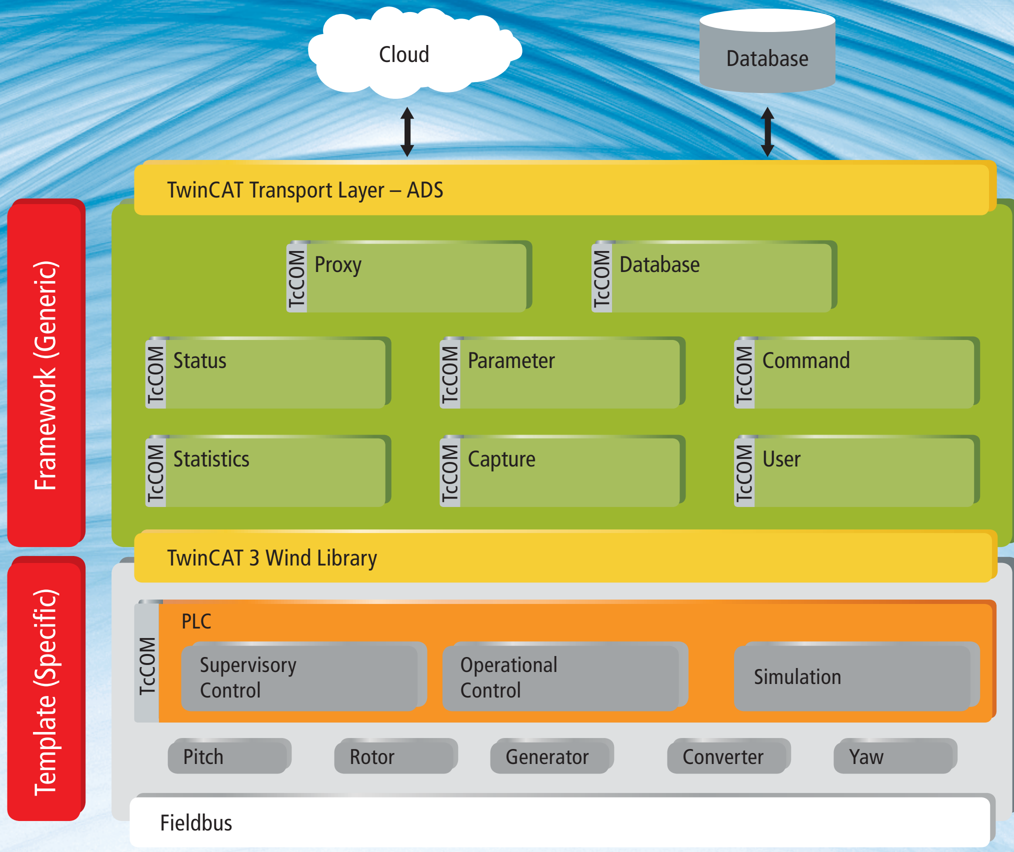
TwinCAT 3 Wind Framework for wind turbine automation

1 Framework and 10 years of expertise through 40,000 installations

The new TwinCAT 3 Wind Framework will enable manufacturers of wind turbines to program their systems quickly and easily on their own. All functions are integrated into one universal software package: from event management to database connectivity, and even basic functions such as state machine and hydraulics. A prefabricated application template considerably simplifies the programming process, enabling developers to concentrate on the essential system functions. The result: efficient engineering, shorter time-to-market, and the benefits of Industry 4.0 for the wind industry.

© AREVA Wind/Jan Oelker



Application templates and encapsulated modules enable modular software architecture with high functionality

Beckhoff has offered advanced wind industry solutions for over 16 years. TwinCAT 2 Wind libraries have been tested and proven in a wide range of applications, offering users a robust basis for the development of software in the operational management of wind turbines.

The ever-accelerating development rate of increasingly larger wind turbines creates new challenges: more intelligent systems with additional sensors and actuators are used, further increasing the complexity of the systems. This makes fault analysis much more complex and stable operation increasingly difficult. To address these challenges, the existing concepts and technologies were combined, resulting in the development of the new TwinCAT 3 Wind Framework.

Comprehensive functionalities are implemented in encapsulated TwinCAT modules, which are then integrated into the TwinCAT 3 architecture. Efficient software development is ensured through a modular architecture in the application template, as well as through proven and directly applicable TwinCAT modules and functions. The flexible configuration makes adaptation to user-specific application requirements very straightforward. System diagnostic functionality is ensured by means of comprehensive data storage capacities in a database. The benefits include future-proof development, efficient commissioning, and optimum operation of wind turbine automation software.

TwinCAT 3 offers the option to implement IEC 61131-3, C++ and MATLAB®/Simulink® modules, load them into different CPU cores, run them in different real-times, and enable them to reliably interact with each other. The basis for this is the TwinCAT module language, which describes the characteristics of the TwinCAT modules, e.g. with regard to the process parameters or the methods.

Programming operational management software based on template and function library

The programming of operational management software using the TwinCAT 3 Wind Framework is facilitated by a library and an application template. The library provides all functions of the Wind Framework as PLC function blocks. The application template provides a modular architecture for the operational management software for wind turbines in the form of a PLC project, through which all the options found in the TwinCAT modules and functions are implemented.

A simplified diagram of the application template is shown in Figure 1. Each subsystem of the wind turbine (such as pitch control or converter status) is represented by a discrete object. In this way, the subsystems (such as pitch, converter, etc.) can be developed, used and tested independently. The subsystems now also feature interchangeable software, as is already common practice in the mechanical modularization of systems. The modularization enables parallel development, and programmers can focus on the specific functions and system components they work on. This increases the quality, flexibility and reusability of the software, while at the same time reducing development time and engineering costs.

The different operating modes for starting and stopping, and the higher-level state machine of the system, are consolidated in the application template as supervisory control and implemented in simplified form as PLC function blocks. This results in higher-level set values for operating the system, which are used for control purposes.

General control functions of the wind turbine, such as pitch and torque control, are prepared in the software as operational control. For these control purposes the integration of other modules is intended, for example to take over the algorithms from the load calculation.

These options include the automatic generation of a TwinCAT module from MATLAB®/Simulink®, or integration of control algorithms via C/C++. Thus, the same controller that is used for load calculation is also used for general control purposes. The controller does not have to be converted to a second programming language, and the error-prone second implementation of the algorithms is no longer required.

TE1400 – TwinCAT 3 Target for MATLAB®/Simulink®
The TwinCAT 3 Target for MATLAB®/Simulink® enables the generation of real-time capable TwinCAT modules from a MATLAB®/Simulink® model. These can then be executed in the TwinCAT 3 runtime, instantiated several times, parameterized and debugged, without the need to use MATLAB®/Simulink®. Transferring a whole block diagram from Simulink® into the TwinCAT module makes it possible to analyze and optimize the controller. This optimization can take place in the field and directly at the system, since only TwinCAT 3 Engineering is required to parameterize the controller.

The demanded values from the operational control are transferred to the subsystem control, where the individual subsystems are controlled. Each subsystem is implemented as a PLC function block in the form of a module with five methods

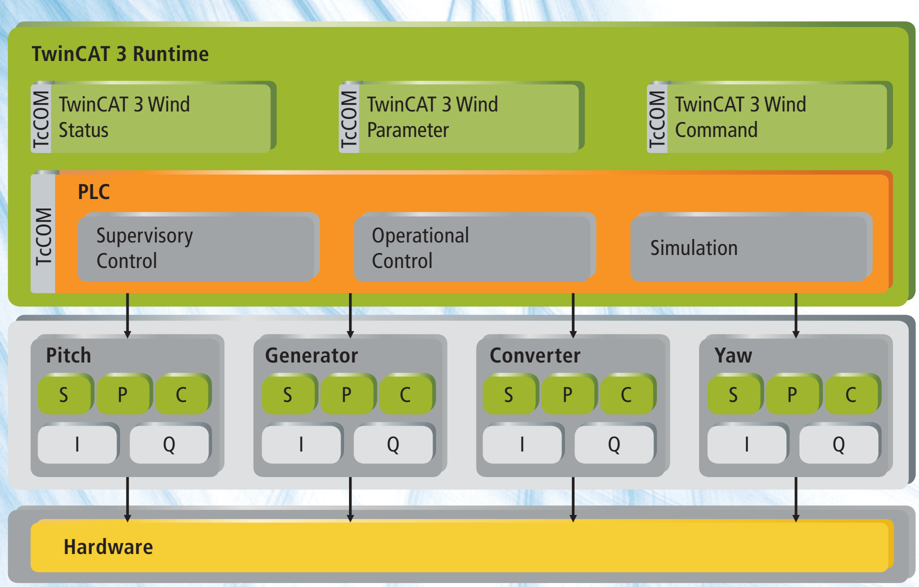


Fig. 1: Operational management is implemented as a standalone TwinCAT module, and the subsystems are implemented as independent objects.

and four data structures. The uniform architecture of the subsystems is shown in Figure 2. The Input/Output data structures are linked to the hardware inputs and outputs, and contain the numeric value of sensors, actuators, and fieldbus systems. The numerical data from the Input data structure are pre-processed in the InputUpdate method and transferred as physical values to the Inbox data structure. The methods ActualUpdate, MonitorUpdate and ControlUpdate are called for monitoring and controlling the system. They directly access the physical values from the Inbox. The new values for controlling the units are written into the Outbox data structure as physical values. The Outbox data structure is post-processed in the OutputUpdate method and provided as a numerical value. Interfacing with the higher-level supervisory control takes place via the Actual and Demand data structures. In addition, the Inbox and Outbox data structures enable simple simulation of the systems based on the physical values. In this way, each subsystem introduces its own simulation into the overall system.

Furthermore, an adaptive simulation of a 5 MW offshore wind turbine is integrated in the application template, which is preconfigured by the reference turbine of the National Renewable Energy Laboratory (NREAL). This enables testing of the entire operational management in the development environment, as the model is adaptable and configurable to match the respective system. The system simulation is provided as a TwinCAT module, although just like the control itself, it is ready to be replaced by a specific model from MATLAB®/Simulink® or C/C++, as required.

The integrated simulation in the application template can be used to reproduce and evaluate the processes of the whole system, as well as the individual operating modes and subsystems. Each subsystem can be operated separately and independently by switching between the simulation and the actual hardware. In this way, it is possible to activate nacelle components, for example, on the factory floor for testing. In addition, test benches can be configured for software-in-the-loop or hardware-in-the-loop simulations, and even for training sessions directly with the original application software. Real-time simulations enable rapid control prototyping and virtual commissioning with a single version of the software, based solely on parameterization.

The operational management and subsystems are complemented by using the available TwinCAT modules from the TwinCAT 3 Wind Framework. Objects are created and configured via PLC function blocks from the PLC library. These objects integrate automatically into the higher-level TwinCAT modules from the Wind Framework, which provides the necessary services and functions. In this way, each subsystem defines an individual set of objects which contribute the information and settings for operational management.

The consistent use of the TwinCAT modules and the uniform architecture of the subsystems create an application standard. This standardization enables programmers to quickly familiarize themselves with the application and the source code, even if it was implemented by another programmer.

Generic modules for higher-level services

The generic TwinCAT modules provide the higher-level services, providing the ability for each module to be used directly, integrated in TwinCAT 3 as a TcCOM module. The modules can also be used separately and independently of each other or in combination, in order to facilitate interaction and data exchange. Figure 3 highlights the available modules.

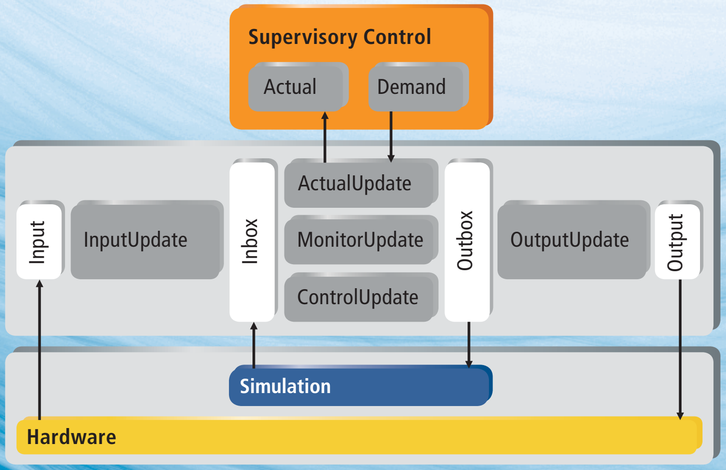


Fig. 2: The uniform architecture of the subsystems creates a standard in the application and enables quick familiarization by programmers.

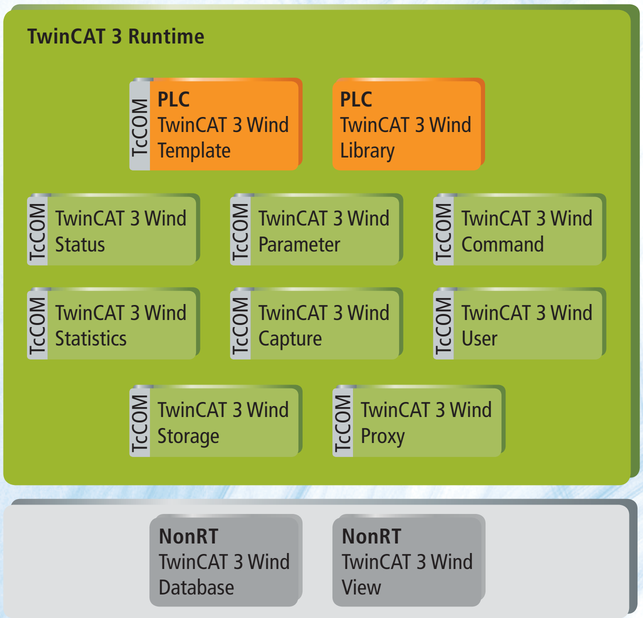


Fig. 3: The TwinCAT 3 Wind Framework provides functions via TwinCAT modules, as well as a PLC library, the database interface, and visualization.

All the information from the different TwinCAT modules is transferred permanently and in real-time to the Database module. This data is managed in a database via the TwinCAT 3 Database Server: it is prepared by the Database module as required and added to, or retrieved from, the database via SQL commands. The interface with a Microsoft SQL Server is fully implemented by the Database module. The Wind Framework provides the corresponding database scheme, including the tables and procedures for the Microsoft SQL Server. The extensive use is reflected in the modules described below.

TF6420 – TwinCAT 3 Database Server
The TwinCAT 3 Database Server enables data exchange from the TwinCAT real-time environment with different databases. SQL commands such as Insert or Select can be used, as well as Stored Procedures. Currently, 11 databases are supported, including Microsoft SQL, MySQL, PostgreSQL, and Oracle. In addition, a configurator for the visual setting of the parameters is included as a PLC library, which provides function blocks to run the SQL commands.

In automation applications, it is customary to cover the user administration in an external visualization or the SCADA system. User administration is integrated into the functions of the TwinCAT 3 Wind Framework, so that the user can check, manage, and record all interactions via the User module. In this way, it is possible to specify during programming which rights are required to use each function. These user rights are checked in the application so the operational management can automatically ensure correct user access, independent of an external management system.

Secure storage of accounts is ensured through cryptographic functions (hashes) in the database. Authentication is based on a name and password for activating the stored access level. The local access privilege can be determined through service switches at the control cabinets. It indicates whether user access is taking place remotely or locally, the latter resulting in higher rights.

The Proxy module provides direct access to the real-time data of all modules and objects. This access can take place directly, via the TwinCAT ADS protocol, to retrieve any features of an object or the recorded data. Safe and vendor-independent communication via OPC UA or standard-compliant communication from IEC 61400-25 is enabled through further TwinCAT 3 functions.

TF6100 – TwinCAT 3 OPC UA Server
The TwinCAT 3 OPC UA Server enables communication based on the OPC Unified Architecture (IEC 62541). As a precursor for Industry 4.0 and the Internet of Things, OPC UA ensures secure, reliable and vendor-independent transfer of raw data from the sensor in the production level to the IT level and the ERP system. In this way, the control system enables object-orientated data communication for current and historic data, alarms and services (methods), and makes them available in a Service-oriented Architecture (SoA).

TF6510 – TwinCAT 3 IEC 61850/IEC 61400-25
The TwinCAT 3 IEC 61400-25 function enables data exchange based on IEC 61850 from the objects specified in IEC 61400-25 for wind turbines. The Manufacturing Message Specification (MMS) is the protocol implemented for communicating hierarchical data objects between the wind turbine and a control center. The TwinCAT Telecontrol Configurator, which is included, provides support during configuration of data models and generates the respective PLC code.

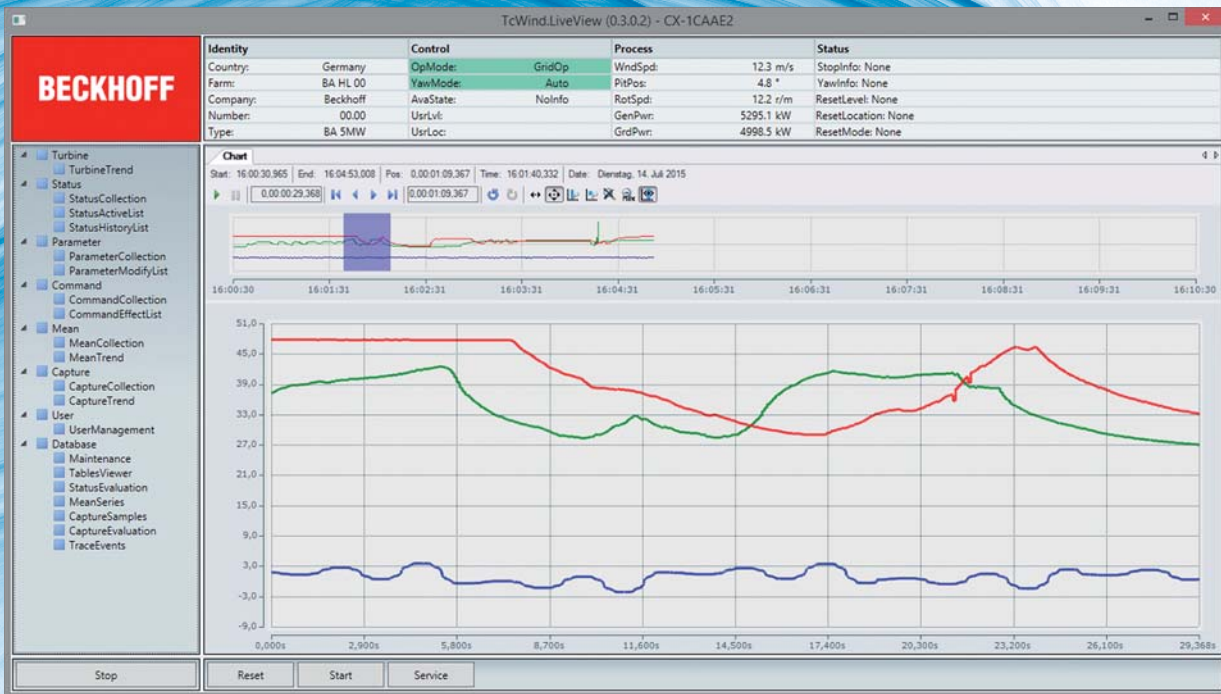


Fig. 4: The visualization is used to display current states, values, and settings.

With the TwinCAT 3 Wind Framework, a simple engineering visualization is available. It retrieves the data from the Proxy module and leverages the user administration integrated in the User module. The current states, values, and settings of all objects can be displayed. Simple reports and database analyses are possible, and the histories of the recorded data can be visualized. Figure 4 shows an example of such a diagram based on recorded data from the database.

A further service is provided by the Status module, which enables monitoring of all wind turbine components, provides error detection, event management, error handling and reporting. Status objects are created, each of which representing an event, and these are used for displaying individual messages, warnings, or errors. Examples include broken-wire sensor warnings or unit malfunction error messages.

A Status object has various configurable properties. For identification, each object is allocated in a group which corresponds to the respective system component, and a name in plain text. Delays for setting and resetting events, as well as various modes for automatic or manual resetting, are possible. For example, error resetting can be limited to an authorized group of persons by setting an access level and a local access authorization. A specific example would be limiting an error reset to a manual action by a service technician, and only if the technician is actually on site and in the nacelle.

Additional features include the option to set a system stop for each event as a system response, as well as triggering high-resolution logging of the system data or sending notifications. This flexible configuration of events is evaluated by the Status module, and the appropriate responses are generated from the current state of all events. An error in the pitch or the converter, for example, can be evaluated such that the system stops in response. Safe operation of the wind turbine is guaranteed through higher-level monitoring of all events. A list of all currently active events and a history of recent events is managed by the Status module, through which the events can be retrieved at any time. Each event is recorded in the database and furnished with a timestamp, which indicates when the event occurred and when it was reset. This allows the user to determine the frequency and duration of each event, and to draw conclusions about the operation and the availability of the turbines. In addition, statistical analysis of the events can be used to determine the most common causes for any downtime, providing a basis for optimization.

The Parameter and Command modules provide services for configuration and interaction with the application. A Parameter object can assume any value of any data type, and all data types from the IEC 61131-3 standard are prepared in this way. Arrays can also be used as vectors or tables. For example, temperature monitoring limits can be implemented in the form of two Parameter objects, whose values indicate the minimum and maximum temperature or are used for switching the heating on or off. The value of a parameter can be limited through the properties of the Parameter object, and the options for changing the value can be limited via an access level. In addition, a default value is set for each parameter, which can be reset, if required.

In this way, the entire wind turbine configuration can be mapped and modified via parameters. Logging of all parameter changes, and persistent storage and loading of configurations, is enabled via the database connection through the Database module. The entire configuration of the wind turbine is thus stored in the database and can be matched to the configurations of other turbines.

Command objects can be used to trigger or activate actions in the application. Each interaction, for example via a switch at the control cabinet door or a button from the visualization, can be implemented via a Command object. Here, different modes for implementation such as pressure switch, toggle switch, or dead man's switch are offered. Their operation also requires authorization and is fully logged in the database. For better diagnosis and visualization, each Command object evaluates a confirmation and feedback. Using the protocol in the database, it is possible to ascertain at any time which operator triggered a system stop or an event reset and at what time, as well as changes of individual parameters.

Signal logging, as well as a statistical analysis, is provided by the Capture and Mean modules. Raw data are flexibly recorded via Capture objects, and the signal type (digital or analog) and the sampling rate for logging are set individually for each Capture object. Initial evaluations for subsequent diagnoses are carried out in real-time. The number of changes and the duration of the active state are evaluated for a digital signal. In this way, it is possible, for example, to monitor a unit and its behavior and to read out the switching frequency and operating time. When a unit is replaced, the service technician can reset the statistics manually. The optional integration of analog signals enables calculation of values such as flow rate, power generation, or consumption.

If such evaluations were carried out outside the real-time, there would be significant deviations from the actual values, due to non-deterministic recording and analysis of the information. The data determined in this way are permanently stored in the database, although they can be retrieved and used as instantaneous values.

The Mean module enables continuous determination of floating mean values; these Mean objects can calculate mean values from any analog signals over freely selectable time intervals. The properties offered include arithmetic mean, RMS value, standard deviation, and mean value calculation based on the wind direction, as well as minimum and maximum or turbulence intensity. The time interval over which the average value is logged in the database can be freely configured. Typical values for calculation and archiving are 30-second or 10-minute average values. In this way, it is possible to generate statistics such as the power curve, wind rose, or a capture matrix from the database over any period and on demand.

For significant events, which are described by a Status object, a Trace module can be used to trigger writing of a high-resolution protocol of the system data. The system data are provided based on the Capture objects and logged by the Trace module in cycle time. This protocol also includes data from configurable periods, before and after the time at which the event was triggered. Data from several seconds prior to the event is available and can be used for troubleshooting. It is also possible to check how the system responds to the event, since data from several seconds after the event is available.

Each determined value and each event are furnished with a timestamp. The timestamp is used by the local system and can also be obtained from a synchronized time source. The Time module is available for this purpose. It uses EtherCAT Distributed Clocks, in order to access a global synchronized time source via IEEE 1588 or PTPv2, for example. The cycles for logging signals via Capture or Mean objects are also synchronized based on this time. 10-minute average values are logged at fixed 10-minutes-interval at 11:00, 11:10, etc. (i.e. not at 11:03, 11:13, etc.). This ensures that data from different sources and systems are truly comparable, since all values were determined at the same time, and at identical intervals.

Database connectivity for detailed analyses

Interfacing with the SQL database via the Database module and the TwinCAT 3 Database Server offers efficient and compact data management, based on a uniform and familiar format. Logging of all events and signals, enhanced by the storing and loading of the entire configuration of all objects, enables detailed analyses. Any preprocessing required for this takes place in the TwinCAT modules in real-time. The Mean module calculates the mean values consistently in each application cycle, and each value from each cycle is used for averaging. The Capture module evaluates the scanning of values and integrations in each cycle, in order to make the calculation as accurate as possible.

Logging and preprocessing of all data in real-time, followed by reliable transfer to the database, forms the basis for evaluations on demand and outside the real-time environment. Based on this historic information it is possible to detect state changes and the causes of faults, calculate detailed statistics, and optimize the system. The database scheme is prepared in such a way that the data from individual or multiple systems can be collected and managed in a single database. The data can then be easily merged via prepared procedures, for higher-level analyses and comparisons. The merging of data is shown schematically in Figure 5.

If the data from all systems are consolidated on a central company server, or in the cloud to form a data warehouse, it is possible to keep the data permanently and over the complete lifetime of the systems. Such data quantities from any number of systems, which are generated in real-time and accumulated on central servers, can generally be referred to as Big Data. Big Data is a further building block towards Industry 4.0 and is supported by the option to integrate additional data from windfarm management or from monitoring and measuring systems.

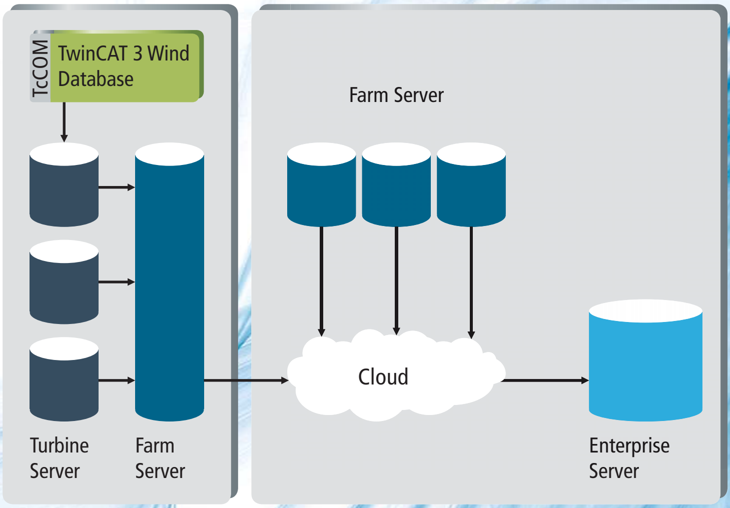


Fig. 5: The system data can be consolidated on central servers for higher-level analyses.

Uniformly accessible, these data allow extensive and automated evaluations. They can be used to detect faults or irregularities, calculate statistics and optimize the operational management, as well as enabling condition-based monitoring and predictive system maintenance. Data mining can be used to gain new insights into system operation. For example, it may be possible to determine relationships between component wear and their switching frequency and operating cycles, so that future components can be replaced before they fail.

Conclusions

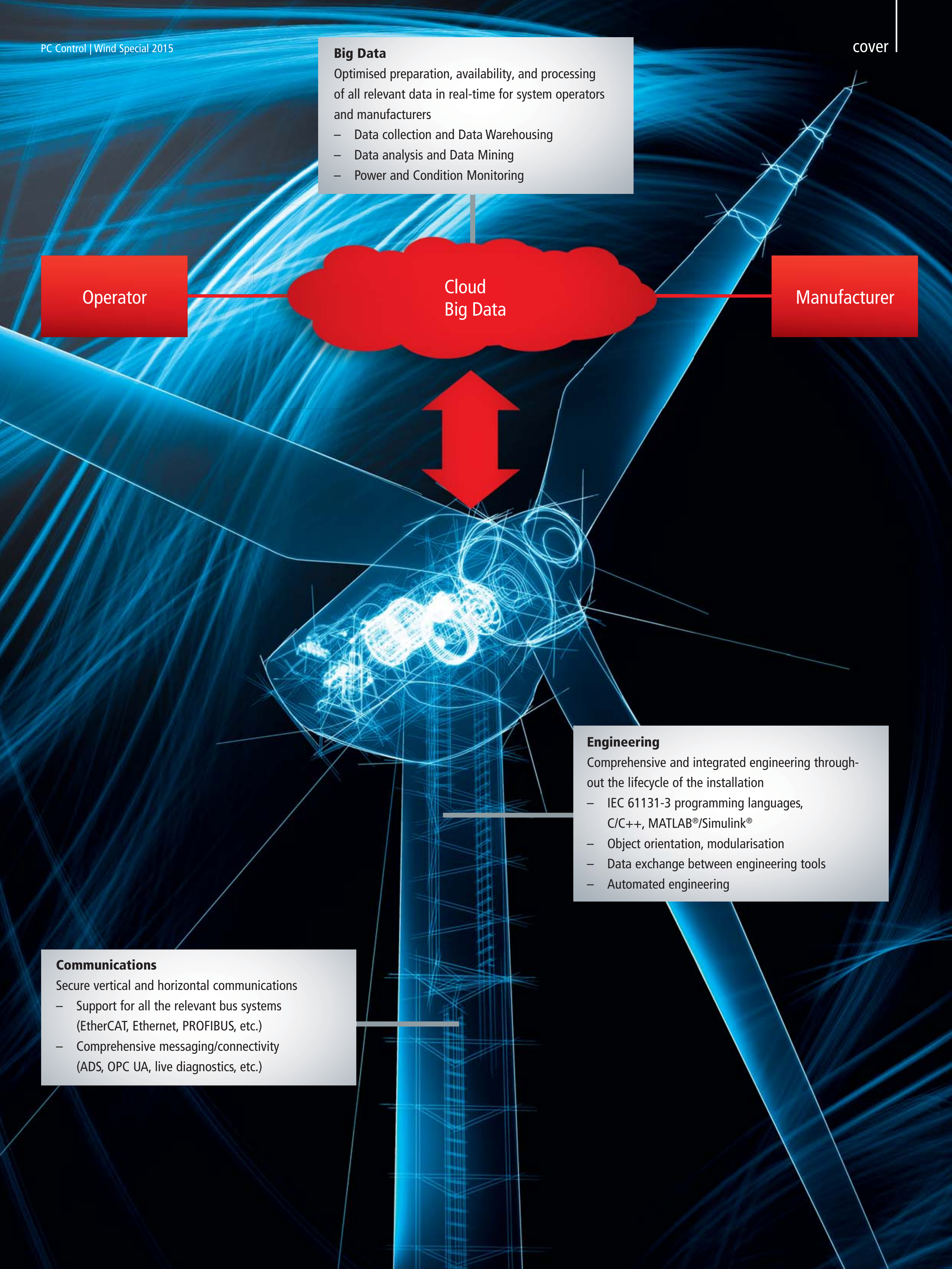
Advanced software engineering and Service-oriented Architecture (SoA) concepts were implemented in the TwinCAT 3 Wind Framework. The broad product portfolio offered by Beckhoff is thus extended with a TwinCAT 3 function library for wind turbines. With the modular architecture, the communication interfaces, the database connection, and the option to store all data centrally, the prerequisites for Industry 4.0 for wind turbines are achieved.

TwinCAT modules provide comprehensive services, in which essential operational management, user administration tasks, as well as comprehensive logging and data management, are already implemented. Programmers can use these functions via PLC function blocks and focus on their actual tasks.

The application template provides a ready-made, modular software architecture, prepared for extensions and adaptations to the specific system. Integration of controllers or simulations from MATLAB®/Simulink® or C/C++ can be easily realized, due to the flexible TwinCAT 3 architecture.

All system information is provided in databases, and provisions have been made to consolidate all data in a central database. Each individual database, as well as the central database, enables extensive monitoring and evaluation of the system states and operating modes of wind turbines.

Further information:
www.beckhoff.com/TwinCAT-Wind


Big Data
Optimised preparation, availability, and processing of all relevant data in real-time for system operators and manufacturers
– Data collection and Data Warehousing
– Data analysis and Data Mining
– Power and Condition Monitoring
Operator
Cloud
Big Data
Manufacturer
Engineering
Comprehensive and integrated engineering throughout the lifecycle of the installation
– IEC 61131-3 programming languages, C/C++, MATLAB®/Simulink®
– Object orientation, modularisation
– Data exchange between engineering tools
– Automated engineering
Communications
Secure vertical and horizontal communications
– Support for all the relevant bus systems (EtherCAT, Ethernet, PROFIBUS, etc.)
– Comprehensive messaging/connectivity (ADS, OPC UA, live diagnostics, etc.)