Design of an independent smart service platform for wind turbines

Michael Pagitsch, Georg Jacobs, Dennis Bosse, Stefan Kock
RWTH Aachen University, Center for Wind Power Drives, Campus-Boulevard 61, 52074 Aachen
E-mail: michael.pagitsch@cwd.rwth-aachen.de

Abstract. The “Internet of Things” has become more and more common. All kinds of facilities are connected to the world wide web and exchange vast amounts of information. The interconnection of wind turbines and their operators is already state of the art. However, independent and overarching platforms can become providers for additional smart services – information generated from available datasets and action plans derived from it. The present paper lists key requirements for the design of such platforms and proposes solutions to meet the requirements.

1. Motivation
Throughout the past few years, the “Internet of Things” (IoT) [1] and “Internet of Services” [2] have increasingly become powerful and important means for interconnecting technical systems. An exchange of information between customers and providers of services of many kinds takes place in real-time. This allows for a new variety of business models and increases the reaction speed to customer requests [3].

The Center for Wind Power Drives in Aachen (CWD) aims at facing the challenges and setting up a prototypical smart service platform together with its research partners. The IoT platform is intended to facilitate the exchange of information between OEMs and suppliers, operators, service providers, research facilities, and entities in the electricity market while dealing with datasets from wind turbines (WTs) of all types. By experimenting with the prototype, potentials of various use cases for generating and utilising information from large amounts of data will be assessed. Even in an early phase of development, crucial design decisions must be taken meaningfully in order to allow for maximum flexibility and scalability. The present paper aims at presenting the most important questions and solution approaches for setting up an IoT platform.

- Handling of heterogeneous datasets: How can disparate datasets be stored efficiently? Which are suitable data structures and formats?
- Data links: How are data transferred between the different participants? How can plausibility be ensured?
- Data centre architecture: Which hardware and software architecture guarantees for flexibility and scalability?
- Disposal of evaluation results: How can processed data be stored and transferred?
• Concepts for utilisation and authorisation: Which roles can be assigned to the individual participants? To which extent can data be anonymised?
• Smart service cases: Which use cases are beneficial for individual participants?
• Procedures for data analysis: Which analytical methods are required for respective service cases?
• Physical models: Which physical models are required for respective service cases? How can data and digital twins of wind turbines be combined?
• Edge computing: What is the benefit from pre-processing data onboard the WT?

2. Requirements and how to meet them
In the following sections, several of these questions are addressed. Possible solution approaches for the design criteria of the platform are discussed.

2.1. Heterogeneous datasets
Different and heterogeneous datasets have to be stored, as they form the base for all further processing:

• WT reference data document the design and designated operational conditions. They are given by data sheets and references to standards.
• Data from the supervisory control and data acquisition (SCADA) system constitute the core of all datasets and stem from the control system of WTs. Usually, data are available with a temporal resolution of 10 minutes. Special events (emergency stop, grid failures such as LVRT – low voltage ride through i.a.) are listed and documented in the SCADA dataset.
• Snippets of time series in high resolution from CMS (prevalently from vibro-acoustic sensors) may help to detect failures and interpret mechanisms.
• The schedule and history of maintenance activities provides additional context for the data mentioned above.

In order to store these fundamentally different types of datasets, all data have to be brought into standardized, homogenised, and machine-readable formats. A decision between different kinds of databases has to be made. The CAP-theorem helps with selecting the right database for meeting the demands of each dataset. According to its creator Brewer, databases can be characterised by the properties of consistency (C, data delivered to the user are always most recent), availability (A, high accessibility and performance are ensured), and partition tolerance (P, incorrect system responds can only be caused by total network failures), see fig. 1. As they can never satisfy the demands for consistency, availability, and partition tolerance, databases are usually designed and optimized with respect to two of the three paradigms.

In the present use case the database is primarily a repository with more read requests than writing processes. Therefore, “consistency” is regarded less important as compared to availability and partition tolerance, which guarantee for scalability. Apart from that, traditional SQL databases are not an option due to the need for flexibility considering the possibility of future changes in available datasets or applied data models. Among the database management systems designed with respect availability and partition tolerance, Apache Cassandra is selected as the standard database for the smart service platform.
As the SCADA dataset constitutes the core of many evaluation strategies, the respective data model will be described a bit more in detail:

SCADA data contain information on the overall operating state of a WT. IEC provides a standard for the SCADA dataset [6], which allocates the individual data channels to classes related to the topology of the WT – see table [1]. These data classes give information on the current operating state. Besides, the standard mentions four more classes (WTUR – Wind turbine general information, WSLG – Wind turbine state log information, WALG – Wind turbine analogue log information, WREP – Wind turbine report information) that contain data on the turbine itself as well as documentation of failures and maintenance activities. These classes are not part of the core SCADA data. The data are commonly recorded as mean values of time intervals of 10 minutes. In order to complete the picture, the standard suggests also to store minimum, maximum, and the standard deviation. The data model for SCADA used at CWD is based on the standard. As OEMs frequently do not stick to the recommendations given in the IEC standard, processes for data conversion are defined. Accordingly, evaluation algorithms can be developed with respect to applicability to a single dataset.

| Class   | Description                      |
|---------|----------------------------------|
| WROT    | Wind turbine rotor information   |
| WTRM    | Wind turbine transmission information |
| WGEN    | Wind turbine generator information |
| WCNV    | Wind turbine converter information |
| WTRF    | Wind turbine transformer information |
| WNAC    | Wind turbine nacelle information |
| WYAW    | Wind turbine yawing information |
| WTOW    | Wind turbine tower information |
| WALM    | Wind turbine alarm information |

Table 1. Classes of SCADA channels according to IEC 61400 [6]

2.2. Data links, security, and integrity
A connection to the individual WTs can either be established directly or via links provided by third parties. WT data are either gathered from a device inside the turbine (which is capable of onboard data pre-processing) or via a well-defined interface to a data storage. Before storing the obtained data, their conformity with defined standards as well as their plausibility has to be ensured. These terms are related to quality assurance of data inserted into the platform’s storage: “Conformity” addresses the formal aspects by enforcing the implemented
standards; “plausibility” describes the quality related to content. If the principle of plausibility is violated, the respective data points are tagged. Insertion in the database is not prohibited, as the plausibility checks might simply have met the description of an unexpected state. In this case, assessment by a human is necessary in order to improve the algorithm. Furthermore, data points tagged “implausible” can easily be excluded from subsequent processing.

If possible, these requirements are met by data pre-processing onboard the WT or application of harmonisation algorithms (i.e. establishing conformity) before loading the data into the storage system. Integrity and secure transmission are guaranteed by utilisation of state-of-the-art encryption mechanisms, checksumming, authentication as demanded by the transport layer security protocol (TLS [7]), and network configurations such as virtual private network (VPN).

2.3. Data centre architecture
Various and arbitrary evaluation algorithms are fed by different subsets of available data. These processes have to run broadly automated. Algorithms, which need to be updated and replaced from time to time, must not endanger or even break the smooth process of generating information from data.

To ensure the claimed procedures, evaluation algorithms need to be designed according to predefined standards which make them “plug and play”. These standards have to be enforced by the components of the platform.

The hardware hosting the platform is subdivided into purpose-dedicated virtual machines (VMs), the state of which is supervised and monitored by superior control structures. Thereby, horizontal scalability (facilitation of growth in terms of data volume and number of involved parties) is guaranteed by design.

2.4. Disposal of evaluation results
Evaluation results need to be made available for involved parties (see fig.2) – either directly in visual form or indirectly in machine-readable data formats that can be handled by individual post processing algorithms. For visual data presentation, a web frontend is available. It is designed in a fully dynamic and scalable way. For handing over result data in a machine-readable form, current standards such as xml [8], json [9], or google protocol buffers [10] are used. These formats allow for customisation and adaption with respect to the intended use of contained data.

3. Basic architecture of the platform
Figure 2 gives an overview on the global architecture of the platform derived from the requirements discussed above. The core is constituted by a data storage that holds source data as well as results of calculations. The storage is enclosed by a middleware which is the interface to data links and evaluation algorithms provided and deployed by the different involved parties, see sec. 1. Those algorithms, or smart services, are so-called “agents” or “bots” – more or less complex computer programs that are acting autonomously and perform different operations on the data in the storage in order to generate beneficiary information. They are virtually connected to the middleware and contain individual interfaces allowing for user-side configuration as well as graphic display of evaluation results.

The middleware is a bundle of software dedicated for

- establishing end points of data links,
- providing encryption and decryption services,
- administrating access privileges,
- and providing appropriate documentation.
Figure 2. Basic architecture of the smart service platform. The core contains the data storage and is enclosed by a middleware. In the outer layer, smart services provided by involved parties perform operations on the data.

Due to the open standard (to be published yet) of the middleware, the list of involved parties can be extended arbitrarily. In addition, there is no fundamental limitation to quality and quantity of smart services, as long as they stick to the standard demanded by the middleware. The data centre architecture, according to which the hardware configuration is organised, is completely hidden from the end user.

4. Exemplary applications
As mentioned above, various applications can be integrated into the platform to perform data analyses. The following section gives two general examples of ingredients of smart services. These examples do not provide explications for the causes of turbine problems. However, they can be useful means for indicating non-obvious problems.

4.1. Dynamic thresholds for condition monitoring
Various signals from the SCADA dataset are used for condition monitoring. In gearboxes, the temperatures of bearings are critical values, as they indicate nascent failures, if they exceed certain values. Those values depend on the current state of operation and can hardly be determined analytically. A dynamic calculation of threshold values corresponding to individual operating states enables for detection of failures before they become catastrophic. One possible way to calculate suchlike thresholds is the use of machine learning algorithms.

In the present example, a prediction model for the temperature of a high-speed-shaft bearing has been set up. It consists of a neural network that has been trained to calculate the bearing temperature from nine further parameters comprising the current wind situation, power generation, and thermal behaviour of the WT. During operation, the temperature value calculated by the neural network can be compared to the corresponding measured value. A growing difference between the two values implies a high probability of upcoming failures. Figure 3 shows a comparison of measured and calculated temperature values over a period of four months. Both curves are broadly congruent (overall RMS error 2.3 %). The largest deviations (at the downward-facing peaks in fig. 3) occur during standstill-periods of the WT and are not relevant for the purpose of condition monitoring. As damages leading to system
failure occur frequently in the bearing considered in the example, the relevance for practical use is evident.

![Figure 3. Comparison of measured and calculated values of a bearing temperature](image1)

4.2. Performance benchmark of WTs in a wind farm
Observing the power curve of single turbines or wind farms has various benefits. The power curve, which constitutes the connection between electric power output of a WT and the wind speed at hub-height, is an important means for e.g. analyzing control and performance of WTs and therewith identifying underperforming turbines within a farm. This can simply be done by comparing the normalized power curves of neighbouring turbines. Fig. 4 depicts the reconstruction of the power curve from measured data (left; method: according to [13]) as well as a comparison of four WTs within the same farm (right). The comparison shows, that the performance of the individual WTs is slightly different in the range of partial load, although it should be equal in theory. WT 1 outperforms the remaining three with WT 4 being the weakest. This is a starting point for further investigation. Possible causes for the difference in performance might be caused by differing configuration of the control system, by the terrain or shadowing effects, or even by badly adjusted sensors. Plugged into a platform, the analysis of power curves can easily be extended across WTs in different wind farms.

![Figure 4. Left: Extraction of the power curve from measured data; right: comparison of several power curves of similar WTs inside a wind farm](image2)
5. Conclusion

Some important design parameters of a platform have to be adjusted to the right values from the beginning – once a data model is chosen it can hardly be replaced or be subjected to other than gradual alterations in a productive system. Impeccable data integrity and security are the prevalent design criteria and must not be violated by all other parts of the platform. They are closely followed by the demand of efficient utilisation of IT resources – which requires the right choice of tools for respective tasks. Quality and standardisation of available data is an enabler for ensuring the universal applicability of various evaluation algorithms.

A system as described above has to be designed with respect to simplicity, safety, and independence from individual participants. Only if these key requirements are met in an appropriate way, a platform for smart services is able to provide an immediate benefit.

Acknowledgements

The authors would like to thank the German Federal Ministry for Economic Affairs and Energy for the financial support of this work.

References

[1] ITU-T Y2060 2012 Overview of the internet of things
[2] Soriano J, Heitz C, Hutter H P, Fernandez R, Hierro Juan J, Vogel J, Edmonds A and Bohnert T M 2013 Internet of Services pp 283–325 ISBN 978-3-642-41568-5
[3] Arbeitskreis Smart Service Welt 2015 Umsetzungsempfehlungen für das Zukunftsprojekt Internetbasierte Dienste für die Wirtschaft
[4] Brewer E 2012 Computer 23–29
[5] Lakshman A and Malik P 2010 Cassandra: a decentralized structured storage system vol 44 pp 35–40
[6] IEC 61400-25-2:2006 Wind turbines – part 25-6: Communications for monitoring and control of wind power plants – information models
[7] RFC 5246 2008 The transport layer security (tls) protocol
[8] W3C 2008 Extensible markup language (xml) 1.0 (fifth edition) URL https://www.w3.org/TR/REC-xml/
[9] RFC 8259 2017 The javascript object notation (json) data interchange format
[10] Google Inc 2008 Protocol buffers URL https://developers.google.com/protocol-buffers/
[11] RFC 2768 2000 Network policy and services: A report of a workshop on middleware
[12] Wooldridge M 2001 Intelligent Agents: The Key Concepts (Springer) pp 283–325 ISBN 3-540-43377-5
[13] IEC 61400-12-1:2005 2005 Wind turbines – part 12-1: Power performance measurements of electricity producing wind turbines