Iberdrola UCC Server.
  - **Drive Train CMS Server** is part of the drive train Condition Monitoring System which is connected to the Data logger.

The components in both Field Hardware Devices/Head End and Utility Control and Management layers are existing systems at Iberdrola for the windfarm operation in the AS-IS architecture of Wikinger.

The Utility Control and Management layer is in charge of providing the real-time and historical data from the demonstrator needed in ROMEO to afterwards upload it to the IBM Cloud Platform. Thus, in this demonstrator there will be one real-time data acquisition API from the PI Collector, and three historical data acquisition APIs: SFTP from the Foundation Iberdrola Server through, SQL from the SCADA DDBB and SFTP from the Drive Train CMS Server.

The **Real Time Acquisition and Integration** layer bridges the gap between the real-time API available and the IoT Cloud platform. It provides the means for real-time connectivity through different components depending on the characteristics of the data sources. For the case of Wikinger, this layer is formed by two components provided by Indra: **Babel**, a real time communications manager that allows communication with different devices through different communication protocols and **iSPEED** a high performance distributed platform for data exchange based on DDS publish-subscribe mechanisms. Babel will be in charge of collecting real-time data from the OPC DA API of the PI Collector, and publish it into the DDS iSPEED Real Time Platform. The published data will be transferred to the IBM Cloud by a Kafka connector.

The **IBM Cloud** based IoT platform is responsible for centrally managing all the interfaces and components within the ROMEO ecosystem. It will be repository of all data to be used for predictive analytics and O&M management for the wind farms. In order to collect data securely, secure access points will be provided to allow communication between the IBM Cloud data consumers and the "on-premise", remote data sources. This platform will store the real-time data provided by iSPEED as well as the historical data directly imported from the Foundation Iberdrola Server, the Scada SQL DDBB and the Drive Train CMS Server. Moreover, for the case of Wikinger the IoT platform will allocate Adwen's physical models and IBM's predictive model suite from WP3 as well as Ramboll's damage detection algorithms from WP4.

IBM Cloud has three internal components:

- **Cloud Message HUB** that connects the internal and external sources to the rest of IBM Cloud components.
- **Cloud Data Store** to store raw, aggregated, context and analysis results data in a relational and no-SQL databases.
- **Cloud Analytics Engine**, which will provide a platform for data processing and machine learning to host WP3 and WP4 models.

There is another analytics module in Wikinger from WP3 for failure prediction which is the Bachmann's Diagnosis and Prognosis Suite. But in this case the models will be running on-premise and the results will be integrated to the Cloud through a REST API.

The last layer of the architecture is the **Utility Application - End User** layer which is connected to the IoT platform through a REST API and SFTP using also a Secure Access Point. This layer contains the WP6 O&M system, Domina G and SAP tools. Uptime's O&M Information Management System is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. Domina G manages all operational processes in Iberdrola Renewable business such as Asset integration, Meteorology forecast, Monitoring and Diagnosis, Reporting, and Documentation. This tool connects operational data from different resources. And finally SAP tool is an enterprise resource planning software used to cover all day-to-day processes of Iberdrola.

3.1.1 Component Description

In the following sections, the descriptions of the Wikinger architecture components are explained including the inputs and outputs of each component as well as the dependencies with the rest of the components in the architecture.

3.1.1.1 Field Hardware Devices/Communications Head-End

In Wikinger's architecture, there are three communication head-end systems which are in charge of monitoring and controlling the foundation vibrations, the components within the wind turbine and the WTGs drive trains respectively:

- The Foundation CMS monitors the vibrations of 7 WTGs and the Offshore Substation. All data loggers report their info to a duplicated central server placed at the substation.
- Wind Turbine **Controllers** are in charge of monitoring and controlling the components within a wind turbine.

- The **Drive Train CMS Data Logger (weblog)** system monitors the WTGs drive trains and reports vibration data acquired into a central server.

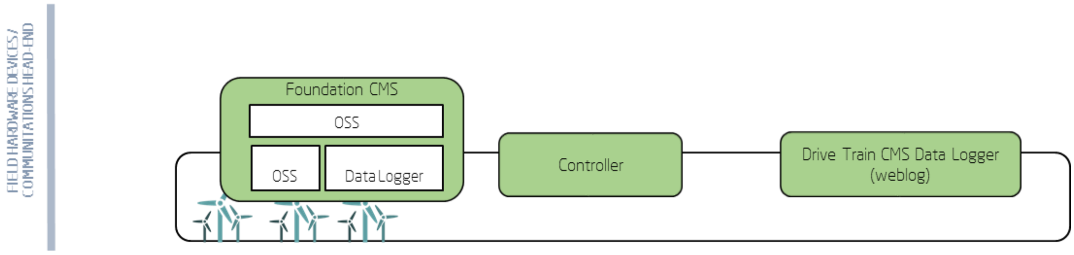


Figure 2: Figure Hardware Devices / Communications Head-End from Wikinger

The data from these three systems is sent to the Utility Control and Management layer components:

- The **Foundation CMS** is connected to the Foundation Iberdrola Server and to the Iberdrola UCC Server.
- **Controllers** are connected to **Adwen's SCADA** for the historical data and the real time data. They are also connected to the **Iberdrola UCC** Server which retrieves data.
- The **Drive Train CMS Data Logger** (weblog) is connected to the **Drive Train CMS Server**.

3.1.1.1.1 Foundation CMS

COMPONENT INFORMATION	
Title	Foundation CMS
Use Case	Condition Monitoring System for Foundations
Partner	IBERDROLA
COMPONENT DETAILS	
Description	This system monitors foundation vibrations of 7 WTGs and the Offshore Substation. All data loggers report their info communicating to a duplicated central server placed at the substation.
Programing Language	Does not apply
Inputs	Accelerometers, extensometric gauges, inclinometers and temperature sensor
Outputs	Sensor data
Integration Mechanisms/Connectors	The collected data stored in the server are available.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	
Dependencies with other components (Outputs)	Iberdrola UCC Server. Foundation Iberdrola Server.

3.1.1.1.2 Controller

COMPONENT INFORMATION	
Title	WTG Controllers
Use Case	Wind Turbine Supervisory Control and Data Acquisition.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	Components within the wind turbine are monitored and controlled by an individual local WTC (Wind Turbine Controller).
Programing Language	Does not apply.
Inputs	Data from each WTG.
Outputs	Data from 70 WTGs.
Integration Mechanisms/Connectors	These 70 controllers connected with the ADWEN SCADA for the historical data and for the real time data. They are connected to the Iberdrola UCC Server.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	
Dependencies with other components (Outputs)	ADWEN SCADA. Iberdrola UCC server.

3.1.1.1.3 Drive Train CMS Data Logger

COMPONENT INFORMATION	
Title	Drive Train CMS
Use Case	Condition Monitoring System for WTGs Drive Train.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	The system monitors the WTGs drive trains and reports vibration data acquired into a central server.
Programing Language	Does not apply.
Inputs	Vibration sensors.
Outputs	Drive train vibration data.
Integration Mechanisms/Connectors	The collected data are available through a Security Gateway Endpoint and are stored into IBM Cloud straightly.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	
Dependencies with other components (Outputs)	

3.1.1.2 Utility Control and Management

There are five components at the Utility Control and Management layer of Wikinger, connected to the Field Hardware Devices/Communications Head-End layer, that are in charge of providing the real-time and historical data from the demonstrator needed in ROMEO. These five components are the following:

- Distributed offshore/onshore self-managed and self-balanced systems:
  - **Iberdrola UCC Server**, which acts as a communication gateway, connecting all devices in a wind farm such as wind turbines, meteorological masts, substations, regulators, etc. This server collects data from Foundation CMS and retrieves data from the Controller. To transfer data from Iberdrola UCC Server to PI Collector (Central Level) there is an interface protocol.
  - **SCADA (ADWEN)** is a software for supervising and monitoring all the devices in a wind farm such as wind turbines, meteorological masts, substations, etc. and it is responsible for the windfarm control and management. It is connected to the controller through two channels: a real-time interface and a historical data connection. Scada data is also stored in a SQL database.
- Back-office systems:
  - **Foundation Iberdrola Server**, which retrieves foundation vibration data from its mirror server at wind farm level. This second server is part of the Foundations Condition Monitoring System.
  - **PI Collector**, located onshore, is based on a Data Archive in which time-series data are stored in tags and an Asset Framework in which the assets are organized in a hierarchy with all the relevant information regarding the asset. This data-base has a compression algorithm which allows the data base to perform more efficiently.
  - **Drive Train CMS Server** is part of the drive train Condition Monitoring System. It retrieves data from the data logger at wind farm level and also connects to the Driven Train CMS Data Logger (weblog).

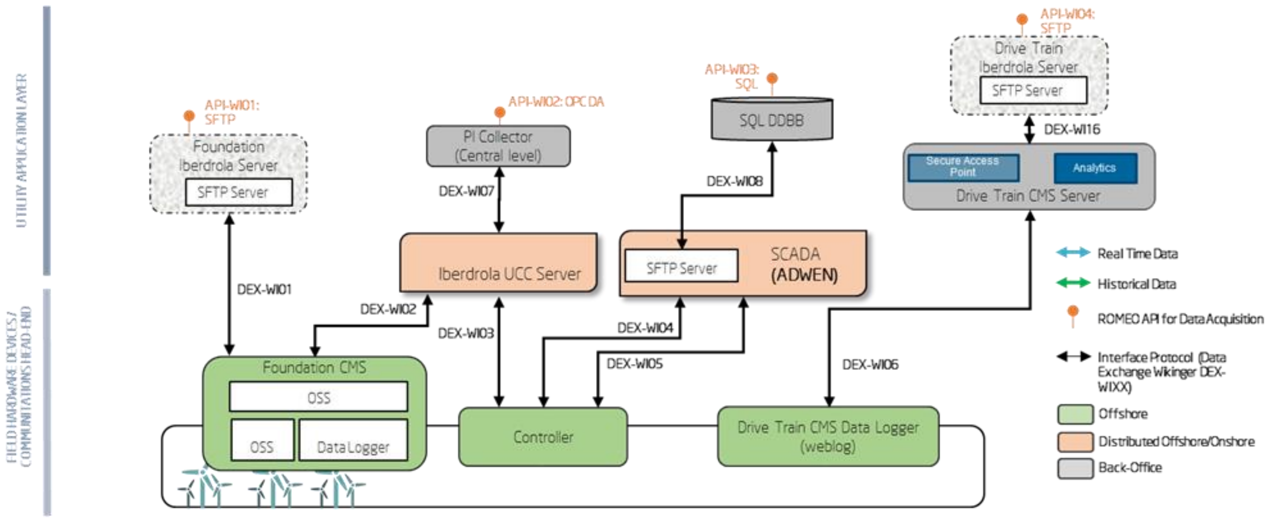


Figure 3: Utility Control and Management layer from Wikinger

In order to connect to the existing control and management ecosystem of Wikinger to afterwards upload the data into the IBM Cloud Platform, four APIS will be available to collect the Real Time and Historical data processed by the Utility Control and Management components.

On the one hand, the real-time data API (API-WI02) will enable a real-time connection through OPC DA to the PI Collector system.

On the other hand, the following three historical-data APIs will provide on a daily basis the foundation, Scada and drive train data:

- API-WI01: SFTP to collect data from Foundation Iberdrola Server
- API-WI03: SQL to collect historical data from the Scada SQL DDBB
- API-WI04: SFTP to collect historical data from the Drive Train CMS Server

3.1.1.2.1 Foundation Iberdrola Server

COMPONENT INFORMATION	
Title	Foundation Iberdrola Server
Use Case	Condition Monitoring Data Server for WTGs and ST Foundations
Partner	IBERDROLA
COMPONENT DETAILS	
Description	This server is placed on Iberdrola's network and retrieves data from its mirror server at wind farm level. This second server is part of the Foundations Condition Monitoring System.
Programing Language	Does not apply.
Inputs	Foundation vibration data.
Outputs	All information is available.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Foundation CMS.
Dependencies with other components (Outputs)	SFTP connection.

3.1.1.2.2 Iberdrola UCC Server

COMPONENT INFORMATION	
Title	Iberdrola UCC server
Use Case	Unit for Communication Concentration.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	This server acts as a communication gateway, connecting all devices in a wind farm such as wind turbines, meteorological masts, substations, regulators, etc. It retrieves data using different communication protocols implemented in-house developed drivers and concentrate and standardize it, storing, processing and serving the data to the control center or other management systems.
Programing Language	Does not apply.
Inputs	Data retrieved from the connected devices.
Outputs	Retrieved data from the connected devices and processed data as 10-min statistical data, alarm and events treatment, calculus, etc.

DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Wikinger: Controller and Foundation CMS Data logger.
Dependencies with other components (Outputs)	Wikinger: PI Collector (Central level).
Other Comments	It can also acts as a supervisory unit for monitoring and controlling the connected devices (SCADA).

3.1.1.2.3 PI Collector

COMPONENT INFORMATION	
Title	PI System.
Use Case	PI System is time-series data base which can be used as a real-time data exchange or a historical data repository which can be used as base for modelling construction. This system enables the possibility of managing large volumes of data and processes that data directly. Can be used as a substitute of the SCADA if the information of it is not available.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	PI System is based on a Data Archive in which time-series data are stored in tags and an Asset Framework in which the assets are organized in a hierarchy with all the relevant information regarding the asset. This data-base has a compression algorithm which allows the data base to perform more efficiently. There are many interfaces which connect the different Data Sources (from different vendors) to the Data Archive as well as other interfaces that allow connecting directly to the Data archive. Depending on the tools used for extracting the data from this data base, it may be necessary to use other ones.
Programing Language	Does not apply.
Inputs	Data Sources from the local SCADA, magnitudes coming from the wind turbine, wind farm, met mast and substation (active power, nacelle wind speed, generator speed, etc.). Usually, this is information is sent to the PI System via OPC Server.
Outputs	Indicators calculated and afterwards stored in a PI tag.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Wikinger: Iberdrola UCC Server.
Dependencies with other components (Outputs)	Wikinger: Babel through OPC-DA.

3.1.1.2.4 SCADA (ADWEN)

COMPONENT INFORMATION	
Title	ADWEN SCADA
Use Case	Supervisory Control and Data Acquisition.
Partner	IBERDROLA

COMPONENT DETAILS	
Description	It is software functionality for supervising and monitoring all the devices in a wind farm such as wind turbines, meteorological masts, substations, etc. It is responsible for the windfarm control and management. It retrieves data from the devices connected to the wind farm such as their state, alarms, warnings or events from devices, etc. It stores data and can generate reports. It is also responsible for connections to control centers or higher level management systems. In short, it is a communications gateway for incoming and outgoing wind farm data.
Programing Language	Does not apply
Inputs	Data provided by the controllers installed at wind turbine and data from the substation control system.
Outputs	Historical Data via server • 10-minute average file. • Operating mode WEC/WMS. • Counter WEC/WMS. • Parameter changes. • Traces: ○ One file with 1-second averages. ○ One file with 200-milliseconds averages ○ One file with 10-milliseconds averages.
Integration Mechanisms/Connectors	Interface with the Substation. Interface with CORE SCADA. Interface with Controllers.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	70 x WT Controller (real time data). Connection Controller.
Dependencies with other components (Outputs)	DB Connection.

3.1.1.2.5 Drive Train CMS Server

COMPONENT INFORMATION	
Title	Drive Train CMS Server
Use Case	Condition Monitoring Data Collector for WTGs Drive Train
Partner	IBERDROLA
COMPONENT DETAILS	
Description	This server is placed in Austria and it is part of the drive train Condition Monitoring System. It retrieves data from the data logger at wind farm level through a proprietary connection.
Programing Language	Does not apply.
Inputs	Drive train vibration data.
Outputs	All information is available using Web Script.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Drive Train CMS Data logger.

Dependencies with other components (Outputs)	SFTP connection to IBM Cloud through a Secure Gateway Endpoint.

3.1.1.3 Real Time Acquisition and Integration

The Real Time Acquisition and Integration layer bridges the gap between the real-time APIs available at the Utility Back-Office and the IBM Cloud. It provides the means for real-time connectivity through different components depending on the characteristics of the data sources. On the one hand it is able to handle the heterogeneity of real deployments where different technologies will have to coexist and interoperate in a transparent way, assuring the interoperability of existing control and sensing systems with additional sensing networks and the deployment of new WT monitoring and control systems. On the other hand, it provides the assembly of a real time data acquisition and processing platform capable of responding to the stringent needs of the WT subdomain, based on the novel edge computing paradigm. Therefore, this solution promotes the distribution of intelligence among nodes located at different levels through an inherently loosely coupled infrastructure.

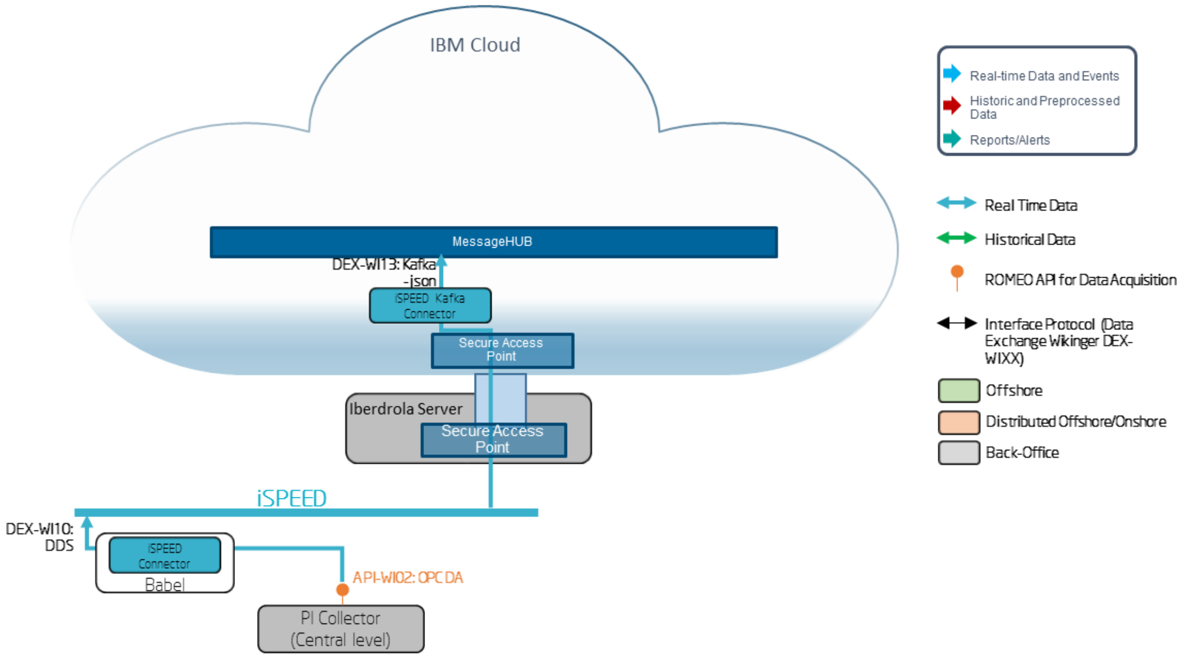


Figure 4: Real time acquisition and integration

This Real Time Acquisition and Integration layer in Wikinger is formed of two components provided by Indra:

- **Babel,** a real time communications manager that allows communication with different devices through different communication protocols, using only one interface. It has different features and components: high availability, remote access, administration console, tolerance against network drops, extension and communications gateway.
- **iSPEED** is a high performance distributed platform for data exchange based on DDS publish-subscribe mechanisms. It has the ability to handle large volumes of data from the network nodes

in a secured, distributed and loosely-coupled way. iSPEED is capable of exchanging data at low latencies and integrating all kind of devices and systems in a common infrastructure.

In Wikinger demonstrator, Babel will be in charge of collecting real-time data from the OPC DA API of the PI Collector and publish it into the DDS iSPEED Real Time Platform.

The DDS data published into iSPEED by Babell will be received by the iSPEED node that will be hosted in the IBM Cloud. This node will be responsible for transforming the DDS data from iSPEED into json and leave it in the Kafka Message Hub of the IBM Cloud for further processing on the IoT platform.

There will be several **Secure Access Points** to create a secure, persistent connection between Iberdrola infrastructure and the IBM Cloud. In this case, all data published into iSPEED by Babel will pass through the secure access point hosted at a Iberdrola server and will pass also through the IBM Cloud security access point before reaching the iSPEED node located in this IoT platform.

3.1.1.3.1 Babel

COMPONENT INFORMATION	
Title	Babel
Use Case	Babel is a real time communications manager that allows communication with different devices through different communication protocols, using only one interface, even though these devices belong to different networks. Babel is a concentrator of real time industrial protocols, that can be used as a simply protocol converter to interact with devices with different protocols, allowing interconnection with IEC 101, IEC 104, DNP3, ICCP, Modbus, OPC, iSPEED.
Partner	INDRA
COMPONENT DETAILS	
Description	The main features and components of BABEL are: • **High availability:** Babel allows establish a cluster configuration, with as many nodes as necessary. • **Remote access:** Babel has different lighter remote access mechanisms (FTP and SSH) that allow to modify the configuration, revise its state and download application logs. • **Administration Console:** Babel allows the monitoring in real time of the values of the measures of each one of the field devices configured. • **Extension:** Babel offers the possibility of modifying some default behaviors of the integrated protocols. • **Communications gateway:** Babel as a communications gateway solves the problem of interconnection between devices placed in different networks.
Programing Language	BABEL has been developed in Java.
Inputs	Wind farm magnitudes (Generator speed, wind direction, …). Events and alarms detected in the field.
Outputs	Wind farm magnitudes (Generator speed, wind direction, …). Events and alarms detected in the field.
Integration Mechanisms/Connectors	BABEL offers several communications interfaces with others systems through IEC 104, Modbus and ICCP.

DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	PI Collector (Central Server) [OPC DA]
Dependencies with other components (Outputs)	iSPEED (Field Message Bus)

3.1.1.3.2 iSPEED

COMPONENT INFORMATION	
Title	iSPEED-Field Message Bus.
Use Case	iSPEED is a high performance distributed platform for real-time data exchange. It has the ability to handle large volumes of data from the network nodes in a secured, distributed and loosely-coupled way. iSPEED is capable of managing millions of signals per second and integrating all kind of devices and systems in a common infrastructure.
Partner	INDRA
COMPONENT DETAILS	
Description	The main features and components of iSPEED are: • **Real Time Middleware** based on the Data Distribution Service (DDS) standard from the Open Management Group (OMG). • **Integrated Information Model** guaranteeing data interoperability (based on CIM and IEC 61850 standards). • **Quality of Service** for data delivery such as reliability, availability, liveliness, etc. • **Security**, ensuring the performance and safety requirements of Industrial IoT environments. Introduces a robust set of security capabilities such as authentication, encryption, access control and logging. • **Edge Computing**: integration with Complex Event Processing (CEP) engines allowing the business intelligence to be distributed in different network layers including low level nodes. • **Connector Adapters** for WebSockets and most used Message Broker systems such as JMS, MQTT and Apache Kafka. • **Routing service** enhancing network interoperability in broad and heterogeneous WAN. The exchange of data through iSPEED provides the following benefits: • Increase of productivity and efficiency in the management of information generated by various monitoring and control applications. • Efficient execution of processes, reducing the chances of error in data manipulation. • Quick and reliable exchange of information while ensuring the update of the latest data. • Distributed and centralized data processing, for large volume of data.
Programing Language	iSPEED has been developed in 3 different programing languages: Java, C++ and .NET in order to facilitate the integration with other systems.
Inputs	Wind farm magnitudes (Generator speed, wind direction). Events and alarms detected in the field. Warnings produced by Edge intelligent nodes (if applicable).
Outputs	Wind farm magnitudes (Generator speed, wind direction). Events and alarms detected in the field.

	Warnings produced by Edge intelligent nodes (if applicable).
Integration Mechanisms/Connectors	• iSPEED API available in Java, C++ and .NET (recommended option to guarantee quality of service). • Web Sockets server. • Message Broker Kafka Client. • Message Broker MQTT Client. • Message Broker JMS Client.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Babel via iSPEED API.
Dependencies with other components (Outputs)	IoT Cloud via Kafka connector.

3.1.1.4 IoT Platform

The IoT Platform is called "IBM Cloud". It is responsible for centrally managing all the interfaces and components within the ROMEO ecosystem. This platform will store the real-time data provided by iSPEED as well as the historical data directly imported from the Foundation Iberdrola Server, the Scada SQL DDBB and Drive Train CMS Server. IBM Cloud will host the statistical algorithms and damage detection for failure prediction models developed in WP3 and in WP4.

The results of the models will be provided to the O&M platform from Uptime as well as to Domina G and SAP tools from Iberdrola, which are systems located outside the IBM cloud platform.

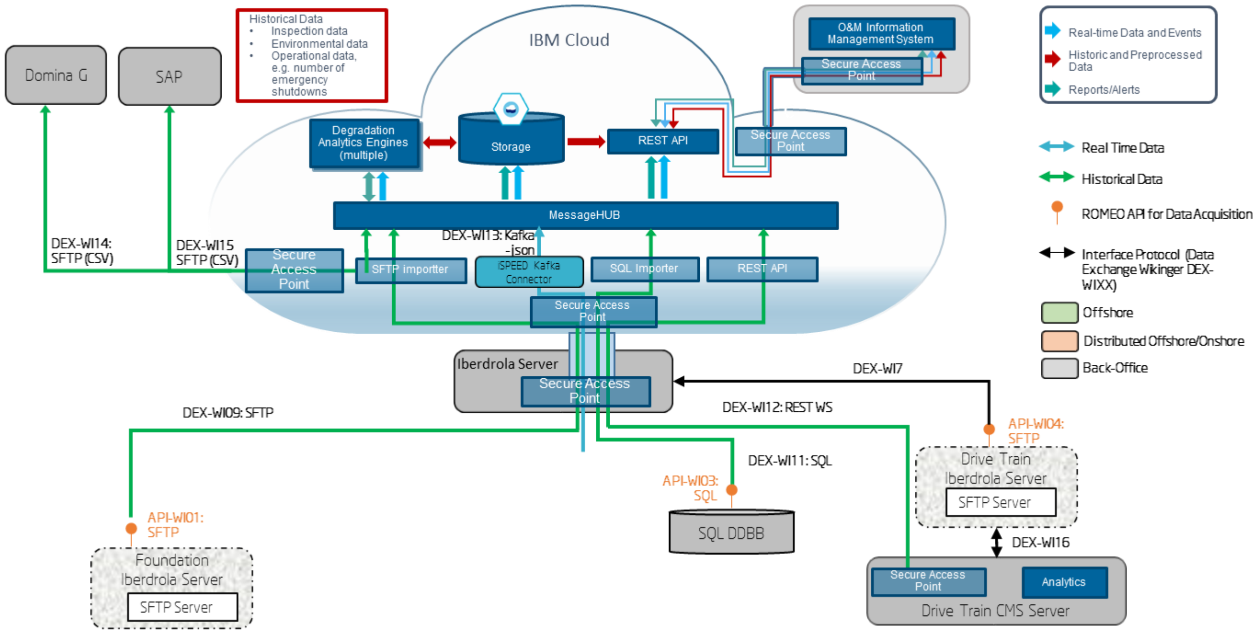


Figure 5: IBM Cloud

All data entering into the IBM Cloud will pass through a Secure Access Point in order to create a secure and persistent connection between the Iberdrola servers where the data is collected and the IBM Cloud. In addition, the end-user tools O&M, SAP and Domina G will also be connected to a secure access point to guarantee the data privacy.

The **IoT Platform** has three internal components:

- **Cloud Message HUB** that connects the internal and external sources to the rest of IBM Cloud components.
- **Cloud Data Store** to store raw, aggregated, context and analysis results data in a relational and no-SQL databases.
- **Cloud Analytics Engine**, which provides a platform for data processing and machine learning to host the WP3 and WP4 statistical and drive train, models of Wikinger.

3.1.1.4.1 Cloud Message HUB

COMPONENT INFORMATION	
Title	Cloud Message Hub
Use Case	Interconnect external data sources (e.g.,from iSPEED, via FTP or SQL) and internal data sources (e.g., analytics) to Cloud components.
Partner	IBM Research
COMPONENT DETAILS	
Description	The MessageHub is an internal component that connects the external and internal data sources to the Cloud components. Data coming from external sources (e.g., iSPEED, FTP, SQL) and used the Message Hub to distribute to data. It interconnects with edge components such as the database and the analytics components.
Programing Language	N/A
Inputs	External data coming from wind farms and other sources via iSPEED, FTP or SQL based sources; result and derived data from analytics.
Outputs	Received data routed to internal components, e.g., the data store and the analytics .
Integration Mechanisms/Connectors	Apache Kafka API
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	iSPEED Kafka Connector and other interfaces (FTP and SQL access to remote servers, REST interfaces) to pull in wind farm data.
Dependencies with other components (Outputs)	Cloud database instances, analytics components.

3.1.1.4.2 Cloud Data Store

COMPONENT INFORMATION	
Title	Cloud Data Store
Use Case	Store data collected from wind farms, including measurement data (e.g., SCADA, CMS, etc) and event/alert logs, as well as derived data and analytics results from WP3 activities.
Partner	IBM Research
COMPONENT DETAILS	

Description	The data storage component stores raw, aggregated, context, and analysis results data in a relational and no-SQL databases.
Programing Language	N/A
Inputs	Data (raw, derived, aggregated) from the MessageHub (from wind farms) and as well as data from analytics components (results, models).
Outputs	Responses to data queries.
Integration Mechanisms/Connectors	Database query language (SQL, REST)
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud MessageHub
Dependencies with other components (Outputs)	

3.1.1.4.3 Cloud Analytics Engine

COMPONENT INFORMATION	
Title	Cloud Analytics Engine
Use Case	Provide platform for data processing and machine learning to be used by WP3, based on data collected from each of the three wind farms, including and not limited to SCADA, CMS and other sources.
Partner	IBM Research
COMPONENT DETAILS	
Description	Data processing analytics engine providing machine learning libraries and data processing capabilities, e.g. Apache Spark. The exact design and the components are still to be defined and depend on the algorithms to be developed in WP3, as well as further requirements WP3 will impose on WP5.
Programing Language	e.g., Scala, Python, Java
Inputs	Raw or derived data from wind farms, machine learning algorithms. Data to be gathered from the central data store built by WP5.
Outputs	Device/refined data, analytics results.
Integration Mechanisms/Connectors	Integration mechanisms already available for component data exchange.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud data store, Cloud message hub.
Dependencies with other components (Outputs)	Cloud data store, Cloud message hub.

3.1.1.5 Analytics

In the Wikinger demonstrator there are four types of analytics developed in WP3 and WP4 that will be hosted in different locations in the architecture: three modules will be allocated in the Analytics Engines available at the IBM Cloud platform and the other suite will be hosted on premises.

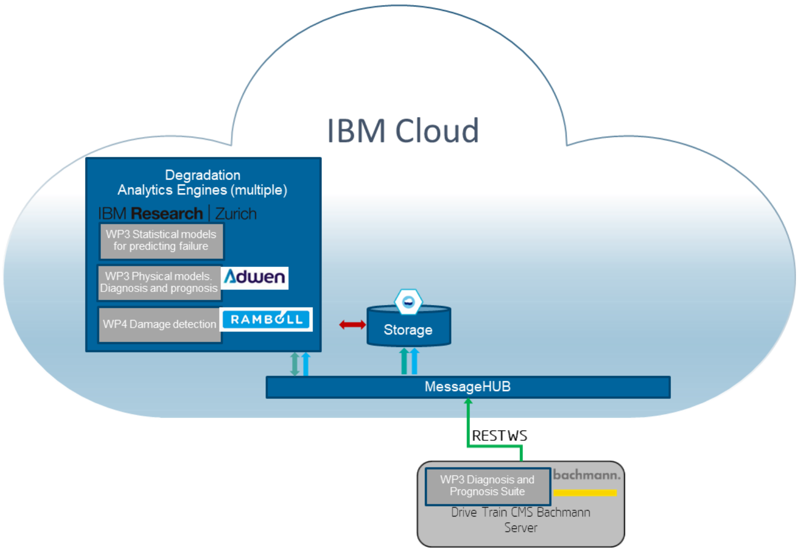


Figure 6: Wikinger Analytics Infraestructure

On the one had, the Analytics Degradation Engines of the IBM Cloud will host the following models for Wikinger:

- **Adwen-WP3-Physical module** (one per failure mode) that gets all the analog variables available with a buffer of data with timestamp that would yield as an output the status of detection of the failure mode, and the prognosis. until the failure progresses to a failed/alarm state on the eyes of the Supervisory system or the O&M procedures of the wind turbine.

  These models will use the Failure data set that contain raw data collections on all the analogic parameters, and as most as it is reasonable to be collected from the supervisory logic, distinguishing Normality and Abnormality Datasets.

- **IBM-WP3-Predictive Model Suite** that is software functionality for providing predictions of incoming failures for main components in a wind turbine, such as blade bearing, electrical drive train, mechanical drive train.

- **Ramboll-WP4 - Damage Detection** is an executable that checks in 10 minutes intervals continuous and automated data quality, extracts modal information, calculate statistics, damage indicator values and fatigue.

These models will be connected to the Data Store and Message Hub of the IoT platform in order to retrieve the data needed as well as to provide the algorithms results.

On the other hand, there will be in Wikinger demonstrator another analytics suit running on a server from outside the IBM Cloud:

- **Bachmann – WP3 - Diagnosis and Prognosis Suite** that is a software for failure prediction based on a FMSA covering the main failure modes of the mechanical drive train (main bearing, gear box). The inputs for these models are collected directly from the Drive Train CMS Data Logger and the results are provided to the IBM Cloud through a REST web service interface.

The results of the four analytics modules will be stored in the IBM Cloud and available to the Utility Application- End User systems.

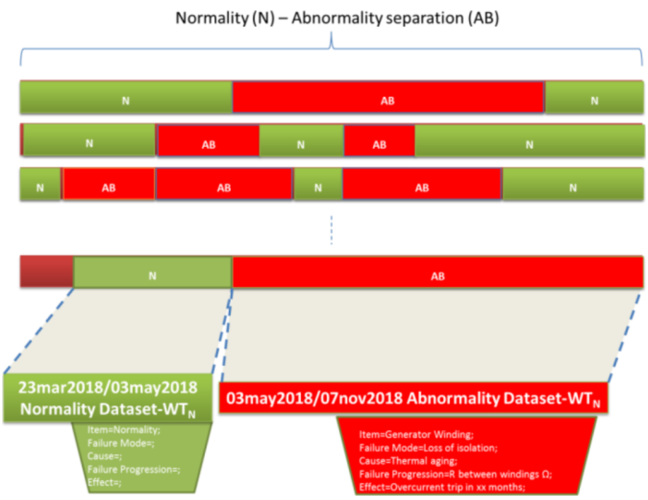
3.1.1.5.1 WP3-Adwen-Physical Module

COMPONENT INFORMATION	
Title	Physical module (library or run time) – One per failure mode.
Use Case	Diagnosis and Prognosis with the traditional physical approach (offline)
Partner	ADWEN GmbH
COMPONENT DETAILS	
Description	Function provided: Getting input of all the analogical variables available with a sufficient buffer of data with timestamp would yield as an output the status of detection of the failure mode, and the prognosis until the failure progresses to a failed/alarm state on the eyes of the Supervisory system or the O&M procedures of the wind turbine.
Programing Language	C or C++
Inputs	For the programming: Adwen's knowledge. For the running of the component: All the analogical parameters from the SCADA, with: • General: At least two year per wind turbine, buffered, if available. • Special cases: At least 10 year buffer if available (Only 10' average data from all the analogical parameters.
Outputs	Default output is diagnosis/prognosis: **Diagnosis:** Phase of failure (String). Associated confidence level (Double from 0 to 1). **Prognosis:** Time to next phase (Double; Units to be defined). Associated confidence level (Double from 0 to 1). **ID [will be yielded by demand. (e.g. module1.exe /id)] :** Each module shall be able to identify itself by failure mode detection module code. The information included shall be: • Code [failure mode] • Component • Software Version / Release • Windfarm • WTG
Integration Mechanisms/Connectors	Call of a function or executable (Typical OS functions) Pass off all the buffered analogical variables (300 approx.) coming from SCADA: • Format will be an array OR 'tab-delimited text files', with as much as values as it could be collected within the SCADA. (See Inputs)

	• First vector will be the time stamps (milliseconds; max resolution 10ms) • One WTG per run.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	IBM Cloud Data Store
Dependencies with other components (Outputs)	Task 3.3 related with the 5MW wind turbine from Adwen. Results will be stored in the IBM Cloud Data Storage
Other Comments	No comments.

3.1.1.5.2 WP3-Adwen-Failure Data Set

COMPONENT INFORMATION	
Title	Failure data set(s).
Use Case	Machine learning of normality and abnormality.
Partner	ADWEN GmbH
COMPONENT DETAILS	
Description	Failure Datasets contain raw data collections on all the analogic parameters, and as most as it is reasonable to be collected from the supervisory logic. Distinguishing Normality and Abnormality Datasets: Normality applies to normal working of the wind turbine (e.g: when the wind turbine is delivering 5MW). Abnormality applies to the failure cases described in the metadata that is delivered with the failure data set (e.g.: component, failure mode).
Programming Language	(Not a programming language) Encapsulation and/or encryption of data. Could be as simple as a CSV.
Inputs	Adwen interprets inputs as the inputs necessary for creating the Datasets: • Window conditions will be a consensus input (defined by Adwen and IBM). • Data from the wind turbine. • Normality/Abnormality definition.



	• Hand, hardcoded data: ○ Date, Wind turbine, time span. Normality / Abnormality. If Abnormality: Component, Failure Mode.
Outputs	The Dataset itself as Raw data + Tags or as an encapsulated class. Possibility of being read by itself or either via getter + Private Key. Eg.:public double. getAllDataInDataset(PrivateKey AdwensPrivateKey)
Integration Mechanisms/Connectors	Possibility of reading the data with encryptions (as per previous line)
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	No dependency from other component.
Dependencies with other components (Outputs)	Adwen Physical modules. Products in Task 3.2 and Task 3.3 related with the 5MW wind turbine from Adwen (if any) will depend on this type of component(s).
Other Comments	No comments.

3.1.1.5.3 WP3-Bachmann--Diagnosis and Prognosis Suite

COMPONENT INFORMATION	
Title	Bachmann Diagnosis and Prognosis Suite.
Use Case	Failure Mode and Symptom Analysis (FMSA) based diagnosis models for failure prediction.
Partner	BACHMANN Monitoring GmbH
COMPONENT DETAILS	
Description	The Bachmann Diagnosis and Prognosis Suite is a software for failure prediction based on a FMSA covering the main failure modes of the mechanical drive train (main bearing, gear box). Drive train failure modes cause specific symptoms within the analyzed vibration data recorded by the Condition Monitoring System (CMS). A comparison with historical measurements and environmental data will be used in order to identify abnormal behavior, which can be used to trigger alarms.
Programing Language	Java
Inputs	Vibration data and SCADA data measured by the CMS.
Outputs	Failure predictor for drive train components (main bearing, gear box) • RDS-PP code • FMSA prediction (0=no failure, 1=failure) • FMSA probability [0,100]% • FMSA severity [0: low, 5: high] • P-F interval (in weeks/months) • P-F interval / prognostic probability [0, 100]%
Integration Mechanisms/Connectors	Secure Interface (API). Rest Web Service
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Failure data sets including documented failure modes and severity classes for the drive train components (main bearing, gearbox) for training, test and verification from Adwen.

Dependencies with other components (Outputs)	Rest web service to provide the analytics results to the IBM Cloud

3.1.1.5.4 WP3-IBM-Predictive Model Suite

COMPONENT INFORMATION	
Title	IBM PREDICTIVE MODEL SUITE
Use Case	Statistical models for failure predictions.
Partner	IBM Research
COMPONENT DETAILS	
Description	It is software functionality for providing predictions of incoming failures for main components in a wind turbine, such as blade bearing, electrical drive train, mechanical drive train, etc. It uses data about various failure modes coupled with the physical models of components and historical sensor measurements to learn and predict what constitutes normal and abnormal behavior. It stores predictions as trigger alerts and can generate time-based reports.
Programing Language	Python
Inputs	Failure data sets, failure modes and physical model APIs provided for several WT components (blade bearing, electrical drive train, mechanical drive train, gearbox) by Adwen, EDF and Bachmann, as well as rules library provided by Siemens.
Outputs	Failure Predictions per WT Component Type • Binary prediction (0=no failure, 1=failure) per component and time window (e.g., upcoming x days, y weeks) • Failure risk with range [0,100]% per component and time window • Bucketed failure risk (e.g., levels = {not problematic, moderately problematic, highly problematic, critical) per component and time window • Remaining useful lifetime (e.g., expressed in days/weeks/months) per component
Integration Mechanisms/Connectors	Interface with the Secure Access Point / iSPEED Kafka connector (Cloud). Interface with the Storage Component (Cloud).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Data from IBM Cloud from Adwen, EDF, Bachmann and Siemens.
Dependencies with other components (Outputs)	Model result storage DB in IBM Cloud.

3.1.1.5.5 WP4-Ramboll--Damage Detection

COMPONENT INFORMATION	
Title	Ramboll Damage Detection.
Use Case	Continuous support structure (WTG and Substation jacket) damage detection and fatigue calculation.
Partner	RAMBOLL

COMPONENT DETAILS	
Description	• Ramboll Damage Detection is delivered as executable for windows machines. • Continuous and automated data quality checks in 10 minute intervals. • Extraction of modal information. • Calculation of statistics, damage indicator values and fatigue.
Programing Language	Python/Matlab
Inputs	SCADA data WK64, CMS data of WK64, CMS data of substation, met ocean data from windpark..
Outputs	Indicator signals for damage and fatigue.
Integration Mechanisms/Connectors	None. Specific adaptors need to be programmed for processing the data.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	IBM Cloud through DB and Message Hub
Dependencies with other components (Outputs)	IBM Cloud through DB and Message Hub
Other Comments	No comments

3.1.1.6 Utility Application Layer - End User

For the case of the Wikinger demonstrator, there will be three components in the Utility Application-End User Layer to collect the results of the analytics models available at the IBM Cloud:

- One component from Uptime: **O&M Information Management Platform**
- Two components from Iberdrola: **Domina G** and **SAP**

The **O&M platform** will be connected to the IoT platform through a REST API using also a Secure Access Point to guarantee the security of the information exchanged. This Information Management System is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. This platform will be fed with the results of the analytics embedded in the IoT platform and will provide KPI's and metrics to WP8 for assessing and quantifying the impact of the overall system on cost reduction potential.

Iberdrola's **Domina G** tool connects operational data from different resources: SCADA's, PI's, CORE, MeteoFlow, Meters, manual data entered by O&M staff and provides raw data and calculated information such as global indicator. It will be connected to the IoT platform through SFTP protocol.

SAP is an enterprise resource planning software used to cover all day-to-day processes of Iberdrola. This database server system stores and retrieves data from the separate modules to provide real-time information. SAP will use the SFTP protocol to connect to the IBM Cloud.

In addition, SAP and DOMINA G will send information to the IBM cloud such as work orders, alerts and other relevant information about the status of the wind turbines that would make easy to understand their behavior.

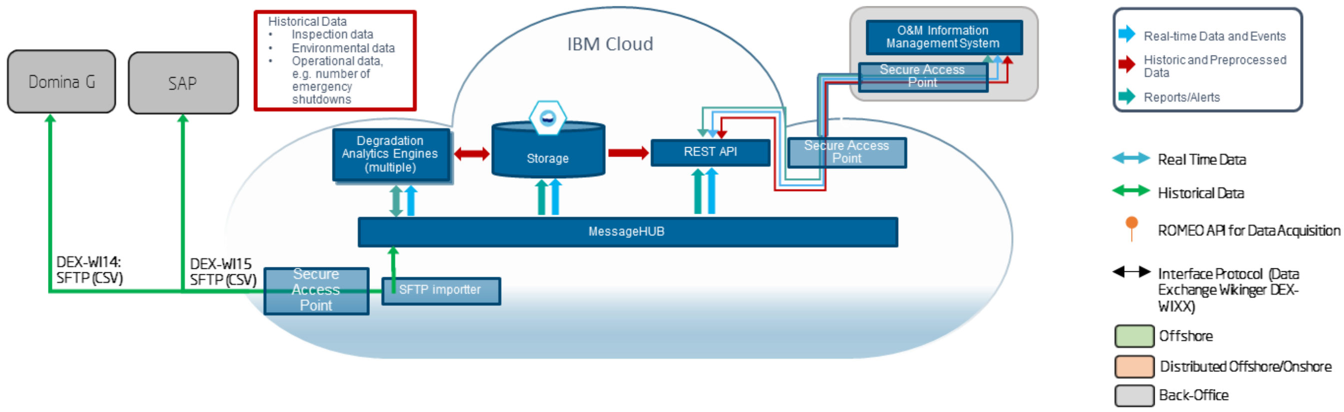


Figure 7: Utility Application Layer - End User from Wikinger

3.1.1.6.1 O&M Information Management Platform

COMPONENT INFORMATION	
Title	O&M Information Management Platform aka Uptime HARVEST
Use Case	Uptime HARVEST is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. As a web application it provides a central point of access to analyze pre-aggregated data and generate specific, actionable information that can be fed into asset management, allowing effective feedback from the field for continuous improvement.
Partner	UPTIME Engineering
COMPONENT DETAILS	
Description	Features of Uptime HARVEST: • **Integration** with IoT platform. • **Effective visualization and reporting** of post processed data. • **Aggregation of data** to ensure information generated by the IoT platform is converted into effective advisory results, based on business-critical processes (defined in WP1) in context of operations and asset management considering needs of pilot tests prepared during WP7. • **Support of maintenance task management.** • **Support of knowledge management.** • **Automation of business processes** to minimise cost. • **KPI's and metrics** to WP8 for assessing and quantifying the impact of the overall system on cost reduction potential.
Programing Language	C#, T-SQL
Inputs	Events generated by IoT Cloud. Pre-aggregated time series information from IoT Cloud.
Outputs	Graphical user interface. No other outputs specified yet.
Integration Mechanisms/Connectors	RabbitMQ already available for integration (message format to be specified). REST API to be developed to integrate with IoT Cloud.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	IoT Cloud. NO integration with iSPEED field message bus.

Dependencies with other components (Outputs)	WP7 (Iberdrola, Testing): No APIs to support automatic testing specified yet. WP8 (Cranfield University, Impact Assessement): No APIs to export data specified yet.

3.1.1.6.2 Domina G

COMPONENT INFORMATION	
Title	Domina-G
Use Case	In-house tool to manage all operational processes in Iberdrola Renewable business such as: Asset integration, Meteorology forecast, Monitoring and Diagnosis, Reporting, and Documentation.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	The tool connects operational data from different resources: SCADA's, PI's, CORE, MeteoFlow, Meters, manual data entered by O&M staff and provides raw data and calculated information such as global indicator (standard availability, effective production, etc) through graphic views, boards, charts and reports.
Programing Language	JAVA
Inputs	Data retrieved from the connected devices and manually entered in the modules of the tool.
Outputs	Retrieved data from the connected devices and calculated data through exportable Excel files.
Integration Mechanisms/Connectors	ORACLE
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	IBM Cloud Message Hub
Dependencies with other components (Outputs)	IBM Cloud Message Hub
Other Comments	Other information related to the component

3.1.1.6.3 SAP

COMPONENT INFORMATION	
Title	SAP
Use Case	Enterprise resource planning software used to cover all day-to-day processes of Iberdrola such as Financial Accounting (FI), Controlling (CO), Asset Accounting (AA), Sales & Distribution (SD), Material Management (MM), Product Planning (PP), Quality Management (QM), Project System (PS), Plant Maintenance (PM), Human Resources (HR) and Corporate Services.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	As a database server, SAP system stores and retrieves data from the separate modules to provide real-time information. Also, the NetWeaver

	platform allows the development of business applications such as customized web servers and the SAP Business Warehouse (SAP BW) runs reports in an agile way.
Programing Language	ABAP/4, C, C++
Inputs	Data entered in the modules in a structured and hierarchical way.
Outputs	Retrieve data from all SAP modules through list editing queries exportable to Excel files.
Integration Mechanisms/Connectors	TCP/IP, RFC, CPI-C, SQL, ODBC, OLE/DDE, X.400/X.500, MAPI, EDI, CAD, JAVA.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	IBM Cloud Message Hub
Dependencies with other components (Outputs)	IBM Cloud Message Hub

3.1.2 Physical Architecture Overview

The tentative IT infrastructure that is needed in Wikinger to host the different ROMEO components which take part in this demonstrator is depicted in the figure below.

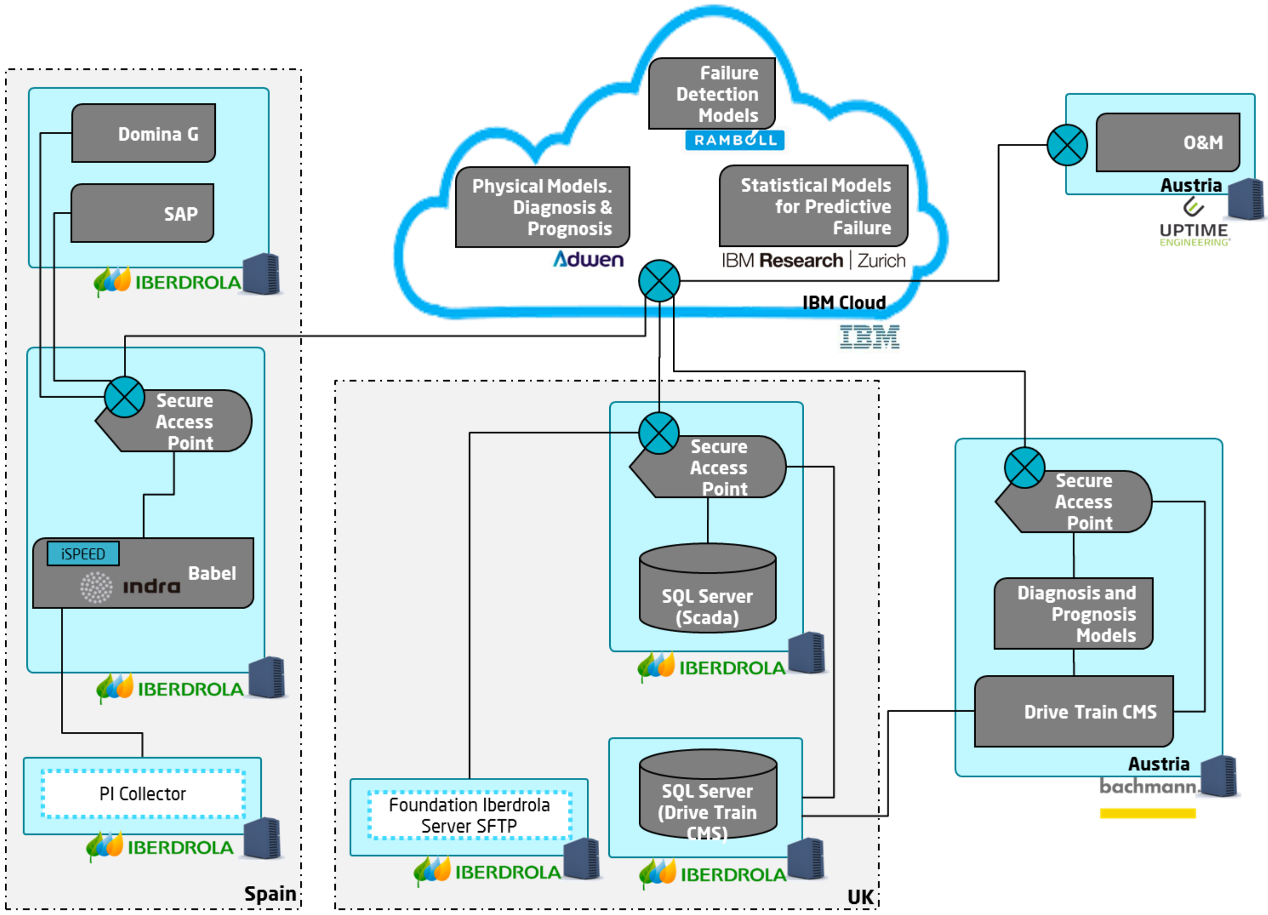


Figure 8: Physical Wikinger Architecture overview

First, there is a **Secure Access Point** to create a secure and persistent connection between all the servers that will be connected to the IBM Cloud. In this case, the data will be exchanged with the Cloud platform by servers located in different places: Spain, UK and Austria.

In each location there will be one secure access point that will be connected to all the servers hosting ROMEO systems in that location. These servers in different countries are from different owners:

- In Spain there will be three Iberdrola Servers hosting the following systems:
  - Secure Access Point and Babel System
  - PI Collector system
  - Domina G and SAP systems
- In UK there will be three Iberdrola Servers to allocate:
  - Secure Access Point and SQL Server from Scada
  - Foundation Iberdrola Server SFTP
  - SQL Server (Drive Train CMS) which is also connected to the Drive Train CMS located in Feldkirch, Austria
- In Austria there will be in Bachmann server to host
  - Secure Access Point, Drive Train CMS and Diagnosis and Prognosis Models

- In another location in Austria there will be another Uptime server to allocate:
    - O&M Information Management System and a Secure Access Point

During the implementation and deployment of the whole infrastructure (WP7), some changes may be needed in the hosting of the different tools depending on the performance of the whole ecosystem.

3.2. East Anglia I architecture

The second pilot selected in ROMEO to prove the methodologies of the project is East Anglia I offshore wind farm.

The figure below shows East Anglia I architecture which contains the different components that take part in this demonstrator from the offshore data acquisition from different sources, upload of the information to the onshore and back-office, the real time integration and processing layer to handle large volumes of data in a secured, distributed and loosely coupled way, the IoT cloud analytics infrastructure that allows the data storage and the execution of the diagnosis and prognosis models up to the end-user layer to show and analyze the results.

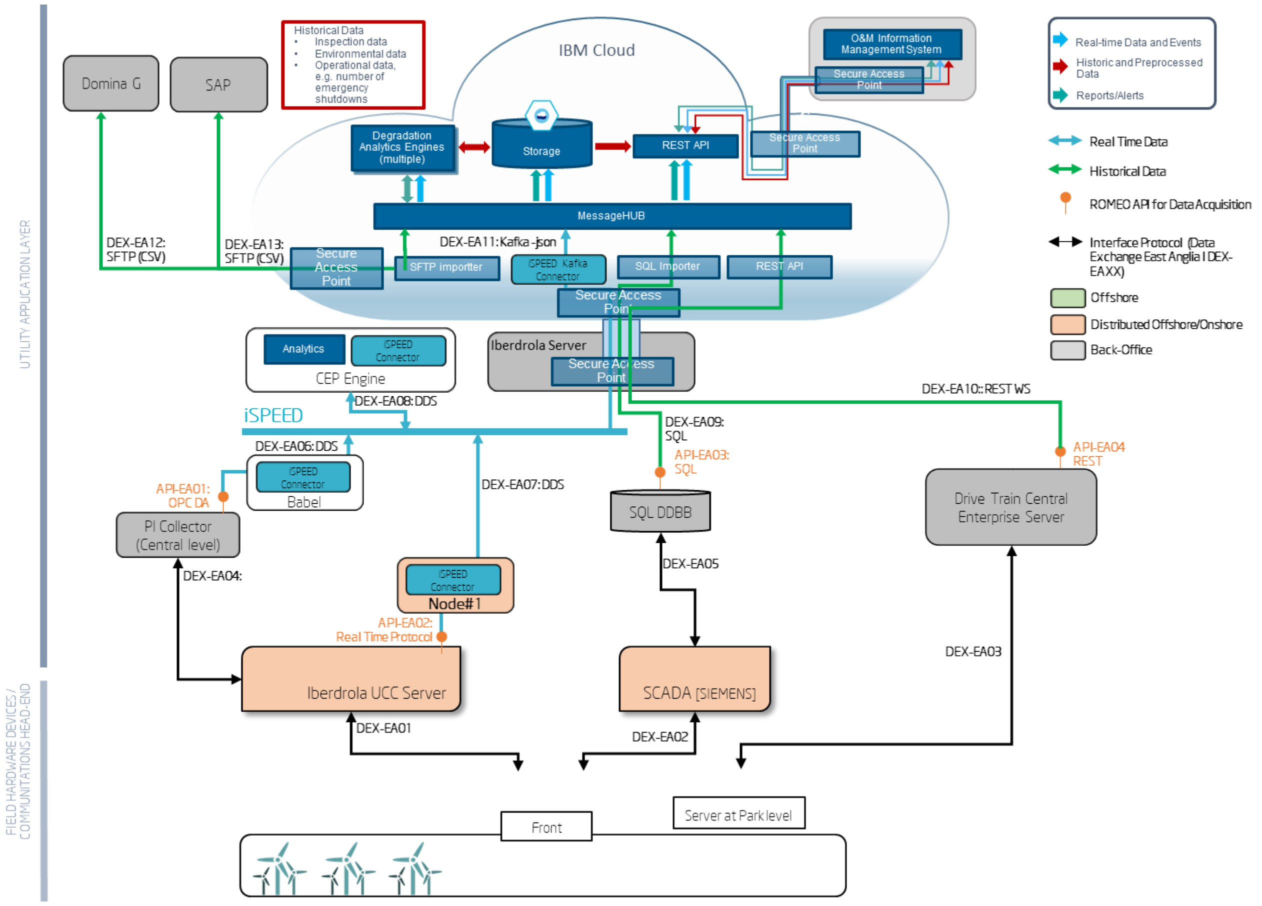


Figure 9: East Anglia I architecture

The first layer we find in the East Anglia I Architecture is the **Field Hardware Devices/Communication Head End layer** which contains 2 head-end systems located offshore that are in charge of collecting the data from the field:

- The **Data Control Siemens** is a sub-system of the Siemens Gamesa Renewable Energy (SGRE) SCADA system that monitors and controls the components within the wind turbine.
- The **Drive Train CMS** system monitors the WTGs drive trains and reports vibration data acquired into a central server placed in the windfarm.

Connected through several protocols to the Field Hardware Devices/Communications Head-End layer there are four components at the **Utility Control and Management layer** in charge of the windfarm control and management. These components are divided in two groups:

- Distributed offshore/onshore self-managed and self-balanced systems:
  - **Iberdrola UCC Server** connects to the Data Control – Siemens, and acts as a communication gateway, connecting connect all devices in a wind farm such as wind turbines, meteorological masts, substations, regulators.
  - **SCADA (Siemens)**, is a system located offshore and has different parts: Wind Park Supervisor (WPS) – SCADA used for supervision, data acquisition, control and reporting of wind farm performance; High Performance Park Pilot (HPPP) – Used for enhanced control of the wind farm and regulatory functions, Wind Turbine Controllers (WTCs) and Grid Measuring Station (GMS) that provides the SCADA system with accurate information about the current grid connection status.
- Back-office systems:
  - **PI Collector**, connects to the Iberdrola UCC Server, and is based on a Data Archive in which time-series data are stored in tags and an Asset Framework in which the assets are organized in a hierarchy with all the relevant information regarding the asset.
  - **Drive Train Central Enterprise Server** retrieves data from its mirror server at wind farm level and is part of the Drive Train Condition Monitoring System.

The components in both Field Hardware Devices/Head End and Utility Control and Management layers are existing systems at Iberdrola for the windfarm operation in the AS-IS architecture of East Anglia I.

Similar to Wikinger, the Utility Control and Management layer in East Anglia I is in charge of providing the real-time and historical data from the demonstrator needed in ROMEO to afterwards upload it to the IBM Cloud Platform. But in this case, there will be two real-time data acquisition APIs: one from the Iberdrola UCC server and the other one from the PI Collector; and two historical data acquisition APIs: SQL from the Scada DDBB and REST web service from the Drive Train Central Enterprise Server.

The **Real Time Acquisition, Integration and Processing layer** bridges the gap between the real-time APIs available at the Utility Back-Office and the IBM Cloud. It provides the means for real-time connectivity through different components depending on the characteristics of the data sources and the assembly of a real time data acquisition and processing platform capable of responding to the stringent needs of the WT subdomain, based on the novel edge computing paradigm. This layer is formed of four components provided by Indra: **Node#1,** a smart gateway prepared to apply on-site analysis techniques, capable of simultaneously processing multiple complex events, guaranteeing communication with multiple devices and sensors; **Babel,** a real time communications manager that allows communication with different devices through different communication protocols; **iSPEED,** a high performance distributed platform for data exchange based on DDS publish-subscribe mechanisms; **CEP Engines** where events and patterns will be automatically identified in order to detect anomalies in the real-time data available in the iSPEED platform.

In East Anglia I demonstrator Node#1 smart gateway will be deployed in the field to collect the Iberdrola UCC server real-time data and perform threshold rules analytics in the edge before uploading the information in the IBM Cloud through the iSPEED Real Time platform. Babel will be in charge of collecting real-time data from the OPC DA API of the PI Collector and publish it into iSPEED. On top of this platform there will be CEP engines with loaded patterns analysing the real-time data available in iSPEED. The results will be published in iSPEED and will be transferred to IBM Cloud by a Kafka connector.

The **IBM Cloud** based IoT platform is responsible for centrally managing all the interfaces and components within the ROMEO ecosystem. It will be repository of all data to be used for predictive analytics and O&M management for the wind farms. In order to collect data securely, secure access points will be provided to allow communication between the IBM Cloud data consumers and the "on-premise", remote data sources. This platform will store the real-time data provided by iSPEED as well as the historical data directly imported from the the Scada SQL DDBB and the Drive Train Central Enterprise Server. Moreover, for the case of East Anglia I, WP3 statistical models for failure predictions developed in WP3.

IBM Cloud has three internal components:

- **Cloud Message HUB** that connects the internal and external sources to the rest of IBM Cloud components.
- **Cloud Data Store** to store raw, aggregated, context and analysis results data in a relational and no-SQL databases.
- **Cloud Analytics Engine**, which will provide a platform for data processing and machine learning to host WP3 statistical models.

The last layer of the architecture is the **Utility Application - End User** layer which is connected to the IoT platform through a REST API and a SFTP interface using also a Secure Access Point. This layer contains the WP6 O&M system, Domina G and SAP tools. Uptime's O&M Information Management System is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. Domina G manages all operational processes in Iberdrola Renewable business such as Asset integration, Meteorology forecast, Monitoring and Diagnosis, Reporting, and Documentation.tool connects operational data from different resources. And finally SAP tool is an enterprise resource planning software used to cover all day-to-day processes of Iberdrola.

3.2.1 Component Description

In the following sections, the descriptions of the East Anglia I architecture components are explained including the inputs and outputs of each component as well as the dependencies with the rest of the components in the architecture.

3.2.1.1 Field Hardware Devices/Communications Head-End

In East Anglia I architecture, there are two offshore communication head-end systems, the Data Control from Siemens and the Drive Train CMS which are in charge of collecting the data from the field.

- The **Data Control Siemens** is a sub-system of the Siemens Gamesa Renewable Energy (SGRE) SCADA system that monitors and controls the components within the wind turbine by the individual local WTC (Wind Turbine Controller) which is capable of operating independently of the SCADA system.
- The **Drive Train CMS** system monitors the WTGs drive trains and reports vibration data acquired into a central server placed in the windfarm.

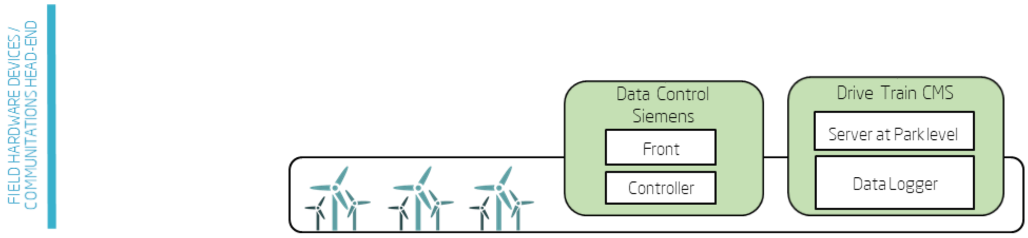


Figure 10: Filed Hardware Devices / Communications Head – End from East Anglia I

The data from these two systems is sent to the **Iberdrola UCC Server**, the **SCADA (Siemens)** and the **Drive Train Central Enterprise Server** components, included in the Utility Control and Management layer, through different protocols:

- Data Control Siemens exchanges data with the Iberdrola UCC Server.
- Drive Train CMS exchanges data with the Drive Train Central Enterprise Server.

3.2.1.1.1 Data Control Siemens

COMPONENT INFORMATION	
Title	Data Control SIEMENS
Use Case	Wind Turbine Supervisory Control and Data Acquisition.

Partner	IBERDROLA
COMPONENT DETAILS	
Description	This is a sub-system of the Siemens Gamesa Renewable Energy (SGRE) SCADA system. Components within the wind turbine are monitored and controlled by the individual local WTC (Wind Turbine Controller). The WTC is capable of operating independently of the SCADA system and turbine operation can continue autonomously in case of e.g. damage to communication cables. A turbine communication gateway placed at the tower base handles the interface between the WTC and the WPS server. Data recorded in the turbine is stored here temporarily so in event that the communication to the WPS server is interrupted it will not be lost and can be transferred to the WPS server when communication is restored.
Programing Language	Does not apply.
Inputs	Data from each WTG.
Outputs	Data from 102 WTGs.
Integration Mechanisms/Connectors	WTCs are connected to the SCADA system.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	
Dependencies with other components (Outputs)	Siemens SCADA. Iberdrola UCC server.

3.2.1.1.2 Drive Train CMS

COMPONENT INFORMATION	
Title	Drive Train CMS
Use Case	Condition Monitoring System for WTGs Drive Train.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	The system monitors the WTGs drive trains and reports vibration data acquired into a central server placed in the windfarm.
Programing Language	Does not apply.
Inputs	Vibration sensors.
Outputs	Drive train vibration data.
Integration Mechanisms/Connectors	The collected data are available and stored in a central enterprise server placed at Iberdrola´s network.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	
Dependencies with other components (Outputs)	Drive Train Central Enterprise Server.

3.2.1.2 Utility Control and Management

There are four components at the Utility Control and Management layer of East Anglia I, connected to the Field Hardware Devices/Communications Head-End layer, that are in charge of providing the real-time and historical data from the demonstrator needed in ROMEO. These four components are as follows:

- Distributed offshore/onshore self-managed and self-balanced systems:
  - **Iberdrola UCC Server** connects to the Data Control – Siemens, located offshore and acts as a communication gateway, connecting all devices in a wind farm such as wind turbines, meteorological masts, substations, regulators.
  - **SCADA (Siemens),** is a system located offshore and has different parts: Wind Park Supervisor (WPS) – SCADA used for supervision, data acquisition, control and reporting of wind farm performance, High Performance Park Pilot (HPPP) – Used for enhanced control of the wind farm and regulatory functions, Wind Turbine Controllers (WTCs) and Grid Measuring Station (GMS) that provides the SCADA system with accurate information about the current grid connection status.
- Back-office systems:
  - **PI Collector,** located onshore that connects to Iberdrola UCC Server, and is based on a Data Archive in which time-series data are stored in tags and an Asset Framework in which the assets are organized in a hierarchy with all the relevant information regarding the asset. Data Sources from the local SCADA, magnitudes coming from the wind turbine, wind farm, met mast and substation (active power, nacelle wind speed, generator speed, etc.). Usually, this information is sent to the PI System.
  - **Drive Train Central Enterprise Server** retrieves data from its mirror server at wind farm level and is part of the Drive Train Condition Monitoring System.

Four APIS will be available to collect and process the Real Time and Historical data managed by these four components in order to afterwards upload it into the IBM Cloud.

On the one hand, the real-time data APIs are the following:

- API-EA01: OPC DA to collect data from PI Collector
- API-EA02: Real time communication protocol to collect data from Iberdrola UCC Server

On the other hand, the historical data APIs are:

- API-EA03: SQL to collect historical data of SCADA from the Iberdrola SQL DDBB
- API-EA04: REST WS to collect historical data from Drive Train Central Enterprise Server

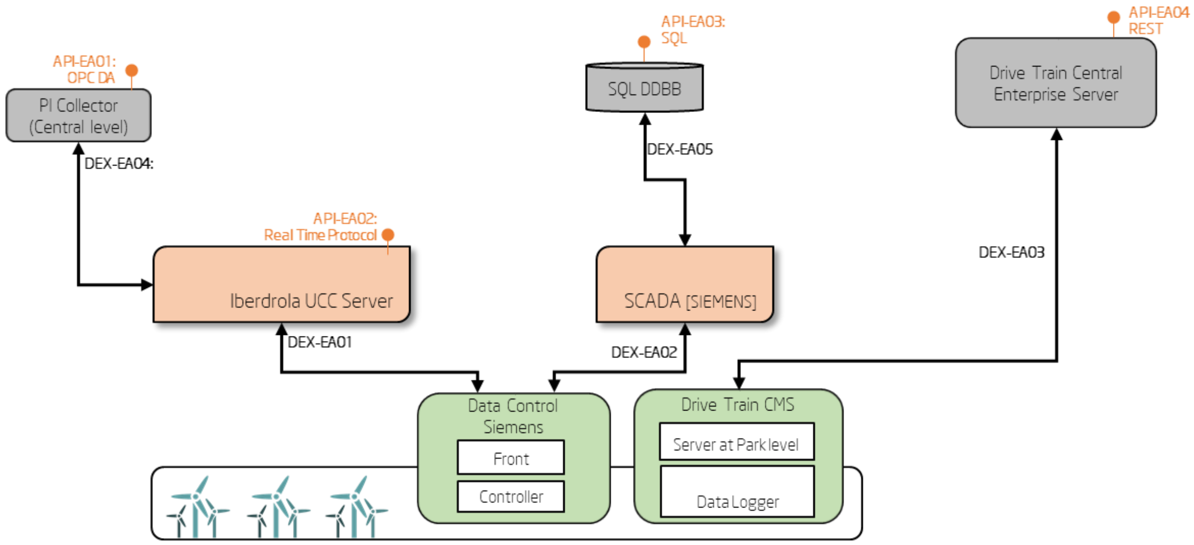


Figure 11: Utility Control and Management layer from East Anglia I

3.2.1.2.1 Iberdrola UCC Server

The information and details of the component are the same as the ones included in Wikinger architecture (see section 3.1.1.2.2). The table below includes the **dependencies** with other ROMEO components specific for East Anglia I demonstrator.

DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	East Anglia I: Data Control – Siemens.
Dependencies with other components (Outputs)	East Anglia I: PI Collector (Central level) and Node#1.
Other Comments	It can also acts as a supervisory unit for monitoring and controlling the connected devices (SCADA).

3.2.1.2.2 PI Collector

The information and details of the component are the same as the ones included in Wikinger architecture (see section 3.1.1.2.3). The table below includes the **dependencies** with other ROMEO components specific for East Anglia I demonstrator.

DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	East Anglia I: Iberdrola UCC Server.
Dependencies with other components (Outputs)	East Anglia I: Babel through OPC-DA.

3.2.1.2.3 SCADA (Siemens)

COMPONENT INFORMATION	
Title	SIEMENS SCADA
Use Case	Supervisory Control and Data Acquisition.
Partner	IBERDROLA
COMPONENT DETAILS	
Description	The Siemens Gamesa Renewable Energy (SGRE) SCADA system consists of the following main sub-systems which will all be located in the OFSS; • Wind Park Supervisor (WPS) – SCADA used for supervision, data acquisition, control and reporting of wind farm performance. Historical data is stored in a SQL database which can be used for generating various reports. • High Performance Park Pilot (HPPP) – Used for enhanced control of the wind farm and regulatory functions (grid code compliance etc.). • Wind Turbine Controllers (WTCs). • Grid Measuring Station (GMS) – Works together with the HPPP and interfaces to the WPS and WTG's. Provides the SCADA system with accurate information about the current grid connection status.
Programing Language	Does not apply
Inputs	The main inputs to the SGRE WPS will be; • HPPP's (8 in total, 4 active & 4 backup) • GMS (8 in total, 4 active & 4 backup) • WTC's (102 in total) • CORE SCADA • Substation Control System SCS
Outputs	The main outputs of the SGRE WPS will be; • Historical data to an SQL database • HPPP's (8 in total, 4 active & 4 backup) • GMS (8 in total, 4 active & 4 backup) • WTC's (102 in total) • CORE SCADA
Integration Mechanisms/Connectors	• Internal (SGRE): HPPP's, GMS and WTC's. • External: CORE SCADA (Communication: Control arbitration between two systems and signal exchange of key digitals, analogues and calculated signals)
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Data Control - Siemens
Dependencies with other components (Outputs)	SQL DB Connection.

3.2.1.2.4 Drive Train Central Enterprise Server

COMPONENT INFORMATION	
Title	Drive Train Central Enterprise Server
Use Case	Condition Monitoring Data Collector for WTGs Drive Train
Partner	IBERDROLA

COMPONENT DETAILS	
Description	This server is placed in Iberdrola´s network and retrieves data from its mirror server at wind farm level through a SQL connection. This second server is part of the Drive Train Condition Monitoring System.
Programing Language	Does not apply.
Inputs	Drive train vibration data.
Outputs	All information is available using Web Script.
Integration Mechanisms/Connectors	
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Drive Train CMS.
Dependencies with other components (Outputs)	REST Web Service connection through a Secure Gateway Endpoint.

3.2.1.3 Real Time Acquisition and Integration

The Real Time Acquisition, Integration and Processing layer bridges the gap between the real-time APIs available at the Utility Control and Management layer and the IBM Cloud. It provides the means for real-time connectivity through different components depending on the characteristics of the data sources. On the one hand it is able to handle the heterogeneity of real deployments where different technologies will have to coexist and interoperate in a transparent way, assuring the interoperability of existing control and sensing systems with additional sensing networks and the deployment of new WT monitoring and control systems. On the other hand, it provides the assembly of a real time data acquisition and processing platform capable of responding to the stringent needs of the WT subdomain, based on the novel edge computing paradigm. In contrast to existing traditional vertical applications, this solution is architected under the Edge computing paradigm and rely on smart distributed nodes that can process large amounts of data in real time and take actions intelligently in an inherently secured, distributed and loosely-coupled way at the local acquisition nodes, without the need of sending all the information back and forth through communication networks with limited bandwidths. Therefore, this solution promotes the distribution of intelligence among nodes located at different levels through an inherently loosely coupled infrastructure. Massive amounts of information will be collected from devices and sensors and they will be partially managed through distributed and unattended Complex Event Processors (CEP engines).

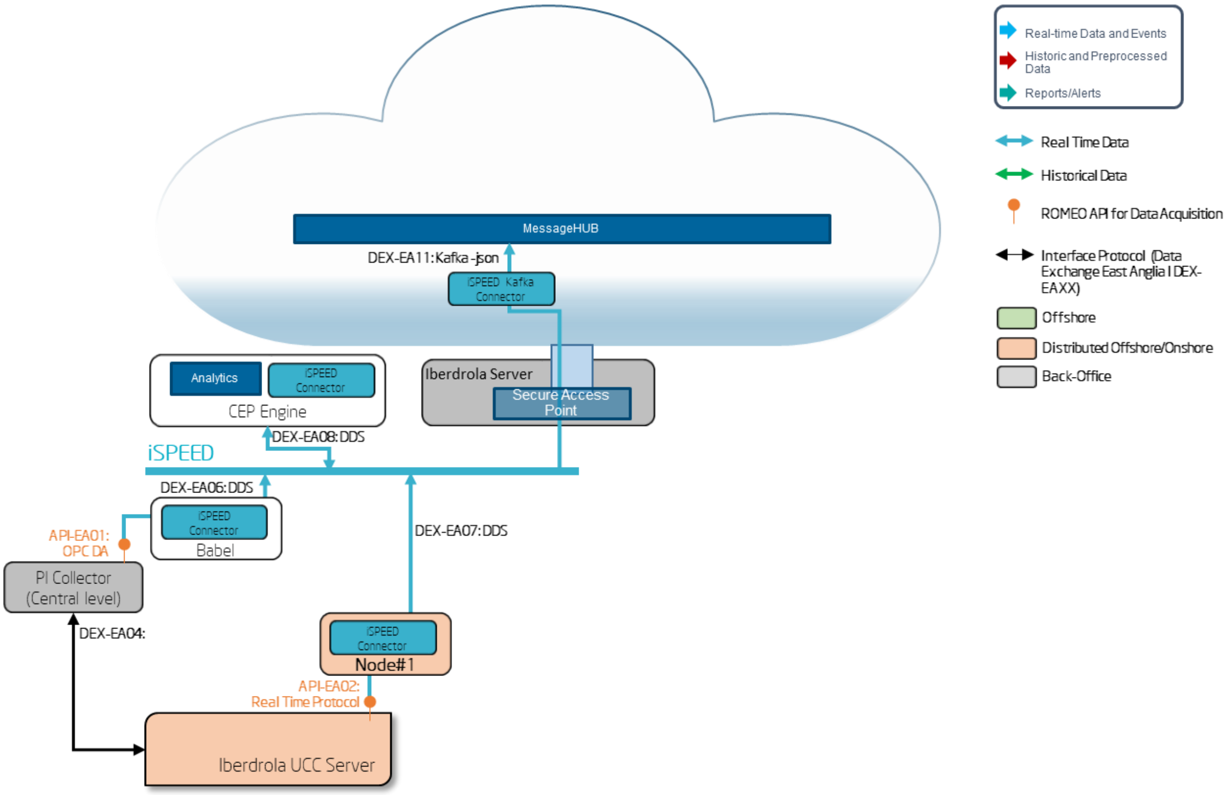


Figure 12: Real time acquisition, integration and processing

This Real Time Acquisition, Integration and Processing layer is formed of four components provided by Indra:

- **Node#1** is a smart gateway prepared to apply on-site analysis techniques, capable of simultaneously processing multiple complex events, guaranteeing bidirectional communication with multiple devices and sensors. It acts as a data concentrator and/or multimedia hub with its computing and data hosting capabilities.
- **Babel** is a real time communications manager that allows communication with different devices through different communication protocols, using only one interface. It has different features and components: high availability, remote access, administration console, tolerance against network drops, extension and communications gateway.
- **iSPEED** is a high performance distributed platform for data exchange based on DDS publish-subscribe mechanisms. It has the ability to handle large volumes of data from the network nodes in a secured, distributed and loosely-coupled way. iSPEED is capable of exchanging data low level latencies and integrating all kind of devices and systems in a common infrastructure.
- **CEP Engine:** the Complex Event Processing engines are able to reside on the edge (Node#1) or on top of the iSPEED real time platform. Rules will be developed inside the CEP engines in order to detect anomalies in the real-time data available in the iSPEED platform. These services are intended to be running on a permanent basis in order to get the highest amount of information out of the real time data generated by the WT. This way, relevant turbine events and patterns can be automatically identified in the huge flow of data that characterizes the wind turbine domain.

In East Anglia I demonstrator Node#1 smart gateway will be deployed in the field to collect the Iberdrola UCC server real-time data and perform on-site analytics in the edge before uploading the information in the IBM Cloud through the iSPEED Real Time platform. Babel will be in charge of collecting real-time data from the OPC DA API of the PI Collector and publish it into iSPEED. On top of this platform there will be CEP engines with loaded patterns analysing the real-time data available in iSPEED. The results will be published in iSPEED and will be transferred to IBM Cloud by a Kafka connector.

All DDS data published into iSPEED by Node#1, Babel and the CEP engines will be received by the iSPEED node that will be hosted in the IBM Cloud. This node will be responsible for transforming the DDS data from iSPEED into json and leave it in the Kafka Message Hub of the IBM Cloud for further processing on the IoT platform.

There will be several **Secure Access Points** to create a secure, persistent connection between the Iberdrola infrastructure and the IBM Cloud. In this case, all data published into iSPEED by Babel, Node#1 and CEP engines will pass through the secure access point hosted at a Iberdrola server and also through the IBM Cloud security access point before reaching the iSPEED node located in this IoT platform.

3.2.1.3.1 Node#1

COMPONENT INFORMATION	
Title	Node#1
Use Case	Node#1 is a Gateway prepared to apply on-site analysis techniques. Gateway capable of simultaneously processing multiple complex events, guaranteeing bidirectional communication with multiple devices and sensors. With its computing and data hosting capabilities, it acts as a data concentrator and/or multimedia hub.
Partner	INDRA
COMPONENT DETAILS	

Description	The main features of Node#1 are: • **Aggregation engine**: Manages the information sent and received between multiple devices, including among others intruder sensors, IP cameras with day/night vision for recording and triggering alerts in the event of movements or sound, and flood alarm systems. Capacity to generate statistical calculations, flows and rules coordinated with IoT services in the cloud. • **Controller and actuator**: Node1 has the ability to execute rules and actions in machines in complex situations, monitor and control the temperature and power consumption, as well as controlling consumption peaks and automatic shutdown of loads. It can also perform detection of anomalous patterns and failure in predictive maintenance. • **Powerful processor**: Nodes with the ability to dynamically deploy applications, increasing real-time processing capacity and maximum response flexibility. • **Intelligent**: Deployment and execution of Artificial Intelligence models in Nodes for the prognosis of machine operation parameters, the detection of anomalous patterns and failure in predictive maintenance and the execution of rules and actions in machines in complex situations
Programing Language	Node#1 has been developed in java, javascript and angular
Inputs	Real Time data from the Iberdrola UCC Server such as
Outputs	Real Time data from the Iberdrola UCC Server and results of the on-site analytics
Integration Mechanisms/Connectors	Node#1 offers several communication interfaces with other devices through the following protocols: • TCP-modbus • RTU-modbus • OPC-UA • Zigbee Home Automation • Zigbee light link In addition it has also several ways to provide the data to external systems such us: • MQTT • DDS through iSPEED • Web Service
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Iberdrola UCC Server through OPC DA
Dependencies with other components (Outputs)	iSPEED (Field Message Bus) via MQTT

3.2.1.3.2 Babel

The information and details of the component are the same as the ones included in Wikinger architecture (see section 3.1.1.3.1). The table below includes the **dependencies** with other ROMEO components specific for East Anglia I demonstrator.

DEPENDENCIES WITH OTHER ROMEO COMPONENTS

Dependencies with other components (Inputs)	PI Collector (Central Server) [OPC DA] Iberdrola SCADA Server [OPC UA] SCADA Front End [SIEMENS] [OPC UA] Drive Train Central Enterprise Server[OPC UA]
Dependencies with other components (Outputs)	iSPEED (Field Message Bus)

3.2.1.3.3 iSPEED

The information and details of the component are the same as the ones included in Wikinger architecture (see section **¡Error! No se encuentra el origen de la referencia.**). The table below includes the **dependencies** with other ROMEO components specific for East Anglia I demonstrator.

DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Edge Intelligent Node (Node#1) via MQTT Babel via iSPEED API
Dependencies with other components (Outputs)	IOT Cloud via Kafka connector

3.2.1.3.4 CEP Engine

COMPONENT INFORMATION	
Title	CEP Engine
Use Case	• Massive amounts of information will be collected from devices and sensors and they will be partially managed through distributed and unattended Complex Event Processors (CEP engines) residing in on the edge (Node#1) or on top of the iSPEED Field Message Bus. • These services are intended to be running on a permanent basis in order to get the highest amount of information out of the real time data generated by the WT. This way, relevant turbine events and patterns can be automatically identified in the huge flow of data that characterizes the wind turbine domain. • These real time services will focus mainly on the validation of the consistency of the WT status, continuously checking that all the turbine measurements are consistent and allowing a more reliable operation.
Partner	INDRA
COMPONENT DETAILS	
Description	• The technology of the CEP Engines that will be implemented in ROMEO will be either Siddhi or InfluxDB. • Siddhi is the Complex Event Processing engine that will be used in ROMEO. Rules defined by Siemenes in D1.4 will be developed inside

	the CEP engine in order to detect anomalies in the real-time data available in the iSPEED Field Message Bus. Complex Event Processing (CEP) engines provide a declarative way of transforming a set of input streams into one or more output streams. The declarative language provided by CEP is usually based on some extension of SQL to include time, pattern matching and causality. CEP engine will be integrated with the iSPEED Field Message Bus which is developed under the DDS standard. The combination of DDS + CEP is not only natural but a perfect fit for building high performance, highly available Stream Processing Systems. Siddhi CEP is a lightweight, easy-to-use Open Source Complex Event Processing Engine (CEP) under Apache Software License v2.0. Siddhi represents events (tuple) using a tuple data structure, which resembles a row in relational database tables. It enables the use of SQL like queries and also incorporates relational database optimization techniques to the system. Onesait Edge Node#1 uses a set of base containers provided by influxdata as part of open source for use at a local node. Database is a high-performance data store written specifically for time series data. It allows for high throughput ingest, compression and real-time querying of that same data. It can handle millions of data points per second. Working with that much data over a long period of time can create storage concerns. It will automatically compact the data to minimize your storage space. Following functionalities are available: • Agent for collecting and reporting metrics • Time Series Database built from the ground up to handle high write & query loads • Administrative user interface and visualization engine • Native data processing engine. It can process both stream and batch data from InfluxDB
Programing Language	The rules inside the CEP engine will be developed in Java.
Inputs	Real-Time data coming from the Iberdrola UCC Server (Generator speed, wind direction, …)
Outputs	Events/Warnings when anomalies are detected in the real-time data
Integration Mechanisms/Connectors	• iSPEED Java API
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Node#1 and iSPEED Field Message Bus (Data coming from Iberdrola UCC Server)
Dependencies with other components (Outputs)	iSPEED Field Message Bus

3.2.1.4 IoT Platform

The IoT Platform is called “IBM Cloud”. It is responsible for centrally managing all the interfaces and components within the ROMEO ecosystem. This platform will store the real-time data provided by iSPEED as well as the historical data directly imported from the Scada SQL DDBB and the Drive

Train Central Server. IBM Cloud will host the WP3 statistical models for failure predictions developed in WP3 as well as the rules libraries defined by Siemens.
The results of the models will be provided to the O&M platform from Uptime as well as to Domina G and SAP tools from Iberdrola, which are systems located outside the IBM platform; in a similar way as in the Wikinger architecture.

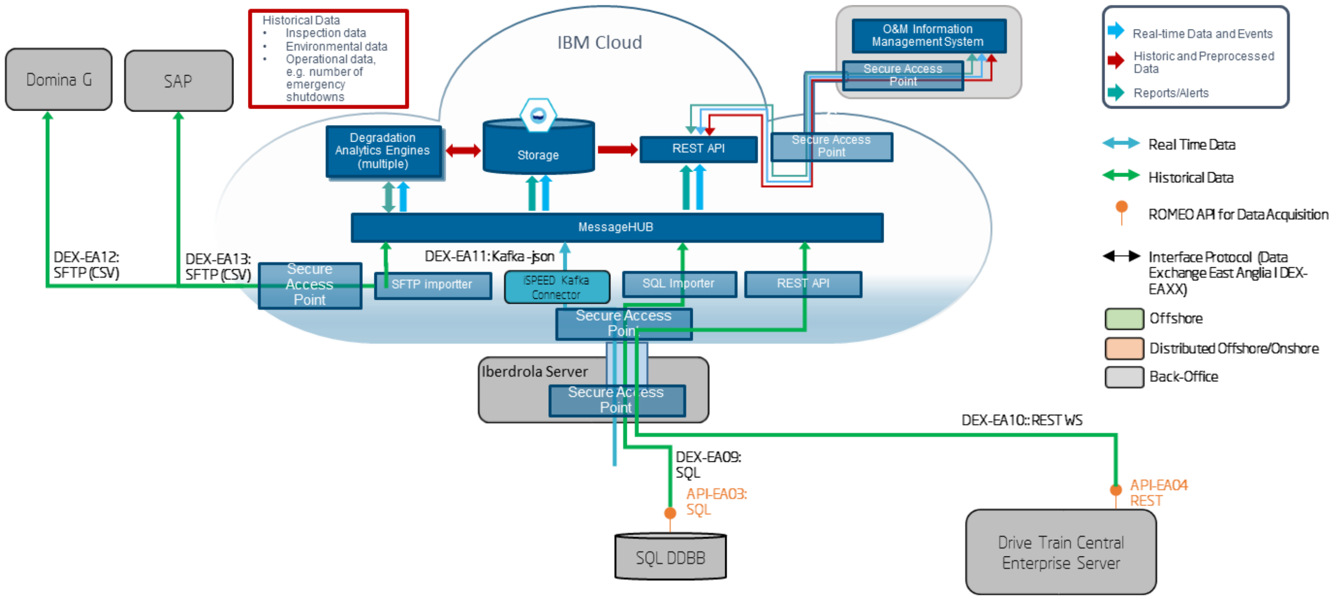


Figure 13: IBM Cloud

All data entering into the IBM Cloud will pass through a Secure Access Point in order to create a secure and persistent connection between the Iberdrola servers where the data is collected and the IBM Cloud. In addition, the end-user tools O&M, SAP and Domina G will also be connected to a secure access point to guarantee the data privacy.

The **IoT Platform** has three internal components:

- **Cloud Message HUB** that connects the internal and external sources to the rest ofIBM Cloud components.
- **Cloud Data Store** to store raw, aggregated, context and analysis results data in a relational and no-SQL databases.
- **Cloud Analytics Engine**, which provides a platform for data processing and machine learning to host the WP3 statistical models of East Anglia I.

3.2.1.4.1 Cloud Message HUB

The information, details and dependencies of the component are the same as the ones included in Wikinger architecture (See section 3.1.1.4.1).

3.2.1.4.2 Cloud Data Store

The information, details and dependencies of the component are the same as the ones included in Wikinger architecture (See section 3.1.1.4.2).

3.2.1.4.3 Cloud Analytics Engine

The information, details and dependencies of the component are the same as the ones included in Wikinger architecture (See section 3.1.1.4.3).

3.2.1.5 Analytics

In East Anglia I demonstrator there are 2 types of analytics hosted in different locations in the architecture: edge computing allocated in Node#1 smart gateway or in CEP engines connected on top of the iSPEED real-time platform and cloud analytics allocated in the analytics engines available at the IBM Cloud.

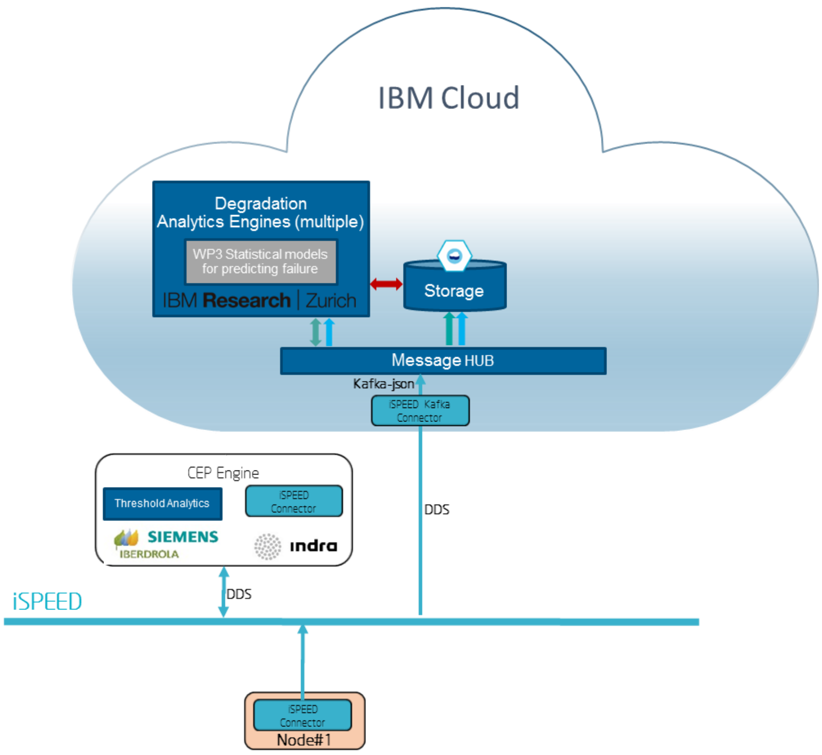


Figure 14: East Anglia I Analytics Infrastructure

<u>Edge Computing Analytics</u>

The real time acquisition and processing platform previously presented in section 3.2.1.3 is capable of responding to the stringent needs of the WT subdomain, based on the novel edge computing paradigm. In contrast to existing traditional vertical applications, this solution is architected under the Edge computing paradigm and rely on smart distributed nodes that can process large amounts of data in real time and take actions intelligently in an inherently secured, distributed and loosely-coupled way at the local acquisition nodes, without the need of sending all the information back and forth through communication networks with limited bandwidths. Therefore, this solution promotes the distribution of intelligence among nodes located at different levels through an inherently loosely coupled infrastructure.

Massive amounts of information will be collected from the Iberdrola UCC server and they will be partially managed through distributed and unattended Complex Event Processors (CEP engines) allocated in the Node#1 smart gateway and in distributed nodes on top of the iSPEED platform.

The threshold rules defined by Siemens in D1.4 [2] will be developed by Indra and hosted in the Node#1 and CEP engines. These engines will be subscribed to the data collected from the Iberdrola UCC server and will detect the cases when the defined patterns are satisfied. When this happens, the results of the threshold rules will be again published into iSPEED and will directly reach the iSPEED Kafka connector allocated in the IBM Cloud for the later processing in the IoT platform.

Rules libraries

From the the outputs of D1.4, rules libreries, assessed and proposed by SiemensGamesa, will be built to hold in threshold values of WT wind turbine monitoring purposes deliverable.
These rules will be built by data collected from different systems:

- BFA13 Low distribution system, that has a condition monitoring of battery capacity
- MDL10 Yaw system that has a measurement to detect failure modes on motor Gera
- MUR20 Common cooling system, for detection of gas leakage in expansion tank
- XSD20 obstacle warning system, UPS battery self-test
- MSE40 Converter cooling system, that detects temperature sensors in converter cabinet

Related to the rules library, due to the small number proposed rules from D1.4, Iberdrola is proposing developing further methodologies to monitor key minor componets/systems section 3.2.1.5.1.

<u>Cloud Analytics</u>

The **IBM Predictive Model Suite**, previously presented in the Wikinger architecture will be also running for the East Anglia I demonstrator. In this case, these software models are allocated in the degradation analytics engines module of the IBM Cloud platform. This software provides predictions of incoming failures for the main components in a wind turbine, such as blade bearing, electrical drive train and mechanical drive train.

3.2.1.5.1 WP3-IBM-Predictive Model Suite

The information, details and dependencies of the component are the same as the ones included in Wikinger architecture. (See section 3.1.1.5.4).

3.2.1.6 Utility Application Layer - End User

The Utility Application Layer-End User of East Anglia I will be very similar to the one from Wikinger demonstrator. There will be three components in this layer to collect the results of the analytics models available at the IBM Cloud:

- One component from Uptime: **O&M Information Management Platform**
- Two components from Iberdrola: **Domina G** and **SAP**

The **O&M platform** will be connected to the IoT platform through a REST API using also a Secure Access Point to guarantee the security of the information exchanged. This Information Management System is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. This platform will be fed with the results of the analytics embedded in the IoT platform and will provide KPI's and metrics to WP8 for assessing and quantifying the impact of the overall system on cost reduction potential.

Iberdrola's **Domina G** tool connects operational data from different resources: SCADA's, PI's, CORE, MeteoFlow, Meters, manual data entered by O&M staff and provides raw data and calculated information such as global indicator. It will be connected to the IoT platform through SFTPprotocol.

SAP is an enterprise resource planning software used to cover all day-to-day processes of Iberdrola. This database server system stores and retrieves data from the separate modules to provide real-time information. SAP will use the SFTP protocol to connect to the IBM Cloud.

In addition, SAP and DOMINA G will send information to the IBM cloud such as work orders, alerts and other relevant information about the status of the wind turbines that would make easy to understand their behavior.

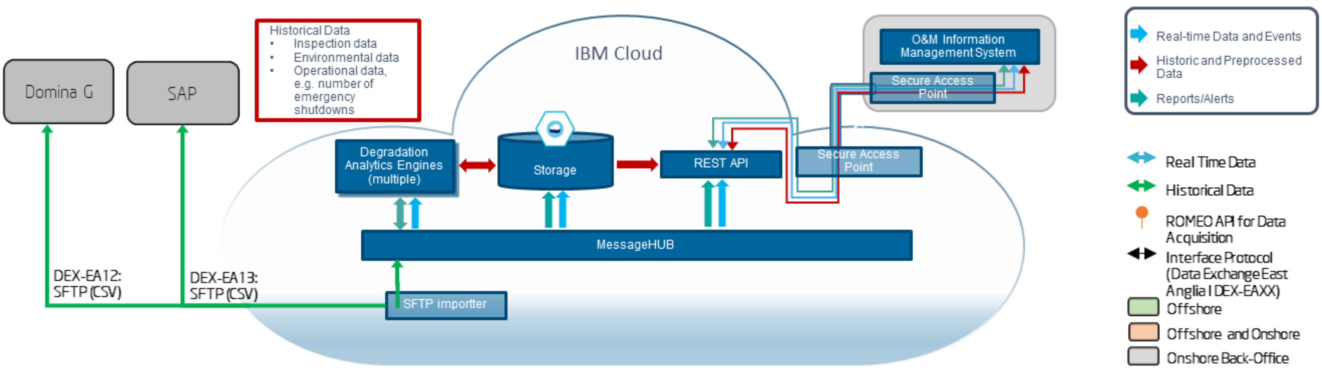


Figure 15: Utility Application Layer - End User from East Anglia I

3.2.1.6.1 O&M Information Management Platform

The information, details and dependencies of the component are the same as the ones included in Wikinger architecture. (See section 3.1.1.6.1).

3.2.1.6.2 Domina G

The information, details and dependencies of the component are the same as the ones included in Wikinger architecture. (See section 3.1.1.6.2).

3.2.1.6.3 SAP

The information, details and dependencies of the component are the same as the ones included in Wikinger architecture. (See section 3.1.1.6.3)

3.2.2 Physical Architecture Overview

The tentative IT infrastructure that is needed in East Anglia I to host the different ROMEO components which take part in this demonstrator is depicted in the figure below.

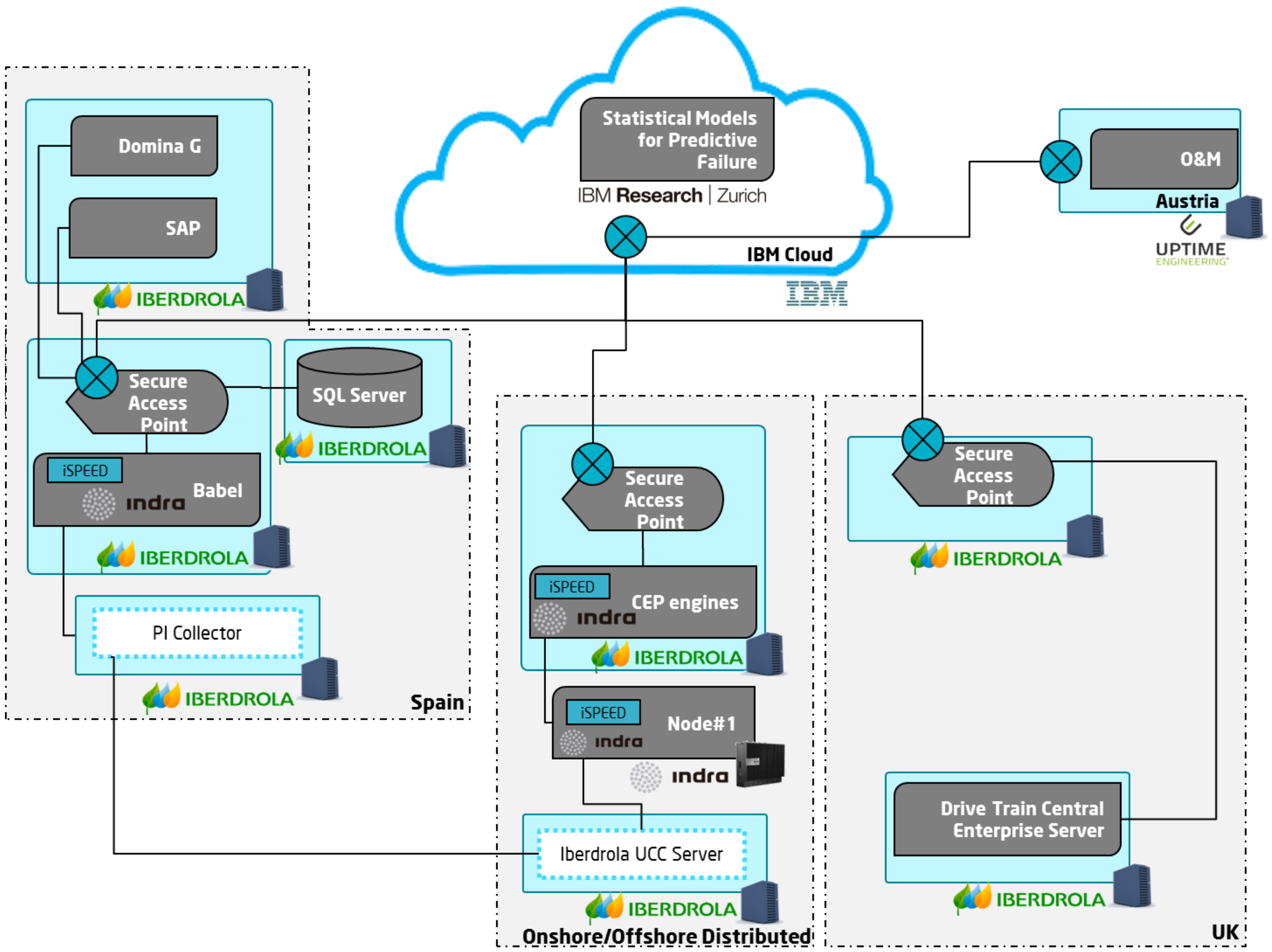


Figure 16: Physical East Anglia I Architecture overview

First, there is a **Secure Access Point** to create a secure and persistent connection between all the servers that will be connected to the IBM Cloud. In this case, the data will be exchanged to the Cloud platform by servers located in different places: onshore/offshore, Spain, UK and Austria.

In each location there will be one secure access point that will be connected to all the servers hosting ROMEO systems in that location. These servers in different countries are from different owners:

- Distributed between offshore and onshore the following systems will be located:
  - Secure Access Point
  - Iberdrola UCC Server
  - Indra's Node#1 smart gateway connected to the Iberdrola UCC Server
  - CEP Engines
- In Spain there will be four Iberdrola Servers hosting the following systems:
  - PI Collector system
  - Secure Access Point and Babel system connected to the PI Collector
  - SQL Server with Scada data
  - SAP and Domina G systems

- In UK there will be two Iberdrola Servers to allocate:
  - Drive Train Center Enterprise Server
  - Secure access point
- In Austria there will be one Uptime server hosting:
  - O&M Information Management System

During the implementation and deployment of the whole infrastructure (WP7), some changes may be needed in the hosting of the different tools depending on the performance of the whole ecosystem.

3.3. Teesside architecture

The third site that has been selected to prove the methodologies of the project is Teesside Offshore wind farm that has been operated by EDF Energy Renewables since 2013.

The figure below shows Teesside architecture which contains the different systems that take part in this demonstrator from the data acquisition from different sources, upload of the information to the back-office layer, the IoT analytics infrastructure that allows the execution of the diagnosis and prognosis models up to the end-user O&M application to show the results.

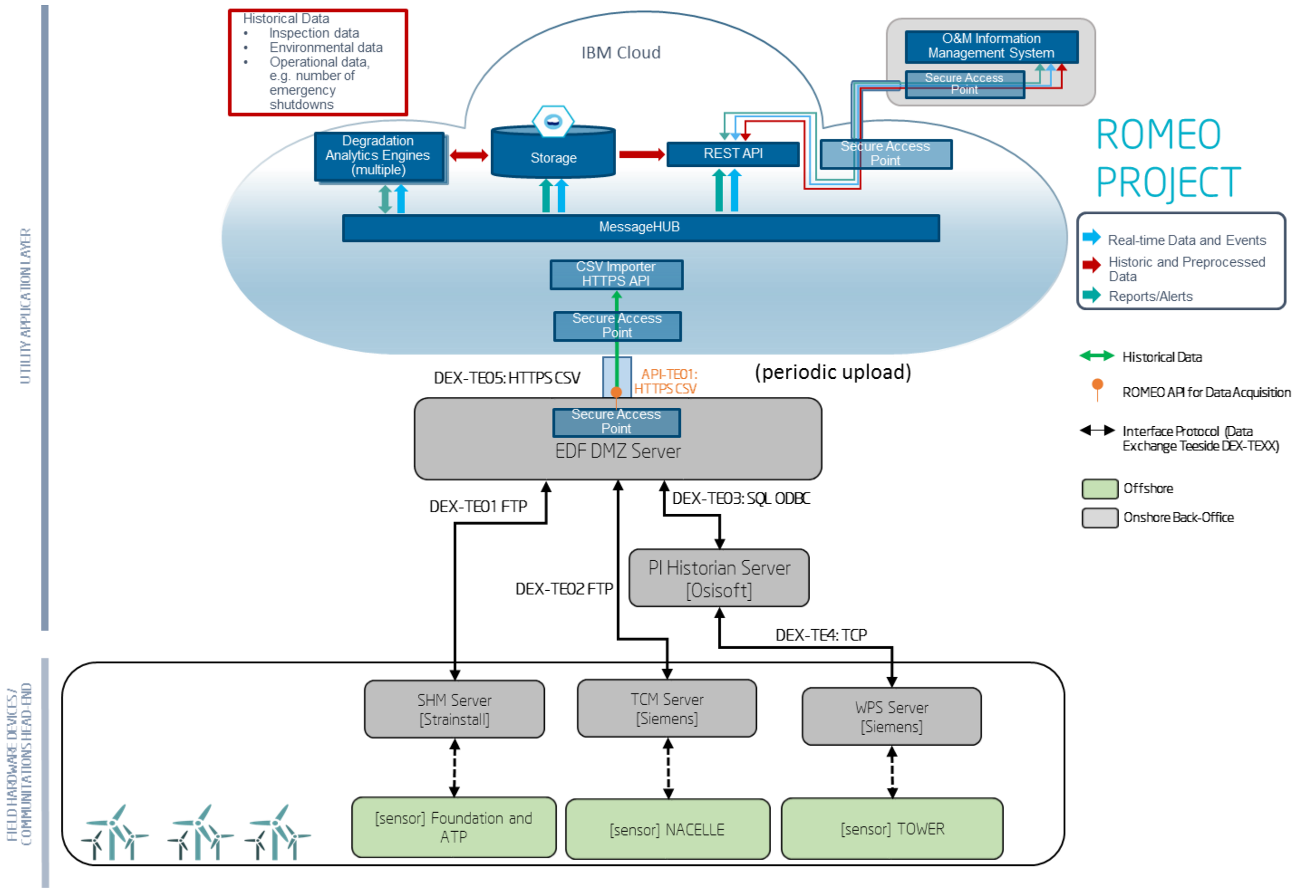


Figure 17:Teesside architecture

The first layer we find in the Teesside Architecture is the **Field Hardware Devices/Communication Head End layer** which contains 3 head-end systems located onshore from Strainstall and Siemens which are in charge of collecting the data from the Foundation and TP, Nacelle and Tower sensors of the windfarm.

At the **Utility Control and Management** layer there is a **demilitarized zone** where we can find the **EDF DMZ Server** where all the data collected from the field needed in this demonstrator will be uploaded. This server is used to connect securely to different servers in the Teesside network and can be used to query, retrieve, process and store data from different Teesside databases. In addition, this server will be used to send data to the IoT platform outside of Teesside network. Moreover, in this layer we have the **PI Historian Server**, which is a time series database that provides data storage with compression algorithm and a data query interface.

On the one hand, the data from the Foundation and TP and Nacelle, collected by the TCM and SHM servers respectively, is exchanged through FTP protocol to the EDF **DMZ Server**. On the other hand, the data from the Tower, which is acquired by the WPS Server, is sent through TCP protocol to the **PI Historian Server** and from this one through SQL ODBC to the **EDF DMZ Server.**

The components in both Field Hardware Devices/Head End and Utility Control and Management layers are existing systems for the windfarm operation in the AS-IS architecture of Teesside.

Unlike Wikinger and East Anglia I, in the Teesside architecture there is not a real-time integration to connect to the next ROMEO layer, the **IoT Cloud platform**, as only historical data will be sent periodically to the IBM Cloud platform through a HTTPS CSV API. Thus, in this demonstrator there will be just one data acquisition point API from EDF to collect data for ROMEO. This connection will be conducted through a **Security Access Endpoint** in order to ensure the data privacy when the data is entering to the platform through the **CSV importer**.

The WP5 IBM Cloud platform in the Teesside demonstrator is aimed to store the data coming from the field, host the WP3 statistical and drive train models and provide the results to the O&M platform. The **IoT Platform** has three internal components:

- **IBM Cloud Message HUB** that connects the internal and external sources to the IBM Cloud components.
- **IBM Cloud Data Store** to store raw, aggregated, context and analysis results data in a relational and no-SQL databases.
- **IBM Cloud Analytics Engine**, which provides a platform for data processing and machine learning to host WP3 algorithms.

The last layer of the architecture is the **Utility Application Layer- End User**, which is connected to the IoT platform through a REST API using also a Secure Access Point to guarantee the privacy of the information exchanged. This layer contains the WP6 Uptime O&M Information Management System, which is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. This platform will be fed with the results of the analytics embedded in the IoT platform and will provide KPI’s and metrics to WP8 for assessing and quantifying the impact of the overall system on cost reduction potential.

3.3.1 Component Description

In the following sections, the descriptions of the Teesside architecture components are explained including the inputs and outputs of each component as well as the dependencies with the rest of the components in the architecture.

3.3.1.1 Field Hardware Devices/Communications Head-End

In Teesside architecture, there are three communication head-end systems from Strainstall and Siemens, which are in charge of collecting the data from the Foundation and TP, Nacelle and Tower:

- The **Foundation and TP** sensors are connected to the **SHM Server** [Strainstall] that provides a set of monitoring data to analyse different measures as vibrations, temperature, wind and structural health.
- The **NACELLE** sensors are connected to the **TCM Server** [Siemens], which is a Turbine Condition Monitoring (TCM) system that accesses to the vibration sensors and provides turbine condition monitoring.
- The **TOWER** sensors are connected to the **WPS Server** [Siemens], which is a control solution that provides access to PLS inputs and outputs for accurate regulation of wind turbine operation.

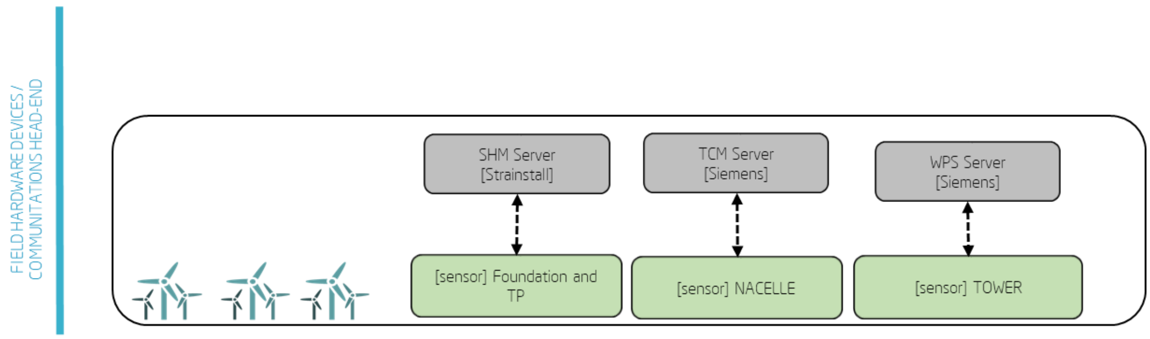


Figure 18: Field Hardware Devices / Communications Head-End. SHM, TCM WPS onshore. Foundation and TP, NACELLE ans Tower offshore

The data from these three systems is sent to the EDF **DMZ Server** component, included in the Utility Control and Management layer, through different protocols:

- SHM Server [Strainstall] and TCM Server [Siemens] exchange data through SFTP protocol with the EDF DMZ Server.
- WPS Server exchange a data through TCP protocol with the PI Historian Server [Osisoft] that connects afterwards through SQL ODBC protocol with the EDF DMZ Server.

3.3.1.1.1 Structural Health Monitoring/ (Foundation Monitoring System)

COMPONENT INFORMATION	
Title	Structural Health Monitoring (SHM) / (Foundation Monitoring System)
Use Case	SHM system provides a set of monitoring solutions to analyze the foundation vibrations, temperature, ph and hydrogen levels, as well as water pressure. It allows a detailed analysis of its structural health.
Partner	EDF
COMPONENT DETAILS	

Description	This is a structural health monitoring system developed by Strainstall. It is composed of 47 sensors on WT03 and WT13. It also provides access to the data through an HMI.
Programing Language	Proprietary
Inputs	Sensor data
Outputs	CSV files to EDF DMZ Server.
Integration Mechanisms/Connectors	SFTP
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	NA
Dependencies with other components (Outputs)	EDF DMZ Server

3.3.1.1.2 Turbine Condition Monitoring (TCM) system

COMPONENT INFORMATION	
Title	Turbine Condition Monitoring (TCM) system
Use Case	The TCM provides turbine condition monitoring through the use of vibration sensors.
Partner	EDF
COMPONENT DETAILS	
Description	The TCM is a monitoring solution that provides access to vibration sensors deployed on critical components of Teesside wind turbines (e.g. Gearbox).
Programing Language	Proprietary.
Inputs	Vibration data from sensors.
Outputs	Fourier spectrum.
Integration Mechanisms/Connectors	SFTP
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	NA
Dependencies with other components (Outputs)	CSV File to EDF DMZ Server

3.3.1.1.3 Wind Power Supervisors (WPS) system

COMPONENT INFORMATION	
Title	Wind Power Supervisors (WPS) system
Use Case	The WPS gathers data from PLC installed by Siemens on their 2.3MW wind turbine.
Partner	EDF
COMPONENT DETAILS	

Description	The WPS is a control solution that provides access to PLS inputs and ouputs for accurate regulation of wind turbine operation.
Programing Language	Proprietary
Inputs	Data from PLCs.
Outputs	Aggregated data (10mn), raw & computed data, error messages.
Integration Mechanisms/Connectors	TCP
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	PLCs installed by Siemens.
Dependencies with other components (Outputs)	TCP connection to Osisoft PI File to EDF DMZ Server.

3.3.1.2 Utility Control and Management

There are two components in the Utility Control and Management layer of Teesside that are the engaged between the Field Hardware Devices/Communications Head-End and the IoT platform. These two components are located at the back-office:

- **PI Historian Server** that uses the interface protocol TCP to connect to the WPS Server [Siemens] and provides data storage with compression algorithm and a data query interface.
- **EDF DMZ Server**, which is a server that will collect the data from the SHM and TCM servers through SFTP as well as from the PI Historian Server through SQL ODBC protocol. It stores data as csv files on its local storage and periodically will send this data to the IBM cloud through HTTPS CSV giving an abstraction of the Teesside architecture to IBM and partners.

There will be a **Security Access Endpoint** embedded in the EDF DMZ Server to connect on a secure way to the IoT Platform.

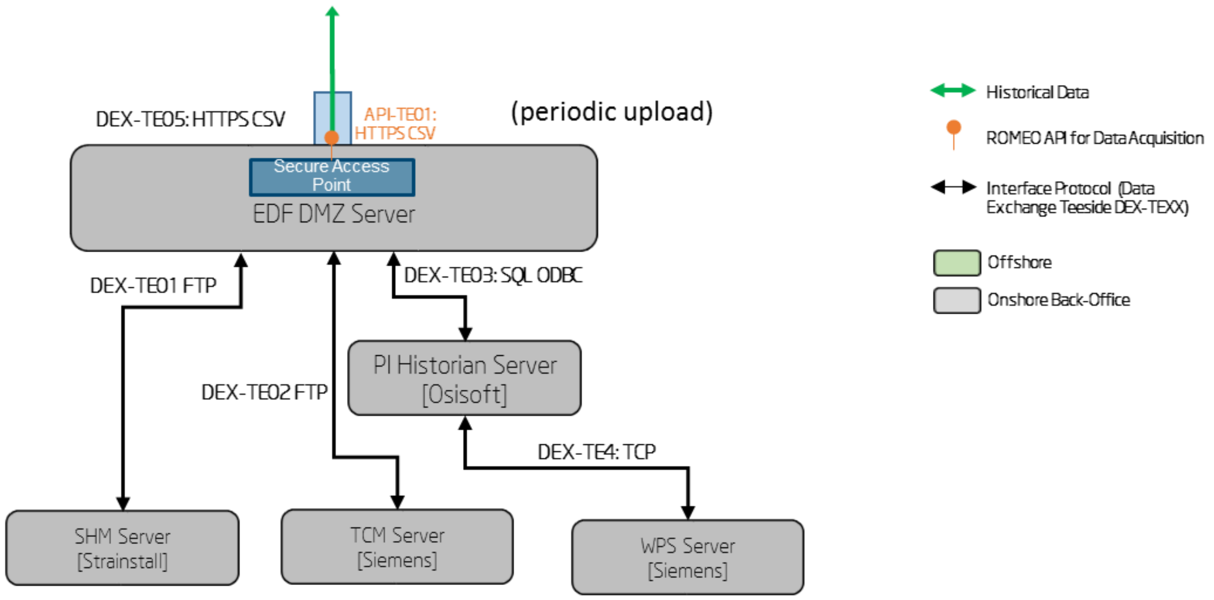


Figure 19: Utility Control and Management layer

3.3.1.2.1 EDF PI Server

COMPONENT INFORMATION	
Title	EDF PI server
Use Case	A data historian that stores times series.
Partner	EDF R&D (UK)
COMPONENT DETAILS	
Description	Osisoft PI is a time series database. It provides: • Data storage with compression algorithm • Data query interface
Programing Language	Unknown (Proprietary software).
Inputs	Wind farm SCADA data through TCP.
Outputs	Compressed and/or preprocessed SCADA data.
Integration Mechanisms/Connectors	TCP & OLEDB SQL query API.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	EDF WPS Server
Dependencies with other components (Outputs)	EDF DMZ server

3.3.1.2.2 EDF DMZ Server

COMPONENT INFORMATION	
Title	EDF DMZ server
Use Case	EDF DMZ server is a server localized in a demilitarized zone. So this server can be used to connect securely to different servers in Teesside network and it can be used to query, retrieve, process and store data from different Teesside databases. It can also be used to send data to a cloud platform outside Teesside network.
Partner	EDF R&D (FRA
COMPONENT DETAILS	
Description	EDF DMZ server is a UNIX server with several scripts developed in Shell, Python and Java. By using these scripts, it can access and retrieve data from EDF PI server through a SQL ODBC connection and data from EDF TCM & SHM servers through SFTP. Then it stores this data as CSV files on its local storage. Periodically, it send this data to the IBM Cloud cloud through HTTPS REST API (upload period still to be defined). EDF DMZ server is the only component that interacts with IBM Cloud in the Teesside architecture. So it provides an abstraction of the Teesside architecture to IBM and others partners.
Programing Language	Shell, Python and Java scripts.
Inputs	Data provided by PI, TCM and SHM servers: Compressed and/or preprocessed SCADA data from EDF PI Historian server. Fourier spectrum from EDF TCM server. CSV file containing sensor data from EDF SHM server.
Outputs	CSV files containing data from PI, TCM and SHM servers – exact format still under discussion.
Integration Mechanisms/Connectors	SFTP (to retrieve data from SHM and TCM servers). SQL ODBC connection (to retrieve data from PI Historian server). CSV files (storage). HTTPS REST API (to send CSV files to IBM Cloud).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	EDF PI Historian server EDF TCM server EDF SHM server
Dependencies with other components (Outputs)	IBM Cloud (through a CSV importer HTTPS API)

3.3.1.3 IoT Platform

The IoT Platform is called "**IBM Cloud**". It is responsible for centrally managing all the interfaces and components within the ROMEO ecosystem. This platform will store the data provided by the EDF DMZ server, host the WP3 statistical and drive train models and provide the results to the O&M platform.

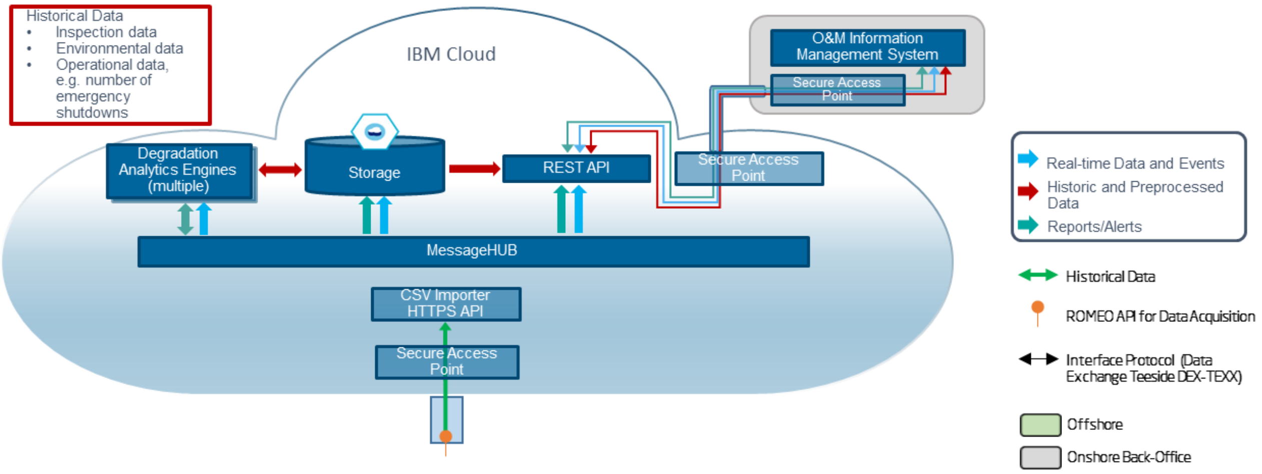


Figure 20: IBM Cloud

First, there is a **Secure Access Point** to create a secure, persistent connection between EDF DMZ server and the cloud. In this case, the data will be entering to the platform only through the **CSV importer** HTTPS API.

The **IoT Platform** has three internal components:

- **Cloud Message HUB** that connects the internal and external sources to the IBM Cloud components.
- **Cloud Data Store** to store raw, aggregated, context and analysis results data in a relational and no-SQL databases.
- **Cloud Analytics Engine** which provides a platform for data processing and machine learning to host the WP3 statistical and drive train models of Teesside.

3.3.1.3.1 Cloud Message HUB

COMPONENT INFORMATION	
Title	Cloud Message Hub
Use Case	Interconnect external data sources (e.g.,from iSPEED, via SFTP or SQL) and internal data sources (e.g., analytics) to Cloud components.
Partner	IBM Research
COMPONENT DETAILS	
Description	The MessageHub is an internal component that connects the external and internal data sources to the Cloud components. Data coming from external sources (e.g., iSPEED, SFTP, SQL) and used the Message Hub to distribute to data. It interconnects with edge components such as the database and the analytics components.
Programing Language	N/A
Inputs	External data coming from wind farms and other sources via iSPEED, SFTP or SQL based sources; result and derived data from analytics.
Outputs	Received data routed to internal components, e.g., the data store and the analytics .

Integration Mechanisms/Connectors	Apache Kafka API
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	iSPEED Kafka Connector and other interfaces (SFTP and SQL access to remote servers, REST interfaces) to pull in wind farm data.
Dependencies with other components (Outputs)	Cloud database instances, analytics components.

3.3.1.3.2 Cloud Data Store

COMPONENT INFORMATION	
Title	Cloud Data Store
Use Case	Store data collected from wind farms, including measurement data (e.g., SCADA, CMS, etc) and event/alert logs, as well as derived data and analytics results from WP3 activities.
Partner	IBM Research
COMPONENT DETAILS	
Description	The data storage component stores raw, aggregated, context, and analysis results data in a relational and no-SQL databases.
Programing Language	N/A
Inputs	Data (raw, derived, aggregated) from the MessageHub (from wind farms) and as well as data from analytics components (results, models).
Outputs	Responses to data queries.
Integration Mechanisms/Connectors	Database query language (SQL, REST)
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud MessageHub
Dependencies with other components (Outputs)	

3.3.1.3.3 Cloud Analytics Engine

COMPONENT INFORMATION	
Title	Cloud Analytics Engine
Use Case	Provide platform for data processing and machine learning to be used by WP3, based on data collected from each of the three wind farms, including and not limited to SCADA, CMS and other sources.
Partner	IBM Research
COMPONENT DETAILS	
Description	Data processing analytics engine providing machine learning libraries and data processing capabilities, e.g. Apache Spark. The exact design and the components are still to be defined and depend on the algorithms to be developed in WP3, as well as further requirements WP3 will impose on WP5.
Programing Language	e.g., Scala, Python, Java

Inputs	Raw or derived data from wind farms, machine learning algorithms. Data to be gathered from the central data store built by WP5.
Outputs	Device/refined data, analytics results.
Integration Mechanisms/Connectors	Integration mechanisms already available for component data exchange.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud data store, Cloud message hub.
Dependencies with other components (Outputs)	Cloud data store, Cloud message hub.

3.3.1.4 Analytics

The analytics involved in the Teesside demonstrator are the following:

- WP3 Mechanical and Electrical Drive Train models that will be developed by EDF.
- WP3 Statistical models for predicting failure that will be developed by IBM.

Both groups of models will run on the Analytics Engine of the IBM Cloud which will be connected to the Data Store and Message Hub in order to retrieve the data needed by the models as well as to provide the algorithms results to be later exchanged to the O&M platform.

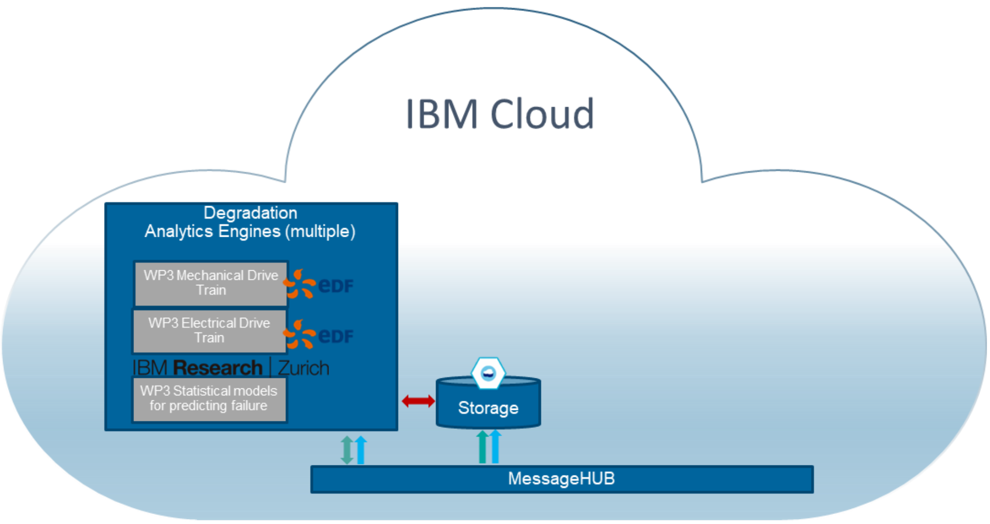


Figure 21: Teesside Analytics

There are ten Algorithms/Models from WP3 that will use Teesside's demonstrator data:

- **IBM Predictive model SUITE** as Statistical models for predicting failure for main components in a wind turbine, such as mechanical drive train, blade bearing and electrical drive train.
- **EDF Models Libraries – WP3 - Mechanical Drive Train – Main Bearing** to take the analogical variables associated to turbine behaviour, for the detection of the failure mode, diagnosis and prognosis.

- **EDF Models Libraries – WP3 – Mechanical Drive Train – Pitch System / blade** bearing to take analogical variables associated to turbine behaviour and pitch system and blade bearing as input data which will be processed to provide an output representative.
- **EDF Models Libraries – WP3 – Mechanical Drive Train – Gearbox** will take the variables associated to turbine behaviour for the gearbox as input data, and after will be processed lo provide input or output representative for the detection of the failure mode.
- **EDF Models Libraries – WP3 – Electrical Drive Train – Transformer Windings** that will take analogical variables associated to turbine behaviour for the transformer windings as input data.
- **EDF Models Libraries – WP3 – Electrical Drive Train – Transformer Cooling System** will take analogical variables associated to turbine behaviour and relevant for the transformer cooling system as input data.
- **EDF Models Libraries – WP3 – Electrical Drive Train – Converter** will take analogical variables associated to turbine behaviour and relevant for the converter system as input data, processed to provide representative detection of the failure mode, for the diagnosis and prognosis.
- **EDF Models Libraries – WP3 – Electrical Drive Train – DC bus – capacity degradation** will take analogical variables associated to turbine behaviour and relevant for the DC bus capacity as input data.
- **EDF Models Libraries – WP3 – Electrical Drive Train – DC bus – open circuit** will take analogical variables associated to turbine behaviour and relevant for the DC bus as input data.
- **EDF Models Libraries – WP3 – Electrical Drive Train – Generator windings** will take analogical variables associated to turbine behaviour and relevant for the Generator Windings as input data.

These models will use the data collected from the field available in the data store of IBM cloud as well as the EDF failure data sets that indicate where the Teesside data is considered as anomalous.

3.3.1.4.1 EDF Failure Data Set

COMPONENT INFORMATION	
Title	EDF failure data set
Use Case	This data set presents data form Teesside. It also provides a signal to define where Teesside data is considered anomalous.
Partner	EDF R&D (FRA)
COMPONENT DETAILS	
Description	This dataset is constituted by one or several csv files. The first column is the timestamp of data measurement. Other columns represents sensors measurements from WPS, TCM and SHM (around 250 per wind turbine). One column (“error”) is the indication that the data is considered as a failure.
Programing Language	Python, csv-utf8
Inputs	loosely based on PI server data, TCM server data, SHM server data.
Outputs	NA
Integration Mechanisms/Connectors	Model will be stored on Cloud either on CSV files or others (Parquet/Orc) depending on technology choice.

DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	NA
Dependencies with other components (Outputs)	Physical libraries / Machine Learning models.

3.3.1.4.2 IBM Predictive model SUITE

COMPONENT INFORMATION	
Title	IBM PREDICTIVE MODEL SUITE
Use Case	Statistical models for failure predictions.
Partner	IBM Research
COMPONENT DETAILS	
Description	It is software functionality for providing predictions of incoming failures for main components in a wind turbine, such as blade bearing, electrical drive train, mechanical drive train, etc. It uses data about various failure modes coupled with the physical models of components and historical sensor measurements to learn and predict what constitutes normal and abnormal behavior. It stores predictions as trigger alerts and can generate time-based reports.
Programing Language	Python
Inputs	Failure data sets, failure modes and physical model APIs provided for several WT components (blade bearing, electrical drive train, mechanical drive train, gearbox) by Adwen, EDF and Bachmann, as well as rules library provided by Siemens.
Outputs	Failure Predictions per WT Component Type: • Binary prediction (0=no failure, 1=failure) per component and time window (e.g., upcoming x days, y weeks). • Failure risk with range [0,100]% per component and time window. • Bucketed failure risk (e.g., levels = {not problematic, moderately problematic, highly problematic, critical) per component and time window. • Remaining useful lifetime (e.g., expressed in days/weeks/months) per component.
Integration Mechanisms/Connectors	Interface with the Secure Access Point / iSPEED Kafka connector (Cloud). Interface with the Storage Component (Cloud).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Data from Adwen, EDF, Bachmann and Siemens. Model result storage DB in IBM Cloud.
Dependencies with other components (Outputs)	-

3.3.1.4.3 EDF Models Libraries – WP3 - Mechanical Drive Train – Main Bearing

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Mechanical Drive Train – Main Bearing

Use Case	Diagnosis & Prognosis based on a physical approach and associated to the Main Bearing.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the main bearing as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	Python – compiled to a library
Inputs	T_MainBearing, T_Nacelle, Rotor RPM, Power Output.
Outputs	Time evolution of error between measured and estimated bearing temperature. The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.4 EDF Models Libraries – WP3 – Mechanical Drive Train – Pitch System / blade bearing

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Mechanical Drive Train – Pitch System / blade bearing
Use Case	Diagnosis & Prognosis based on a physical approach and associated to the Pitch System / Blade Bearing.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the pitch system and blade bearing as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	Python – compiled to a library
Inputs	Power output, Pitch angle, Generator speed, Wind speed (average, max), Rotor RPM, Absolute difference in pitch angle position, pitch motor torque if available.
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).

DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.5 EDF Models Libraries – WP3 – Mechanical Drive Train – Gearbox

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Mechanical Drive Train – Gearbox
Use Case	Diagnosis & Prognosis based on a physical approach and associated to the Gearbox.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the gearbox as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	Python – compiled to a library
Inputs	Gearbox temperature, Vibrations, Oil Particles counting, Wind Speed, Power output, Nacelle temperature, ambient temperature.
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.6 EDF Models Libraries – WP3 – Electrical Drive Train – Transformer Windings

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Electrical Drive Train – Transformer Windings
Use Case	Diagnosis & Prognosis based on a physical approach and associated to the transformer windings.
Partner	EDF R&D (FR)
COMPONENT DETAILS	

Description	The component will take analogical variables associated to turbine behavior and relevant for the transformer windings as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	First version Matlab / then to be defined – compiled to a library.
Inputs	Availability to be confirmed: • Current & Voltage on LV side - rms value. • Current & Voltage on HV side - rms value. • Active & Reactive power. • Current & Voltage on LV side - high sampling. • Current & Voltage on HV side - high sampling. • Oil Temperature, Other temperature probes.
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.7 EDF Models Libraries – WP3 – Electrical Drive Train – Transformer Cooling System

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Electrical Drive Train – Transformer Cooling System
Use Case	Diagnosis & Prognosis based on a physical approach and associated to the transformer cooling system.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the transformer cooling system as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	First version Matlab / then to be defined – compiled to a library
Inputs	Availability to be confirmed: • Current&Voltage on LV side - rms value. • Current & Voltage on HV side - rms value. • Active & Reactive power - rms value. • Oil Temperature, Other temperature probes.
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time.

	A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.8 EDF Models Libraries – WP3 – Electrical Drive Train – Converter

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Electrical Drive Train – Converter
Use Case	Diagnosis & Prognosis based on a physical approach and associated to the converter.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the converter system as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	First version Matlab / then to be defined – compiled to a library.
Inputs	Availability to be confirmed: • Current & Voltage on both sides of the converter - rms value. • Alarms of the converter.
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.9 EDF Models Libraries – WP3 – Electrical Drive Train – DC bus – capacity degradation

COMPONENT INFORMATION

Title	EDF Models Libraries – WP3 – Electrical Drive Train – DC bus – capacity degradation
Use Case	Diagnosis & Prognosis based on a physical approach and associated to the DC bus capacity.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the DC bus capacity as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	First version Matlab / then to be defined – compiled to a library.
Inputs	Availability to be confirmed: • DC bus voltage - high sampling. • Current and voltage - high sampling. • Active power - high sampling.
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.10 EDF Models Libraries – WP3 – Electrical Drive Train – DC bus – open circuit

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Electrical Drive Train – DC bus – open circuit
Use Case	DC Diagnosis & Prognosis based on a physical approach and associated to the bus.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the DC bus as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	First version Matlab / then to be defined – compiled to a library
Inputs	Availability to be confirmed: DC bus voltage - low sampling
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.

Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.4.11 EDF Models Libraries – WP3 – Electrical Drive Train – Generator windings

COMPONENT INFORMATION	
Title	EDF Models Libraries – WP3 – Electrical Drive Train – Generator windings
Use Case	Diagnosis & Prognosis based on a physical approach and associated to the Generator windings.
Partner	EDF R&D (FR)
COMPONENT DETAILS	
Description	The component will take analogical variables associated to turbine behavior and relevant for the Generator Windings as input data. These data will be processed to provide an output representative for the detection of the failure mode, for the diagnosis and the prognosis. Details of the component will be made available through a descriptive document and the use of executable files (library).
Programing Language	First version Matlab / then to be defined – compiled to a library.
Inputs	Availability to be confirmed: • Generator voltage - high sampling. • Generator current - high sampling. • Rotational speed - high sampling.
Outputs	The failure levels (corresponding to those detailed in the D3.1 [3] and describing the failure risk) and the time. A coefficient corresponding to the status of the component and the time.
Integration Mechanisms/Connectors	Model will be run on Cloud with the data uploaded from csv files. Call of a function or executable (Typical OS functions).
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	Cloud components: MQ message bus, Data storage.
Dependencies with other components (Outputs)	O&M management system through Cloud.

3.3.1.5 Utility Application Layer - End User

For the case of the Teesside demonstrator there will be only one component in this layer, which is the WP6 **O&M Information Management Platform** from Uptime.

The O&M tool is connected to the IoT platform through a REST API using also a Secure Access Point to guarantee the security of the information exchanged. The O&M Information Management System is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. This platform will be fed with the results of the analytics embedded in the IoT platform and will provide KPI's and metrics to WP8 for assessing and quantifying the impact of the overall system on cost reduction potential.

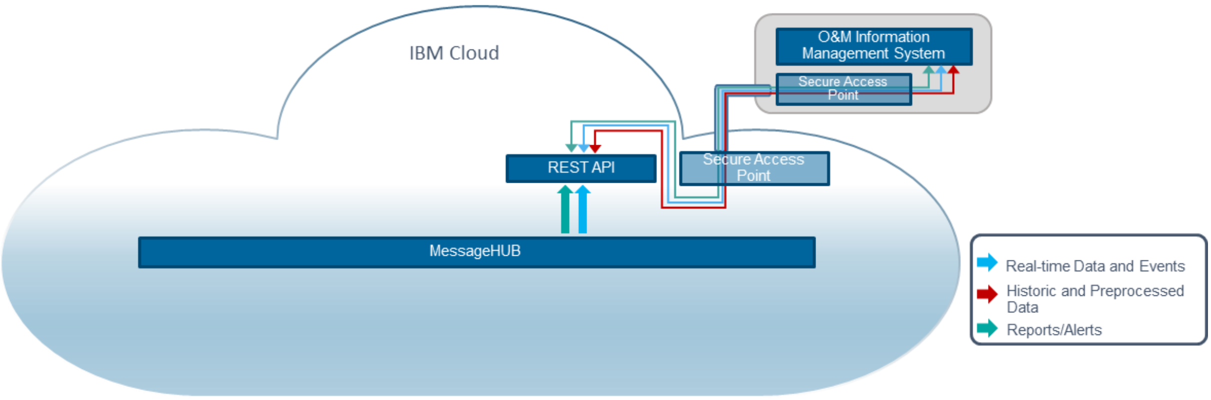


Figure 22: Utility Layer - End User

3.3.1.5.1 O&M Information Management Platform

COMPONENT INFORMATION	
Title	O&M Information Management Platform aka Uptime HARVEST
Use Case	Uptime HARVEST is a holistic, business wide platform for O&M and reliability optimization, combining various inputs in order to support monitoring, inspection, and maintenance of wind farms. As a web application it provides a central point of access to analyze pre-aggregated data and generate specific, actionable information that can be fed into asset management, allowing effective feedback from the field for continuous improvement.
Partner	UPTIME Engineering
COMPONENT DETAILS	
Description	Features of Uptime HARVEST: • **Integration** with IoT platform. • **Effective visualization and reporting** of post processed data. • **Aggregation of data** to ensure information generated by the IoT platform is converted into effective advisory results, based on business-critical processes (defined in WP1) in context of operations and asset management considering needs of pilot tests prepared during WP7. • **Support of maintenance task management.** • **Support of knowledge management.** • **Automation of business processes** to minimise cost. • **KPI's and metrics** to WP8 for assessing and quantifying the impact of the overall system on cost reduction potential.
Programing Language	C#, T-SQL

Inputs	Events generated by IoT Cloud. Pre-aggregated time series information from IoT Cloud.
Outputs	Graphical user interface. No other outputs specified yet.
Integration Mechanisms/Connectors	RabbitMQ already available for integration (message format to be specified). REST API to be developed to integrate with IoT Cloud.
DEPENDENCIES WITH OTHER ROMEO COMPONENTS	
Dependencies with other components (Inputs)	IoT Cloud. NO integration with iSPEED field message bus.
Dependencies with other components (Outputs)	WP7 (Iberdrola, Testing): No APIs to support automatic testing specified yet. WP8 (Cranfield University, Impact Assessement): No APIs to export data specified yet.

3.3.2 Physical Architecture Overview

The IT infrastructure that is needed in Teesside to host the different ROMEO components that take part in this demonstrator is depicted in the figure below. There will be several servers located in different places or countries:

- EDF DMF Server is located in EDF R&D, France, which is the only access point in Teesside architecture to collect data from the field. It is connected to the IBM Cloud through a Security Access Point.
- IBM Cloud will be located in different Data Centers from EU and will host both IBM Statistical models and EDF Mechanical and Electrical Drive Train models.
- O&M Information Management System is located in Austria and will be connected also to the IBM Cloud through a Secure Access Point.

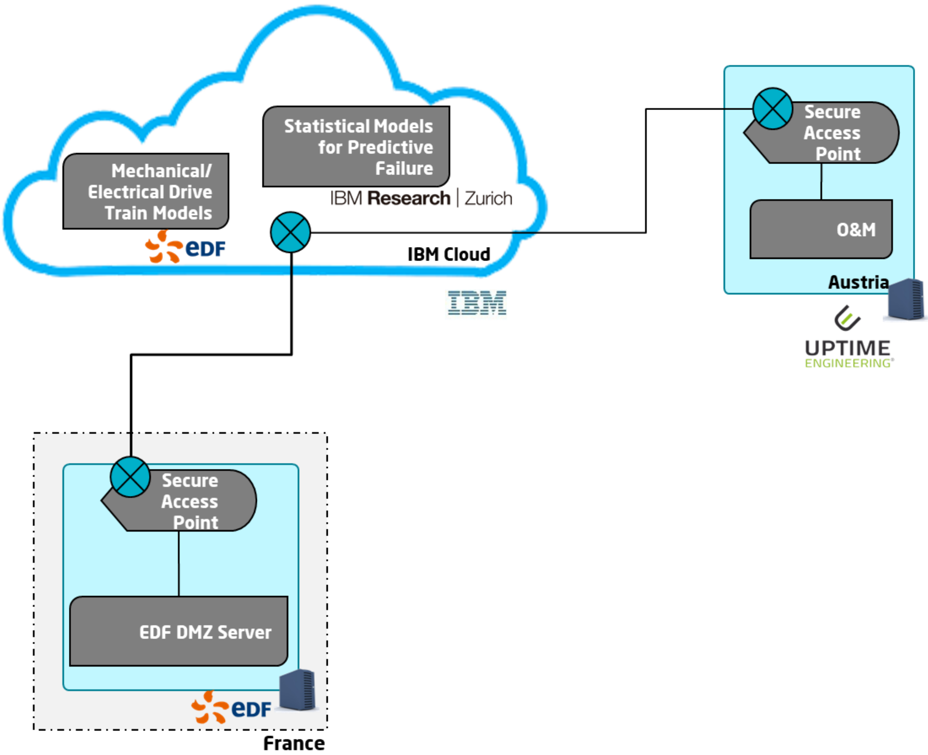


Figure 23: Physical Tesside Arquitecture overview

3.4. Cybersecurity

With regard to the security on the Real Time Platform that would be deployed in East Anglia I and Wikinger demonstrators, the DDS implementation utilized in iSPEED delivers the security, performance and safety required for the deployment of the Industrial Internet of Things. It complies with the new Data Distribution Service (DDS) Security specification from the Object Management Group (OMG). Its main characteristics and benefits are:

- Provides authentication, authorization, non-repudiation, confidentiality and integrity
- Protects discovery information, metadata and data
- Defends against unauthorized access, tampering and replay
- Operates without centralized servers for high performance, scalability and availability
- Runs over any transport including TCP, UDP, multicast and shared memory

Securing critical infrastructure is essential for safety and economic reasons. And it must be pursued without sacrificing performance or reliability and introduces a robust set of security capabilities. These include authentication, encryption, access control and logging. Secure multicast support enables efficient and scalable distribution of data to many subscribers.

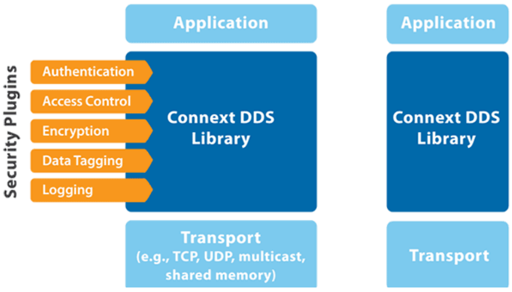


Figure 24: Cybersecurity

Security is implemented above the transport layer and does not require a secure transport protocol such as TLS/SSL or DTLS. Any DDS transport can be used securely, including UDP, TCP and shared memory. Support for UDP multicast (both reliable and best effort) enables very efficient data distribution when there are many subscribers to the same data.

Depending on the security infrastructure that will finally be deployed to connect Iberdrola servers to the IBM Cloud, the needed extra iSPEED security policies for the real-time data exchange in each demonstrator (authentication, access control, encryption or logging) will be configured and implemented in Task 5.4.

The IBM Cloud based IoT platform will be repository of all data to be used for predictive analytics and O&M management for the wind farms. In order to collect data securely, secure access points will be provided to allow communication between the IBM Cloud data consumers and the "on-premise", remote data sources.
Although there are multiple alternatives that can be used to secure communications to and from IBM Cloud, initially, two solutions are considered.

Remote access to containers running in the IBM Cloud, e.g., iSpeed Kafka connector, will be provided using a vpn based on the open-source openvpn. The openvpn server will run as a container in the same cluster as the containers to be accessed remotely. The external "on-premise" server will run the openvpn client using a custom, provided configuration, setting up a secure, encrypted vpn between the external data source, and the IBM Cloud internal data consumer.

Access to remote data sources, e.g., MS SQL databases, from containers running in the IOT platform will be enabled using the Secure Gateway component of the IBM Cloud. This component establishes an SSH tunnel between the container and a remote, “on-premise” port. To enable this connection, a client is installed and configured to on the remote server at a specified port.

As mentioned above, when necessary, other solutions will be considered during the implementation of the IBM Cloud IoT platform.

3.5. Interfaces identified in the architectures

Taking into account the final ROMEO architectures of each demonstrator, all the data exchange interfaces between components to be developed within the project were identified indicating the protocol to be used, the partners involved and the related task where the interface is going to be developed depending on the related WP5 component that is involved in the interface.

Table 3-1 Wikinger Interfaces

Interface	Protocol	Responsible	Implementation
PI Collector (Central level)-BABEL	OPC DA	Iberdrola/Indra	T5.3
Babel (PI Collector)-iSPEED	DDS	Indra	T5.4
iSPEED-IBM Cloud	Kafka-json	Indra/IBM	T5.4/T5.5
Foundation Iberdrola Server-IBM Cloud	SFTP	Iberdrola/IBM	T5.5
SCADA ADWEN SQL Server-IBM Cloud	SQL	Iberdrola/IBM	T5.5
Drive Train CMS Austria Server-IBM Cloud	SFTP	Iberdrola/IBM	T5.5
WP3 Bachman models-IBM Cloud	REST WS	Bachmann/IBM	T5.6
WP3 Adwen models-IBM Cloud	SQL/NoSQL Message Hub	Adwen/IBM	T5.6
WP3 Statistical models for predicting failure	SQL/NoSQL Message Hub	IBM	T5.6
WP4 Ramboll models	SQL/NoSQL Message Hub	Ramboll/IBM	T5.6
WP6 Uptime O&M - IBM Cloud	REST WS	Uptime/IBM	T5.6
Domina G- IBM Cloud	SFTP	Iberdrola/IBM	T5.6
SAP- IBM Cloud	SFTP	Iberdrola/IBM	T5.6

Table 3-2 East Anglia I Interfaces

Interface	Protocol	Responsible	Implementation
PI Collector (Central level)-BABEL	OPC DA	Indra/Iberdrola	T5.3
Iberdrola UCC Server –Node#1	Real-Time protocol	Indra/Iberdrola	T5.3
Babel (PI Collector)-iSPEED	DDS	Indra	T5.4
Node#1 (Iberdrola UCC Server)-iSPEED	DDS	Indra	T5.4
iSPEED-IBM Cloud	Kafka-json	Indra/IBM	T5.4
SCADA [SIEMENS] SQL Server-IBM Cloud	SQL	IBM/Iberdrola	T5.5
Drive Train Central Enterprice Server- IBM Cloud	REST WS	IBM/Iberdrola	T5.5
iSPEED-CEP Engine (Threshol rules)-iSPEED	DDS	Indra/Iberdrola /Siemens	T5.6
WP3 Statistical models for predicting failure	SQL/NoSQL Message Hub	IBM	T5.6
WP6 Uptime O&M - IBM Cloud	REST WS	IBM/Uptime	T5.6
Domina G- IBM Cloud	SFTP	IBM/Iberdrola	T5.6
SAP- IBM Cloud	SFTP	IBM/Iberdrola	T5.6

Table 3-3 Teesside Interfaces

Interface	Protocol	Responsible	Implementation
EDF DMZ Server-IBM Cloud	HTTPS CSV	EDF/IBM	T5.5
WP3 EDF models -IBM Cloud	SQL/NoSQL Message Hub	EDF/IBM	T5.6
WP3 Statistical models for predicting failure	SQL/NoSQL Message Hub	IBM	T5.6
WP6 Uptime O&M - IBM Cloud	REST WS	Uptime/IBM	T5.6

4. Data framework and interoperability

In this section of the report, the data elements to be stored in the IBM Cloud ecosystem, the storage strategies under consideration and the data flows within the ecosystem are described. The platform and the data requiremenets are expected to evolve as the work packages WP3 and to some extent WP4 will begin to use them in 2019. Nevertheless, significant progress has been made in collecting the requirements and enabling the collection of relevant datafrom the wind farms, and these are reported here.

4.1. Subsystems and components

4.1.1 Overview

Based on FMECA workshops of significant faults in wind turbine systems, the following set of 10 components were selected to be targeted by WP3 and WP4, and documented in the deliverable D1.2 (WP1):

1. Blades
2. Blade Bearing
3. Pitch System
4. Main Shaft
5. Gearbox
6. Generator
7. Transformer
8. Converter
9. Yaw System
10. Jacket and Monopile Support Structure

In the following section, the initial work of the WP3 Task 3.1 as reported in the interim deliverable D3.1 [3] (M14) is summarized. It provides the work completed so far in Task 3.1, which is due to complete in M32. Note that not all of the target components are covered by the interim D3.1 deliverable.

4.1.2 List of subsystems and components

4.1.2.1 EDF

Precise description of the required signals including a common RDS-PP designation is being considered.

EDF: Mechanical				
Module	**Subtask**	**Component**	**Instrumentation**	**Failure mode**
1	3.1.1	Main bearing failure early detection	The main bearing is equipped with a specific temperature sensor. Vibration data are not available.	Detect a failure but do not identify/localize it
2	3.1.2	Pitch system failure early detection	Blade pitch position and hydraulic brake pressure can be found in the SCADA data.	Detect a failure but do not identify/localize it
3	3.1.3	Gearbox failure early detection	- Accelerometers placed on different locations of the gearbox structure. The following proposed model will use RMS values of the cepstral analysis. - Oil and Gearbox Temperature sensors, - Particles count sensors.	Detect a failure but do not identify/localize it unless additional design information

EDF: Electrical				
Module	**Subtask**	**Component**	**Instrumentation**	**Failure mode**
4	3.1.3	Transformer - Winding	- Temperature sensor in the transformer - Temperature sensor in the transformer room - Ambient temperature sensor - Voltage sensors - Current sensors - Active and reactive power sensors	Interturn short circuit of windings
5	3.1.3	Transformer - Cooling System	Same as Module 4	Insufficient cooling, multiple cause, ventilator problem
6	3.1.3	Converter	The module will rely mainly on alarms from the converter (electric data of the converters are missing).	IGBT in open circuit
7	3.1.3	DC bus	The module can rely on the DC bus voltage to calculate the capacitor degradation through the voltage ripples and discharge when turbine shutdown. However, the DC bus voltage measurments are not available.	Degradation of the DC bus capacity
8	3.1.3	DC bus	None	Open circuit

EDF: Electrical				
Module	**Subtask**	**Component**	**Instrumentation**	**Failure mode**
9	3.1.3	Generator	High frequency sensors are not available for the moment, only the 10 minutes SCADA data are available. Opportunity to implement additional sensors is not planned for the moment but will be thought on need.	Interturn short circuit of windings

4.1.2.2 Adwen

Precise description of the required signals including a common RDS-PP designation is being considered.

Module	Subtask	Component	Instrumentation	Failure mode
Adwen 01	3.1.1	Gearbox journal bearings	- Two journal bearings support each planet shaft, resulting in 8 sliding bearings in total. Since every journal bearing is equipped with temperature sensors, 16 temperature signals are available for monitoring the bearings condition. - Other sensors, like temperature sensors, capacitive sensing, pressure sensors, flow sensors, displacement sensors, angular speed sensors, acceleration sensors, degrees sensors, slip clutches, low speed switches, overheating protections, gearbox oil level switches, supervisory switches are used on this system.	Gearbox sliding bearings wear/blockage
Adwen 02	3.1.2	Converter	- Voltage and current sensors - Other sensors: water detectors, supervisory switches, angular speed sensors, angular position sensors, flow sensors, pressure sensors, water conductivity, setting of the	DC Link capacitor degradation

Module	Subtask	Component	Instrumentation	Failure mode
			valves, electrical angle, voltages in phases, currents in phases, frequency sensors, pressure sensors can be made available to the module.	
Adwen 04	3.1.3	Generator	- current and voltage sensors mounted in the converter power measurement module PT100 - temperature sensors which are mounted in the stator windings to sense in the winding temperature rises in the coils. - Other sensors: water detectors, supervisory switches, angular speed sensors, angular position sensors, flow sensors, voltages in phases, currents in phases, frequency sensors, pressure sensors can be made available to the module.	rotor magnet demagnetization
Adwen 06	3.1.2	Blade bearing	- angle and angle speed sensors - current flowing through the motors of the pitch will be measured with current intensity sensors. - Other sensors, like rotor pitching moment, voltages, the different torques of the individual blade control, bending and impact moments, supervisory switches, angular speed sensors, angular position sensors of the rotor, spike detectors, temperature switches, limit switches, lubrication related sensors can be made available to the module.	fatigue and wear of raceways detection module for the blade bearing.
Adwen 13	3.1.3	Transformer	- oil level - temperature	Compromised structural integrity (structural fatigue)

Module	Subtask	Component	Instrumentation	Failure mode
			- current - voltage - From other systems, the grid frequency, the accelerometers on the nacelle, the temperature and the humidity on the environment of the section can be measured. Also electrical switches, pressure switches, gas formation switches, supervisory control relay feedbacks, electrical safety mechanisms switches and others are able to be used by the algorithm.	

4.1.2.3 Bachmann

Bachmann : Mechanical				
Module	**Subtask**	**Component**	**Instrumentation**	**Failure mode**
1	3.1.1	Main bearing Gearbox	- MDK10 BS011 - BS001 Sensonics Acceleration Ch 1 - MDK10 BS012 - BS001 Sensonics Acceleration. Ch 2 - MDK10 BS013 - BS001 Sensonics Acceleration. Ch 3 - MDK10 BS014 - BS001 Sensonics Acceleration. Ch 5 - MDK20 BS101 - BS001 JMI Acceleration Ch 6 - MDK20 BS102 - BS001 JMI Acceleration Ch 7	In progress

4.2. Data requirements for WP3 and WP4 models

In this section, the data that will be available from the three wind farms and will be collected, stored and processed within the IBM Cloud ecosystem are documented. The requirements for WP3 and WP4 models will use all or part of the collected data, based on the objectives of the models.

The categories of data to be provided to analytics work in WP3 (predictive analytics and physical models) and WP4 (structural monitoring) from all or some of the wind farms are briefly as follows:

- **SCADA:10-minute average data, electronic, mechanical, weather and power grid sensors.**
- **SCADA (PI):** Based on the PI server, emitted with data values from most SCADA sources when the values change.
- **CMS (Condition Monitoring System)**: drive train vibration data.
- **SHM (Structural Health Monitoring)**: sensors monitoring foundation.
- **Meteorological**: wind speed and direction, temperature and atmospheric pressure.
- **Wave**: wave sensors, e.g., direction, intensity.
- **Events**: alarms, maintenance actions and SCADA (PI) based CEP events.

4.2.1 List of sensor data

The available data is provided for each of the three wind farms, i.e., Teesside, East Anglia I and Wikinger.

4.2.1.1 Teesside

Data	Src Intf	Description	# metrics	Format
SCADA (PI)	TE05 HTTPS	SCADA data from the PI Server (WPS) Mix: PI server (reported at each value change) and 10-min averages	~ 136 (per WT)	• csv file, one per WT • per month
CMS	TE05 HTTPS	CMS for the drive train from the TCM server		• tgz file per power range, per analysis type, per DT component • compressed file contains a *.txt file for each WT
Met-Mast	TE05 HTTPS	Wind speed and direction, temperature and pressure 10 minute averages	24	• csv file • per month
Wave Buoy	TE05 HTTPS	Wave positions, 30-minute averages csv: 30 min avg; hxv: raw buoy msgs; raw: raw buoy displacement data; spt: spectral	9 (csv) 4 (hxv) 4 (raw) 6 (spt)	• one csv file per month • others: a file per day for a month, containing data sampled every 30 minutes

Maintenance Actions	TE05 HTTPS	Maintenance actions	4	• csv file, containing data sampled every 30 minutes for a month
Alarms	TE05 HTTPS	SCADA generated alarms	16	• csv file • per month

4.2.1.2 East Anglia I

Data	Src Intf	Description	# metrics	Format
SCADA (PI)	EA11 Kafka	SCADA data from the PI Server CEP generated events (in progress)	155 (per WT)	• Asynchronous, per metric • JSON format
SCADA (UCC)	EA11 Kafka	SCADA data from the Iberdrola UCC Server		• 155 real time metrics from 1 WTG
SCADA	EA09 SQL (Note: File transfer option under consideration)	10 min average SCADA data	338	• Defined in the example excel sheet
CMS	EA10 REST (Note: File transfer option under consideration)	CMS for the drive train central enterprise server		• tgz file per power range, per analysis type, per drive train component • Files: Cepstrum: 3 files per WT FFT: 11 files per WT Time: 13 files per WT • compressed file contains a txt file for each WT
Domina G	EA12 SFTP	In-house tool to manage operational processes		• csv file(s) • General data: Location (Wind farm, Functional location or equipment), Dates (start and end date), Activity type, etc. • Breakdown information: Alarms, Downtime, Cause element...in case of correctives actions. • Materials used / replaced.
SAP	EA13 SFTP	Enterprise resource planning software used to cover all day-to-day processe		

4.2.1.3 Wikinger

Data	Src Intf	Description	# metrics	Format
SCADA (PI)	WI13 Kafka	SCADA data from the PI Server	47 (per WT)	• Asynchronous, per metric • JSON. format
SCADA	WI11 SQL (Note: File transfer option under consideration)	10 min average SCADA data Trace files		• Daily counters; 10-minute avg; operational modes; and traces • updates daily The list of files/data groups are provided below (•)
CMS		Bachmann drive train CMS data		In progress
Bachmann Phys. Model	WI12 REST	Bachmann physical model output		In progress
Foundation	WI09 SFTP	Foundation CMS data		In progress
Domina G	WI14 SFTP	In-house tool to manage operational processes		• csv file(s) • General data: Location (Wind farm, Functional location or equipment), Dates (start and end date), Activity type, etc. • Breakdown information: Alarms, Downtime, Cause element...in case of correctives actions. • Materials used / replaced.
SAP	WI15 SFTP	Enterprise resource planning software used to cover all day-to-day processes		

The Adwen SCADA data grouped by function is included below:

filename	collection frequency	data columns	type
hubAnalogue.csv	1000 msec	144	trace
hubDijital.csv	1000 msec	31	trace
nacelleAnalogueGen.csv	10 msec	20	trace
nacelleAnalogueHyd.csv	10 msec	31	trace
nacelleAnalogueOth.csv	10 msec	48	trace
nacelleAnalogueTrans.csv	10 msec	52	trace
nacelleDitigalGen.csv	10 msec	3	trace

nacelleDitigalHyd.csv	10 msec	3	trace
nacelleDitigalTrans.csv	10 msec	3	trace
nacelleDitigalOth.csv	10 msec	16	trace
towerAnalogueAux.csv	10 msec	39	trace
towerAnalogueMain.csv	10 msec	61	trace
towerDitigalAux.csv	1000 msec	12	trace
towerDitigalMain.csv	1000 msec	35	trace
CountData.csv	day values	4	day total
OpData.csv	time of occurence	11	day total
hubTenMin.csv	10 minutes	356	10 min
nacelleTenMinGen.csv	10 minutes	84	10 min
nacelleTenMinHyd.csv	10 minutes	124	10 min
nacelleTenMinOth.csv	10 minutes	184	10 min
nacelleTenMinTrans.csv	10 minutes	236	10 min
towerTenMinAux.csv	10 minutes	140	10 min
towerTenMinMain.csv	10 minutes	204	10 min

4.2.2 Data Retention

The retention period for the collected data varies for the applications that will run in the IBM Cloud ecosystem and will also depend on the forthcoming WP3 and WP4 work and their requirements. However, the current intent is to keep up to 2 years of historical data. Whether the historical data will be in original/raw form or aggregated will be decided later, based on requirements and platform costs.

4.3. Data Storage

The data storage strategy in the IBM Cloud ecosystem is driven by the individual needs of the computations to take place in the ecosystem as well as the need to keep the computational and storage costs within the allocated budget.

All data except for SCADA PI metrics reported in East Anglia I and Wikinger will be stored as the original CSV files in the IBM Cloud Object Store (COS), where each file is a sequence of rows containing the timestamp and the associated data columns. These files will be retrieved by or pushed into the ecosystem at set intervals, either as a new file, or merged with an existing file in COS, depending on the requirements of the computations using the data.

The SCADA PI metrics will be reported as a timestamp and metric pair asynchronously, when the value for the metric changes. The values will be kept in files, organized according to the metric and the source, quickly accessible using a mapping table.

The results of the analytics and physical models are to be stored in a relational database, and made accessible to processes running within and outside of the IBM Cloud ecosystem.

4.4. Data and Event Flows

Section 3 describes the data exchange interfaces in each of the wind farms for providing monitoring data to the IBM Cloud ecosystem. For each of the interfaces, an IBM Cloud component will be provided to pull or receive and store the data.

A messaging system will be used within the ecosystem to drive physical model computations and predictive analytics. For example, when new data is received and stored, the corresponding component will issue a message which will be picked up by any service component that needs to use this data, triggering a computation.

Similarly, when there are results available from the analytics computations, these will be stored, and a message will be sent to inform components to forward or to further process the result, e.g., send an even to external O&M components.

5. Conclusions

This document provides the **results** of the work conducted in **task 5.1 Architecture for data acquisition and analytics O&M ecosystem and task 5.2 Information model data interoperability**.

On the one hand, the **overall system and communication architecture** of the 3 ROMEO demonstrators: Wikinger, East Anglia I and Teessideare has been presented. For each architecture component, the use case, a description of the functionality, the inputs and outputs of the component, the interface protocol and their dependencies with other components were described. In addition, the physical architecture as well as cybersecurity mechanisms to be considered in the deployment have been described.

On the other hand, the **data framework** has been presented to ensure the availability of data from partners to enable the computation of physical fault models and the predictive models, including the strategies for storing them in the IBM Cloud ecosystem.

Both results establish the bases for the development tasks to be conducted in tasks T5.3 to T5.6 where the real-time and cloud analytics infrastructure will be developed for ROMEO. The architecture and data framework have been defined with the information that is already available from the pilots as well as from the analytics that are being defined in WP3 and WP4. During the progress of the project and in the development phase, some changes may be conducted for an optimal integration and performance of the whole ecosystem.