AN INFLUXDATA CASE STUDY

Optimization of Wind Turbine Operations Using Factry OPC-UA Collectors

Frederik Van Leekwyck
Co-Founder & Business Development,
Factry.io

MAY 2019

Company in brief

VLEEMO NV is a collaboration between the Flemish energy companies Aspiravi NV (50%) and Polders Investeringsfonds NV (50%). These companies specialize in renewable energy. VLEEMO built and operates 33 wind turbines on the Right Bank of the Port of Antwerp, with more turbines under construction.

VLEEMO has a wealth of experience and knowhow in the field of financing, development, construction and operation of wind turbine parks in the port area. These projects are particularly complex due to the proximity of seveso companies, CTR zones, pipelines, freight rail infrastructure and specific nature reserves. Thanks to the various studies and projects that VLEEMO carried out, the company is a pioneer and a reference in the field of wind energy in the port area.

Case study overview

Optimizing electricity output from wind turbines requires constant adaptation to environmental conditions, while staying within the legislative boundaries. Specifically, wind turbine operators such as VLEEMO have to respect shadow flicker and ice formation limitations but still keep their turbines running as much as possible within these boundaries.

In the past, turbines were primarily monitored using remote-controlled SCADA systems. Now, all sensor information from the VLEEMO turbines is gathered in InfluxDB using multiple Factry OPC-UA collectors. Factry then augments this information with external data such as weather forecasts.

The turbine statuses are interpreted on the fly, based on the raw data in InfluxDB. Relevant information is subsequently displayed centrally in Grafana, so VLEEMO can take immediate action when necessary. In the end, this allows them to become more productive and save time in their day-to-day operations.

VLEEMO Turbines, In The Port Of Antwerp, Fitted With Sensors Whose Information Is Gathered In InfluxDB Using Factry OPC-UA Collectors

“

Before Factry, VLEEMO had to log in via remote desktop into each individual SCADA system per wind farm to have a look at how the turbines were doing.”

Frederik Van Leeckwyck, Co-Founder & BD Manager

The business problem

VLEEMO is a wind turbine operator that currently operates 33 wind turbines across seven wind farms, generating power for about 65,000 households. VLEEMO is responsible for ensuring the turbines are running in accordance with operational regulations in effect in Belgium, and so far, they have a maximum power of about 90 megawatts. What's peculiar is that the turbines they operate are not located in open fields but in the very busy Port of Antwerp (Europe's second biggest port), scattered throughout between industrial and office buildings. Thereby, they need to operate their turbines taking into account this port's environment.

VLEEMO’s business challenge was to generate as much electricity as possible given variable weather conditions while staying within regulatory boundaries. As with other nature-dependent renewable energy sources, weather conditions require switching equipment on and off which, when done manually, is slow and not optimal for production. Apart from VLEEMO’s energy production being dependent on variable wind conditions not within one's control, it was also restricted by the limitations set by the two regulatory boundaries relevant here. Clearly, these regulations slow down energy production, as explained below.

Nuisance - shadow flicker limitations

The first regulatory boundary involves controlling Shadow Flicker: the nuisance that wind turbines can cause, in the presence of direct sunlight, for homes or office buildings in their vicinity. Shadow flicker occurs on sunny days when wind turbine blades cast a shadow across the landscape, causing an on-and-off flicker effect as the turbine blade rotates. The location of the turbine shadow varies by time of day and season and can induce adverse human health effects including annoyance and/or stress. Shadow flicker is a concern in Northern Europe where high latitude and low sun angle exacerbate the effect.

To prevent excessive nuisance, the regulatory boundary in Belgium is to allow a maximum of 30 minutes of flickering per day, a maximum of 8 hours per year for private residences and 30 hours per year for office buildings.

Predicting shadow flicker by adding related data sources such as weather forecasts and additional sensor data would enable VLEEMO to maximize energy production within the regulatory limitation as well as achieve time savings.

Safety - ice formation hazard

The second regulatory boundary involves public safety:

- Ice formation - When it snows in Belgium and temperatures drop below freezing, this results in potential ice formation on turbine blades. As soon as ice forms there, the turbine needs to be stopped to avoid ice chunks being hurled around and damaging buildings or tenants.
- Blade heating - Within those turbines are blade heating systems that are used to melt the ice from the blades so that when the ice is gone, the turbine can be restarted. VLEEMO can only restart the turbine after visual inspection by a mobile dispatching team. When ice formation happens at night, the turbine can only be restarted in the morning when visual inspection is possible, since such inspection requires actual visits to each turbine.

The technical problem

To solve their business challenges, VLEEMO needed real-time insight into the status of all their turbines. Yet unlike familiar portable consumer devices, industrial IoT (IIoT) equipment is much larger and of a totally different scale. It is designed to have a useful life of decades rather than years, with investments reaching several million euros. The scale at which data is generated can be confined (as with biomass or manufacturing plants) or distributed (as with VLEEMO).

Shots from a VLEEMO Exhibition: Turbines Under Construction

Photos by Frederik Van Leeckwyck. Reproduced with permission.

Given such equipment, VLEEMO's need to solve its data challenges led it to choose Factry Historian, a process historian that uses InfluxDB to provide real-time and historical insights for its clients.

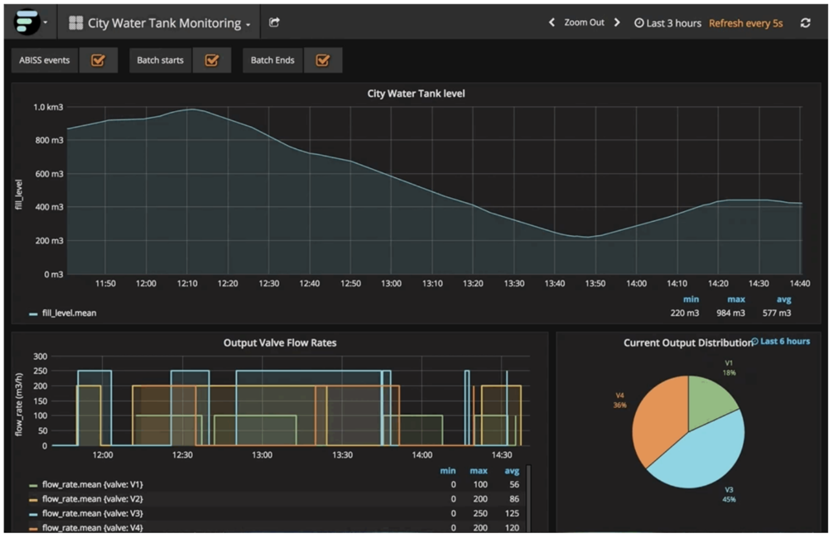
Each of VLEEMO's six wind farms in the Port of Antwerp has a Supervisory Control And Data Acquisition (SCADA) system. Before Factry, VLEEMO were using remote desktop connection to log into each individual SCADA system, view the turbine status, and make decisions based on that:

- There was a historian system present on some of these systems. But it was either not used or underutilized, so they had no or limited data collection.
- This resulted in limited opportunities for data-driven optimization. Optimization was done based on experience, which they most definitely have, yet the availability of a variety of data would enable faster time to action and improve operational efficiency.

The solution

> *“The database [InfluxDB] is extremely easy to set up, requires no external dependencies, has a SQL like query syntax and is fully open source.”*

Factry used InfluxDB and Grafana among its components and added to that its own software and the service that comes with it.

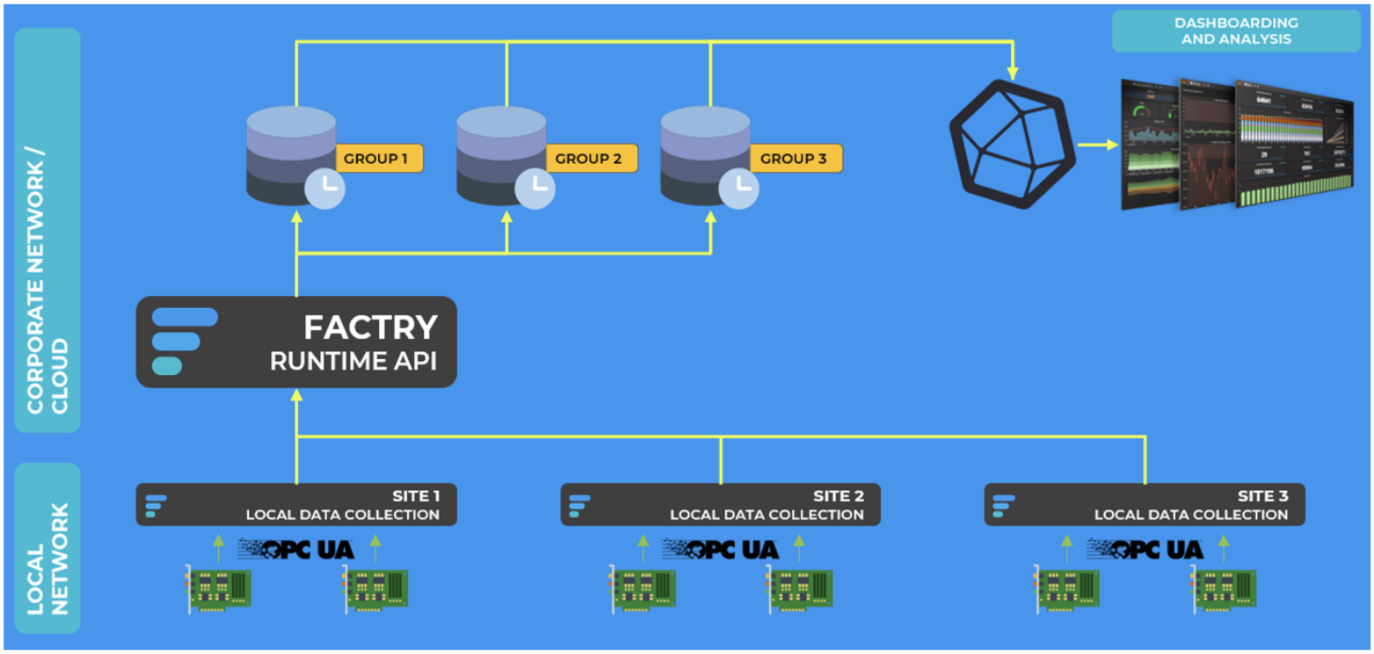


Prior to adopting Factry Historian, VLEEMO was only collecting one set of data with their SCADA systems. Over time with Factry, they were able to layer additional data sources on top of what they were collecting, making it richer and so much better for forecasting and boosting production efficiency.

Besides the basic sensor data, VLEEMO started collecting temperature sensor data, temperature forecasting, neighbor scheduling, and personnel scheduling to guide their turbine operation scheduling decisions within variable weather conditions and regulatory restrictions. Powered by InfluxDB — which allows data ingestion and storage at massive volumes and retention at flexible durations for short-term and long-term historical analysis — Factry Historian's modular architecture allows VLEEMO to add even more data sources and develop more forecasting capability.

The richness of the data made available through Factry’s solution allowed VLEEMO to send instructions back to the turbines (via OPC-UA Server as discussed below), thereby taking optimization full circle from and back to the turbines in a two-way information stream that effectively maximizes productivity and minimizes lost revenue.

Below is an overview of the solution Factry developed for VLEEMO.



Factry Historian: Modular Architecture

- The lower part of the chart shows a local network while the top part's components are situated in the corporate or business network (on-premise installations or the cloud).
- The chart, for demo purposes, shows three sites (whereas VLEEMO has six sites). Via the OPC-UA protocol, Factry collects data from the SCADA systems.
- All data is sent on the Factry runtime API which performs data validation, checks whether the data is coming in, and gives the user a UI to control what data should be collected from the industrial control systems.
- Then the data is put into different groups, represented by different databases in InfluxDB.
- Finally, this data can be used for dashboarding and analysis.

Factry's collectors

Factry's OPC-UA collectors are open source software built by Factry and used as input for InfluxDB. They talk in industrial protocol on one end (OPC-UA) and HTTP on the other. They are built according to a store-and-forward model: If there's no connection to the storage backend, they keep a local buffer of the data that's been gathered, and as soon as the connection is restored, the data is forwarded. They also keep a local copy of the configuration to enable collector startup in rare cases where the connection remains unavailable.

Collector setup

There are two separate networks:

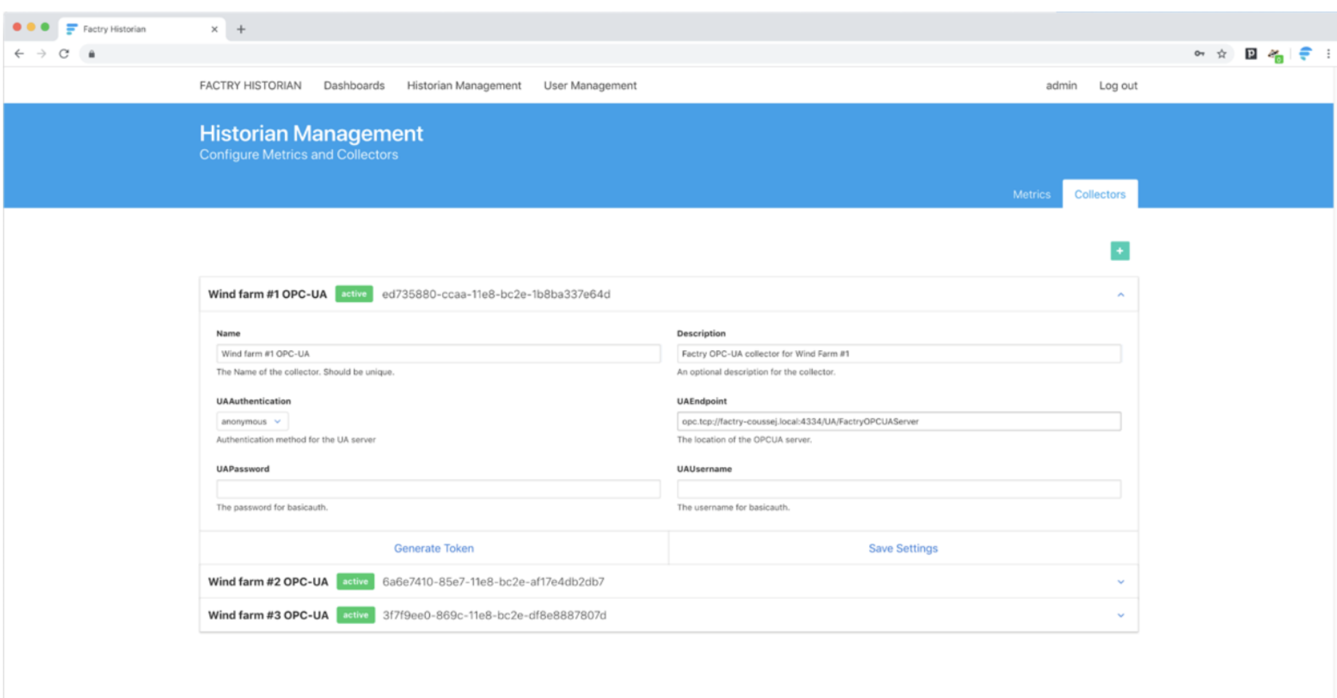
- A **production network** close to the equipment - Each individual wind farm has a small network, where the assigned collectors would run (therefore quite close to data source).
- The **business network** - The data is sent through the business network or through the internet. To make this possible and easy to configure, only outgoing connections from the production network to the business network are allowed so that you don't need to open any ports or any network connections from the internet (the connections are opened from the production network to the business network or client).

Collector registration

Factry started with six collectors. Here is how collectors are added:

- **Create a collector in the operator interface** - Factry created each collector and gave it a name (e.g. Wind Park #1) and a type (e.g. OPC-UA).
- **Generate JWT token for new collector** - They generate one token for this new collector which is used for authorization and for finding out the Historian API endpoint's location.
- **Start collector with JWT token env var** - Then they start the collector as a service, and the only environment variable that needs to be passed is this JWT token. Mostly, these collectors are single-binary deployments, which makes them easy to update. As soon as they have started, they connect to the API and validate themselves.

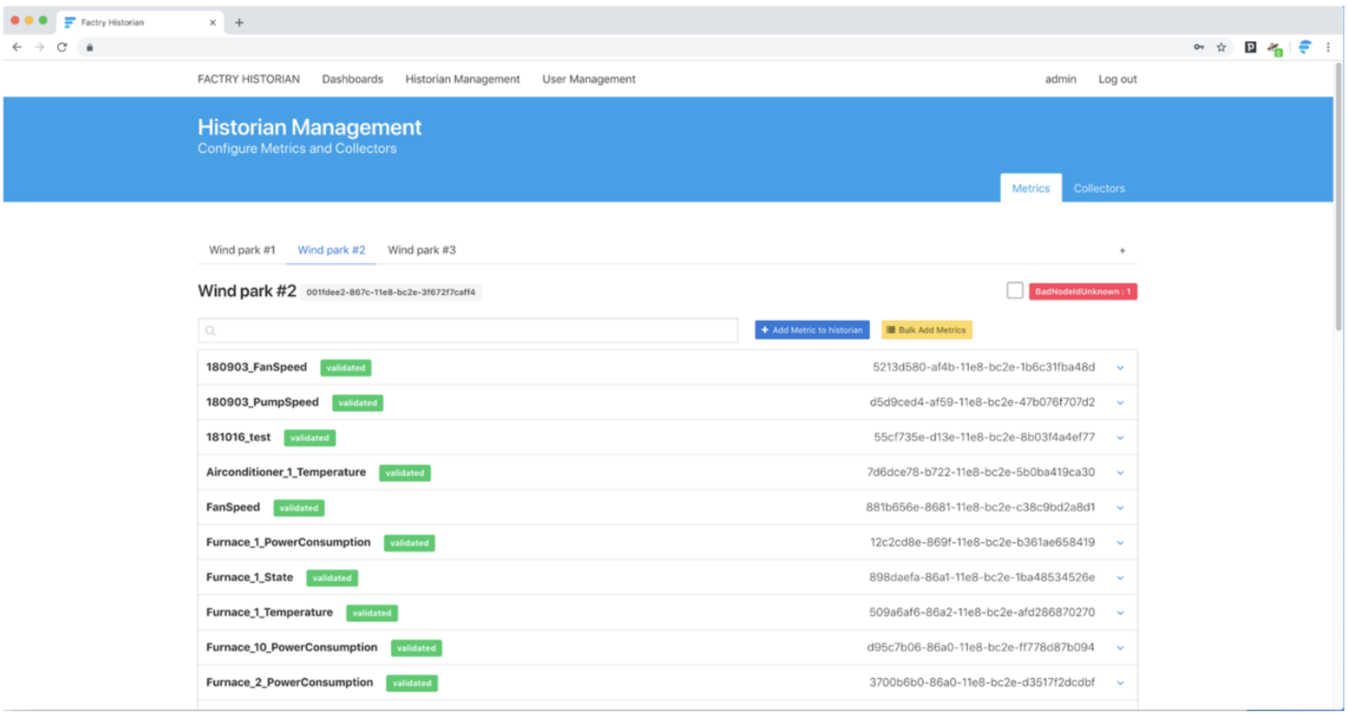
Configure in the operator interface - Then the collectors can be configured. Collectors can be added in the graphical administration panels — with names, endpoints, usernames, and passwords. All three collectors are active, which means that they are valid and have been able to connect to the OPC-UA server.



Collector Registration Screenshot

Configure metrics

Once collectors are added, metrics can be added to them and are called tags in the industry.



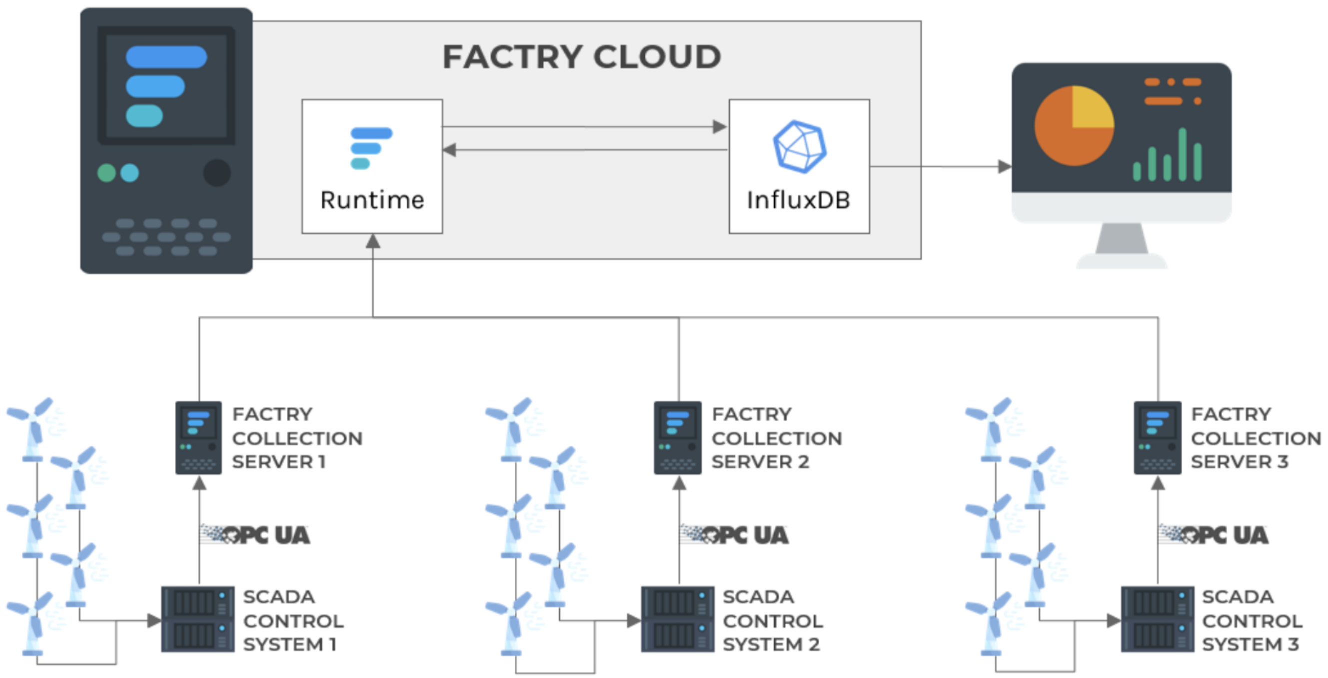
Screenshot Showing Metrics Configuration

In the OPC-UA protocol, those tags can be either polled or monitored. There are different groups, and there you can add one or more data points.

Technical journey

Factry's solution for VLEEMO underwent five iterations in which value was added at every stage.

Iteration #1: Architecture and Basic Setup



Iteration #1 is shown above:

- From VLEEMO's 33 turbines, Factry gathers some 600 data points of raw data, comprising basic data as well as additional data from Meteo Systems and additional sensors. (They also add more data that is calculated in real-time, as discussed further below.)
- The basic data that can be collected from turbines (from the SCADA Control Systems) is:
    - Energy produced
    - Wind speed
    - Wind direction
    - Position of nacelle (wind turbine's cover housing)
    - Turbine status

- Each wind farm is controlled through its SCADA system. With the OPC-UA protocol, Factry gathers the data in its collectors. The collectors send that into the cloud via a runtime which is finally sent to InfluxDB for long-term storage, where it can then be used for visualization. (The arrow pointing backwards from InfluxDB is explained later below).

Benefits gained from central data collection:

- **Grafana dashboards provide central overview** - The basic data shown in Grafana dashboards gives VLEEMO a central overview of their complete turbine operations. The benefits they received just from having data centrally available is that they have now this central overview that everyone at VLEEMO can use to have a quick overview of how everything was running.
- **Archiving of historical status** - On the top left of the dashboard below, the specific wind park and turbine can be selected to view the right data. This represents basic data collection offering historical insights into what is happening over the course of weeks, months, and years.

Dispatch

Taking that visibility to a higher level, the status dashboard below now gives them a central overview of how all their different turbines are running. A green box means that everything is ok. An orange box means the turbine is in maintenance.

Having data centrally available has enabled everyone at VLEEMO to have a quick overview of how everything was running. Because this data is now archived, VLEEMO gets insights into the status of its assets over the long term so that they can increase reporting functionality.

Turbine Status

- **Go-to tool for new hires** - Grafana has become the go-to tool for new hires so they can immediately see how the turbines are running.
- **Selectively share data with third parties** - Thanks to an expressive and open API, VLEEMO is now able to selectively share data with third parties, such as research institutes.

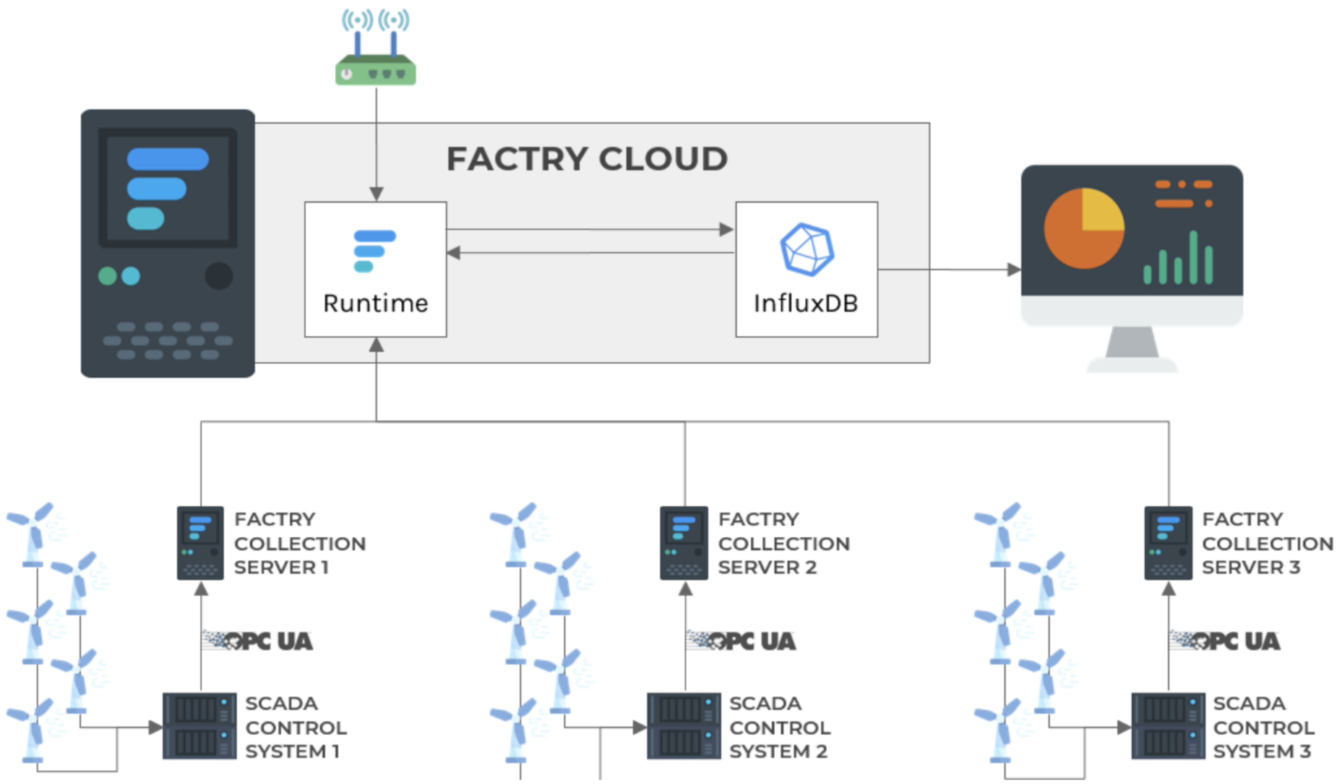
In addition to simple data selection from the industrial control system, Factry added a few layers of additional value in order to really improve VLEEMO's operations.

Additional sensor data

In Iteration #2, additional sensor data was added. For a few turbines, VLEEMO has purchased external sensors that are stuck to the turbine blades in various positions and that give fine-grained information on blade temperature and potential ice formation. These are standalone systems that also transmit their data to those sensors' supplier.

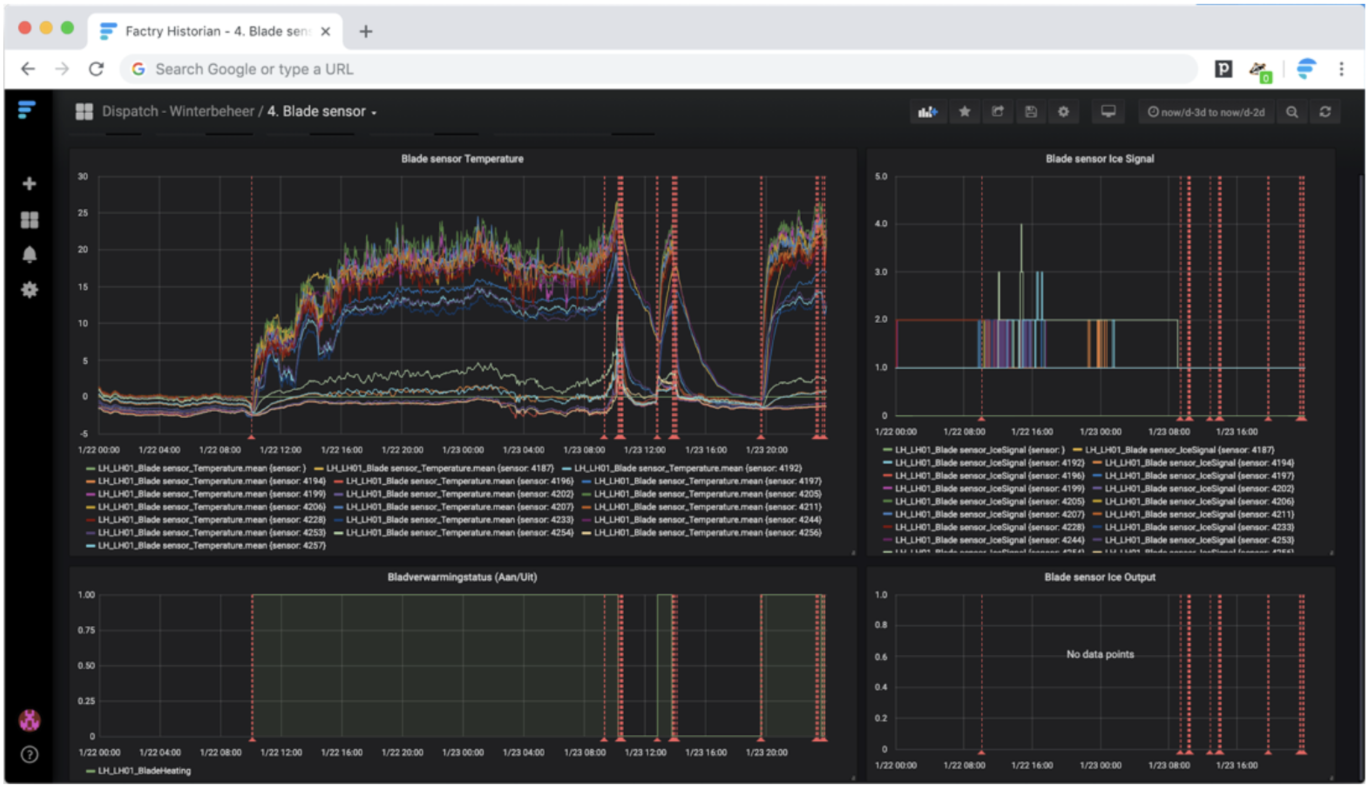
Factry contacted that supplier and had them open up that data, via Factry's REST API, so Factry could query that data and add it to the database in a structure that is in line with the basic data already being collected. These additional sensors (represented in the green sensor icon in the below chart) constitute a new data source being added and then stored in InfluxDB.

Iteration #2: Add Sensor Data



The dashboard, even for one turbine, now shows many different sensors, which gives VLEEMO much more fine-grained information into turbine blade temperature and ice detection.

Factry Historian Dashboard - Blade Sensor



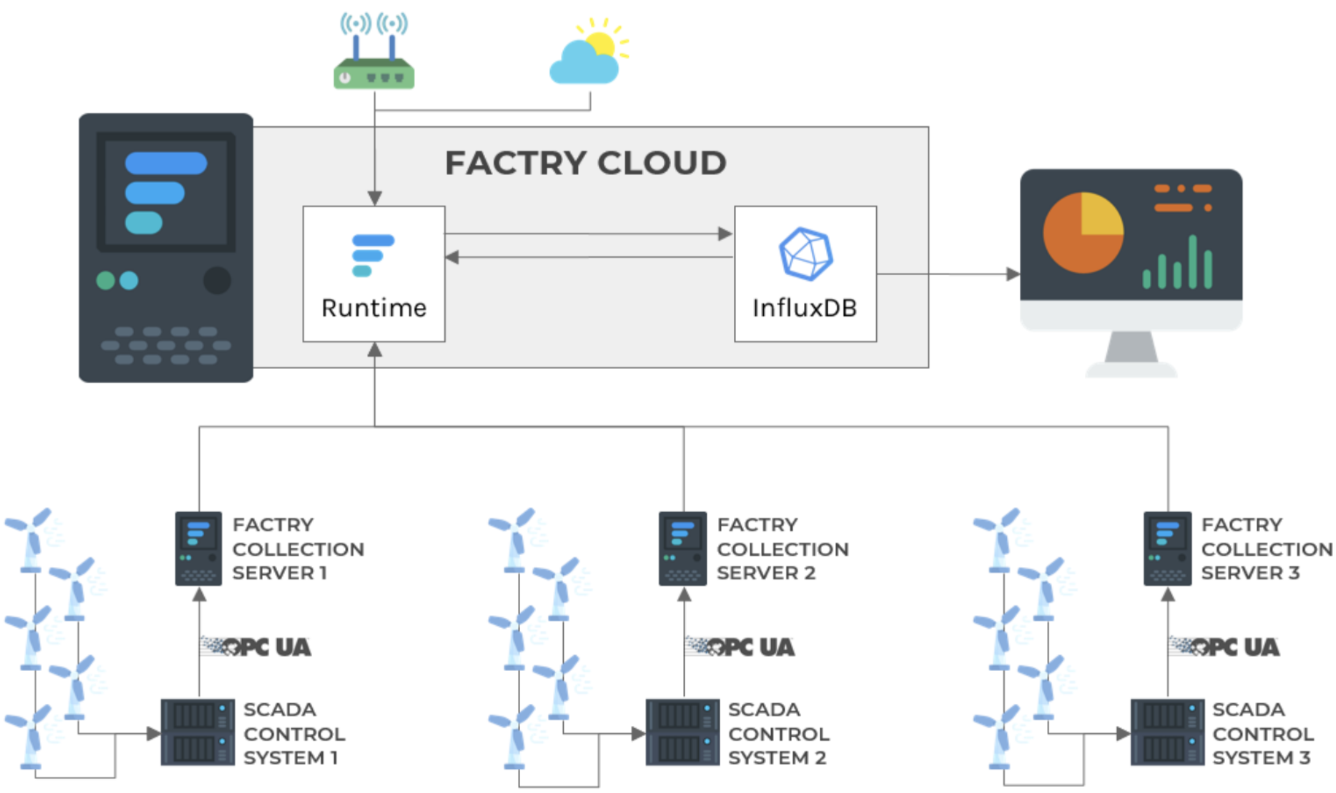
As soon as the blade heating turns on, the temperatures start rising.

Benefit:
VLEEMO, through gathering more fine-grained data, now have more detailed information to either operate or stop the turbine from running during winter conditions. They're also able to validate whether that additional sensor system makes sense in their specific case. If so, they can roll it out to more turbines.

Weather forecasts

Iteration #3 added weather forecast data. Rather than subscribing to a weather forecast service, they decided to create a REST API and add it to the database. They edited its initial data source, query that regularly, and write that to InfluxDB with future timestamps. The chart below shows this newly added weather data (as represented in the weather symbol), which is also stored in the same data structure.

Iteration #3: Add Weather Forecast Data

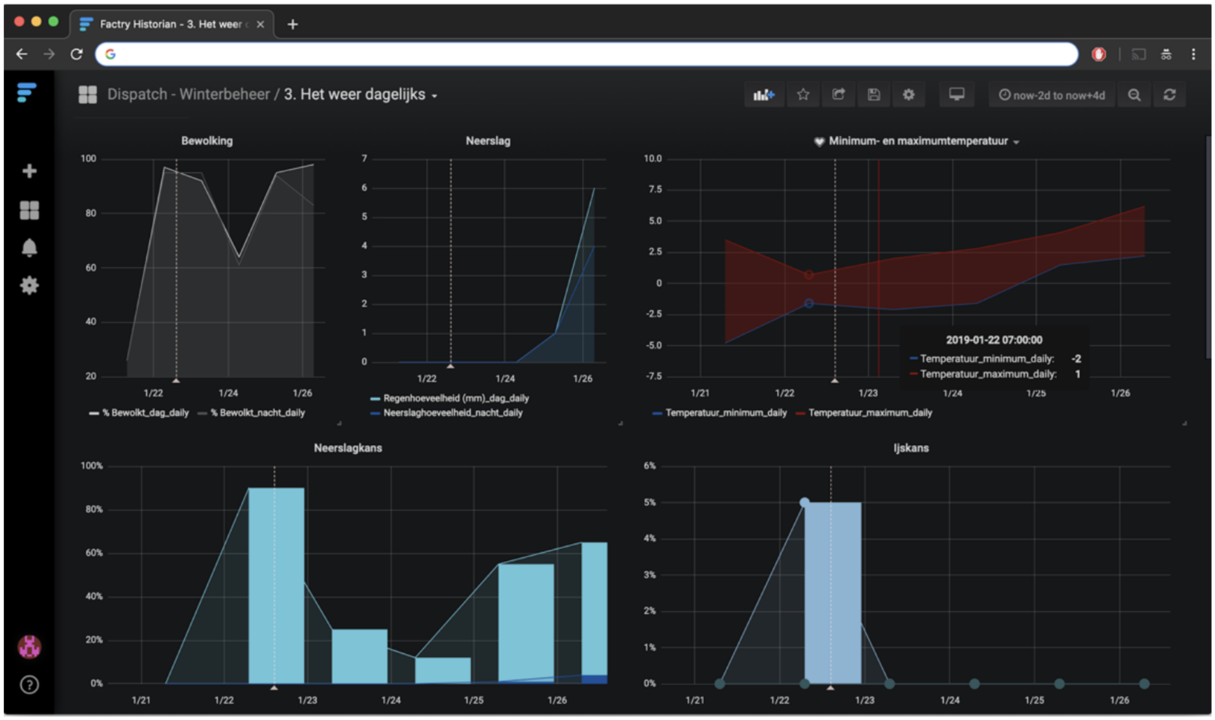


Benefit:
Iteration #3 improved planning of the dispatching team. As shown in the dashboard below, control or centralization of all data in InfluxDB enables VLEEMO to anticipate future action with the same system.

Whether or not it's going to be freezing on that day can determine whether to expect the turbines to be operational.

When ice forms on the blades, you can restart the turbine only after visual inspection of the turbines. This means that, from the moment you are permitted to re-start them again, you should be able to do so as quickly as possible. So you need a dispatching team that is able to release each individual turbine as quickly as possible.

The sooner you know that you are going to have winter conditions, the better you can plan a dispatching team that's available on the ground to give the okay to restart the turbine when temperatures are again above zero. Alerts are set up for minimum temperatures in the next few days so that VLEEMO are always aware, and thus better plan, the dispatching team.

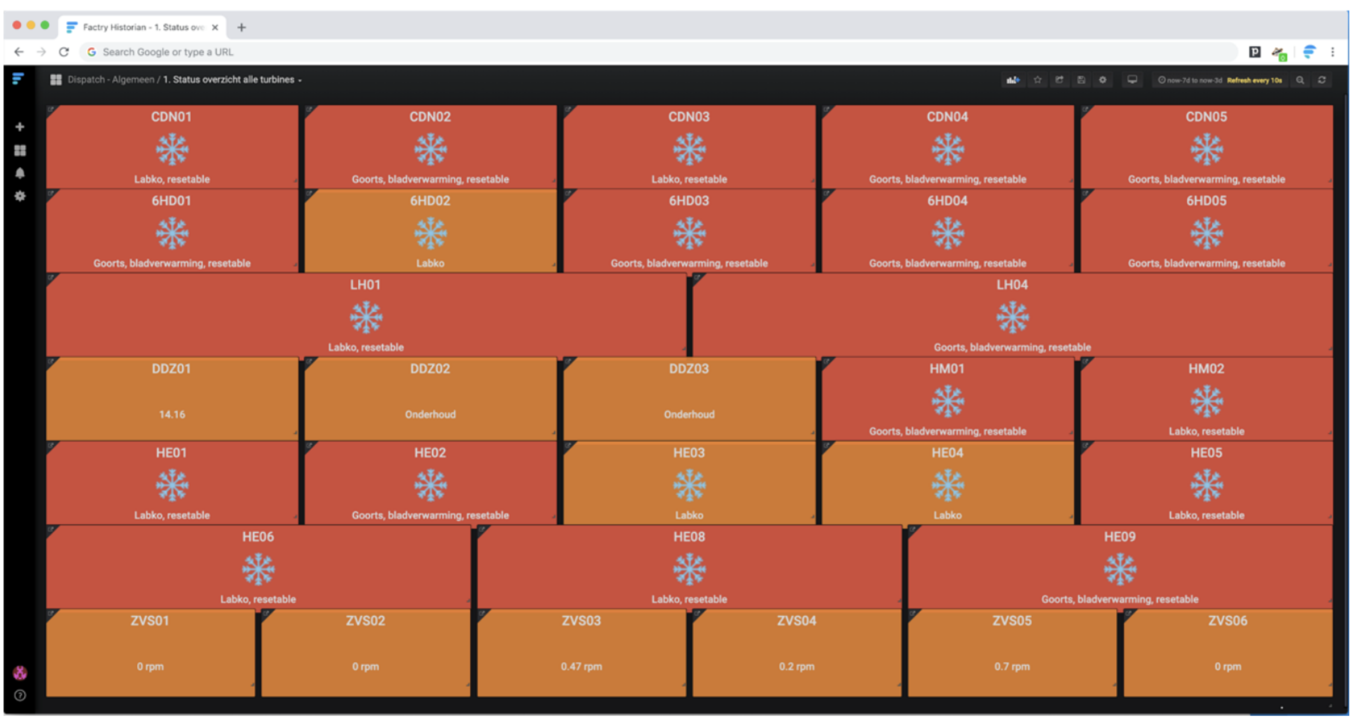


Adding status interpretation

Adding further status interpretation has been a very big — and ongoing — addition to the data being collected. Among the basic data collected is turbine status. The turbine manufacturer has a list of status codes that describe the turbine's status functionality. In some cases, this information is a bit too high-level. Based on all the data now available, Factry can interpret this turbine status in more detail than the turbine manufacturer does (see the new icon near InfluxDB below).

Iteration #4: Manual Status Interpretation

Factry has written and developed, together with VLEEMO, a custom algorithm to outperform turbine status classification provided by the turbine manufacturer. To do so, they have written a custom Golang daemon to read and write to InfluxDB. This daemon is constantly being improved over time. It's built with replay functionality so that every time they have new insights into how to improve this classification, they can replay this daemon. The algorithm runs and gets data from InfluxDB and writes it back. The result is a very fine-grained interpretation of turbine status:



Adding status interpretation

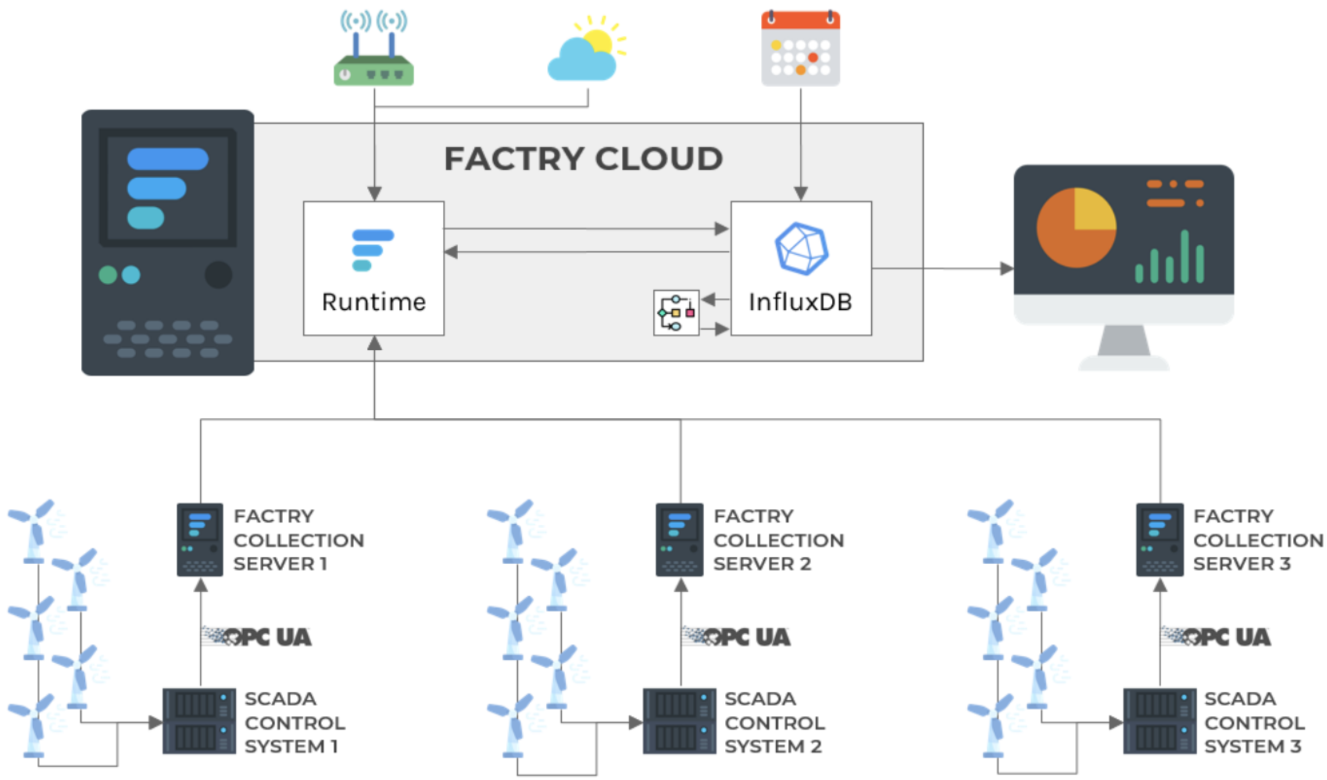
Benefits:

- **Improved Grafana overview** - In the above chart, each block represents one turbine, but now more detailed status codes are available than what the turbine manufacturer is able to supply. This detailed status overview provides insights into where optimization potential exists as well as providing the ability to act very quickly. Fine-grained information allows VLEEMO not only to take more appropriate action but also to save time for dispatching rather than getting a turbine status code from the manufacturer. Now they have a human-readable turbine status, which allows faster decision-making.
- **More detailed reporting on lost revenue** - Because of their custom daemon's replay functionality, they can increase their reporting on lost revenue. Lost revenue is the energy generation missed by a turbine that is not running that could have been running because the wind was blowing. Detailed insights into why a turbine is either running or not running can prevent lost revenue.

Shadow flicker calendar

VLEEMO knows where all the turbines are located in the Port of Antwerp and have dedicated Geographical Information Systems (GIS) software to determine potential shadow flicker on windows. They know the position of the sun and the possibility of shadow flicker, but they don't really know whether or not it will happen because it depends on direct sunlight. So VLEEMO exported this data and were able to upload that data themselves into InfluxDB and thus add a new data source (as shown by the calendar icon below).

Iteration #5: Addition of Shadow Flicker Calendar



Benefits:
The below Grafana dashboard shows a status overview of when a turbine could be stopped:

- At the top is a stacked graph in which each block represents a window — a receptor that could be susceptible to shadow flicker (but only when there is direct sunlight).
- The bottom row, between zero and one, shows whether or not at that point in time there was direct sunlight.

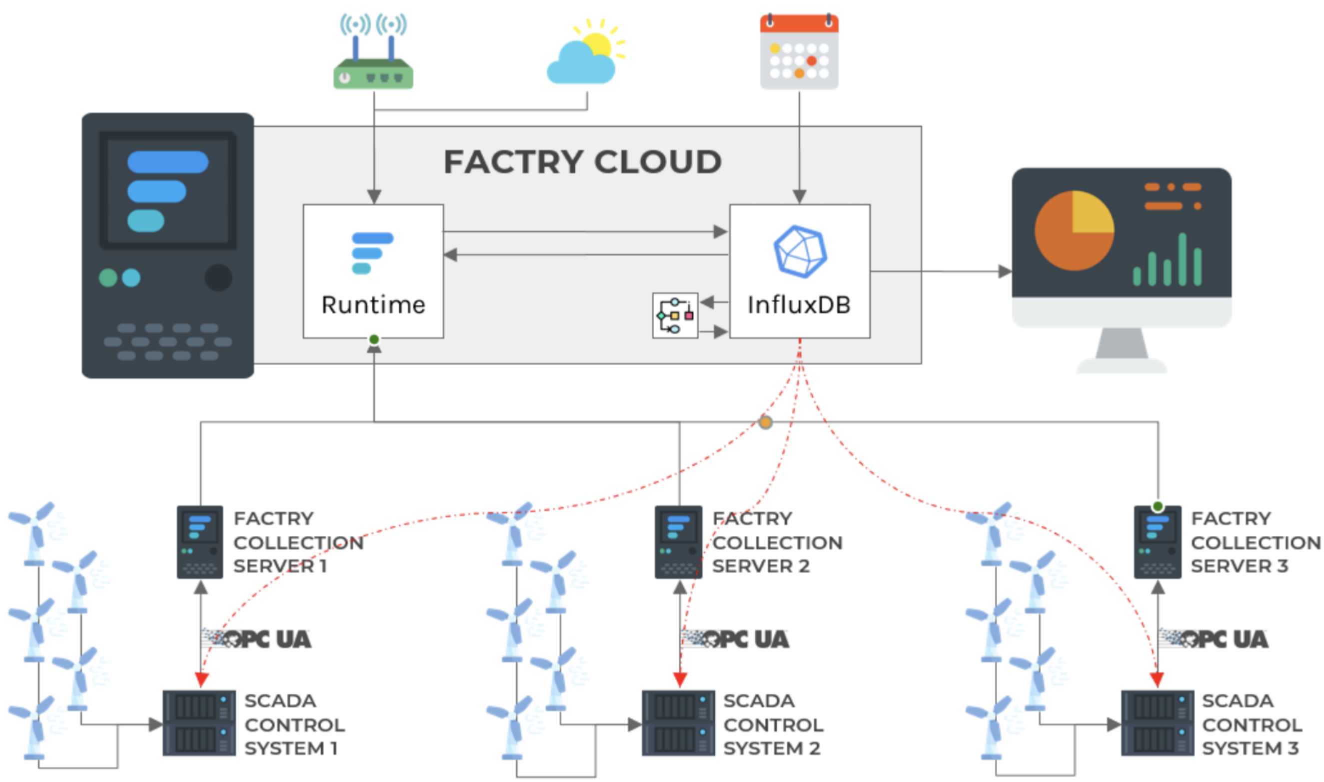
The combination of (1) and (2) indicates shadow flicker potential. In this case, there was direct sunlight, so you have shadow flicker. Only then, the turbine should be stopped, taking into account the limitation of 30 minutes per day or eight hours per year.

Now centrally available, in the above Grafana dashboard, is all the data regarding when a turbine could be stopped to adhere to the shadow flicker regulatory boundary. Optimization can be pursued even further if shadow flicker is tolerated by “window owner” who might state that the building will be empty and therefore the shadow flicker calendar can be overwritten. Yet to realize this additional optimization potential, you need to put more pieces together.

Technical architecture

> *If you start putting these pieces together, at least the last one or any other additional decision criteria, you can see that this data is only available or would only be available in InfluxDB.”*

Final Architecture and Basic Setup: Fitting It All Together



The above chart shows all the pieces of data being fit together. Now centrally available in InfluxDB are: shadow flicker potential; real-time and forecasted information about direct sunlight; and potential acceptance by ‘window owner’ (for example via SMS, informing VLEEMO that the office will be vacant on a given day, so they don't have to respect the limitation on that day).

This central information is used to control turbine behavior based on more information than normally available in an industrial control system. So, is there a shadow flicker potential? Is there direct sunlight? Has the window owner not give permission to overwrite? Then you need to stop the turbine. This near real-time feedback can be represented as follows:

```
Shadow flicker potential (1/0)
x
direct sunlight (1/0)
x
!window owner permission (1/0)
= stop turbine
=> OPC-UA server exposing Influx last() calculated value
```

Factry thought it would be ideal to have an OPC-UA server expose a calculated value based on these three criteria and expose that back to the industrial control system. They have therefore written an OPC-UA server that is able to take one or more measurements available in the database and expose them over OPC-UA; then the industrial control systems can subscribe to the data point and potentially overwrite another decision they would have normally made purely in the industrial control system. This offers the possibility for data flow back to the turbines. Factry has released, in open source, the core functionality of its OPC-UA server.

Monitoring Factry Historian

Factry has a lot of data flowing in from different sources and uses InfluxDB and its ecosystem of tools to monitor the whole Factry Historian cloud setup. If no new data comes in for 30 seconds, they get alerted immediately. This allows Factry to offer to the market Historian as a Service.

Results

> *By gathering all this data centrally and in InfluxDB from all of these different sources and adding custom interpretation on that, VLEEMO is able to perform operational improvements."*

Factry's solution for VLEEMO achieved quantifiable and qualitative business value:

1. **Quantifiable** - VLEEMO is now able to quantify the amount and source of lost revenue since they have a very clear insight into when a turbine is not running when it should have been running. They can very easily make the necessary changes in their operations to decrease lost revenue. Also, because of this custom status interpretation that goes much further than what the turbine manufacturer provides, and by having a human-readable interpretation of a turbine's status, VLEEMO constantly save time for ad-hoc root cause analysis by a dispatching team in the port.
2. **Qualitative** - VLEEMO can provide a signal (to other companies in the port that are there constantly monitoring their turbine equipment) that they are able to intervene very quickly. Because the data is centrally available, interpreted with a custom algorithm, it becomes a daily tool for them.

This project has led Factry to reach the following conclusions:

1. **Operational improvement** can be achieved by providing a single source of information on turbine operations: VLEEMO has been able to improve their operations just by having a single source of information on all relevant data for their turbine operations. By collecting data from different sources, interpreting it and presenting it visually, VLEEMO is able to work with data very quickly, and it has become a go-to tool for them.
2. **Interpretation of time series data** results in reporting of optimization potential: By interpreting the gathered time series data, Factry is able to do more detailed reporting and give VLEEMO clear insights into additional optimization potential and thereby enable them to apply data-driven optimization.
3. **Technology stack is fit for purpose**: Through this project, Factry has once again proved that the technology stack that they use, which is completely based on open source technology, is definitely fit for purpose. They've been running this system for VLEEMO for over a year now, and it is running really smoothly.

Factry's solution for VLEEMO goes full circle: from data collection to analysis, adherence to regulations, operational optimization, and data flow back to the devices themselves in order to further control the environment.

Through sensor information gathered in InfluxDB using Factry OPC-UA collectors, VLEEMO is adhering to regulations while maximizing energy production and optimizing its operation in real-time through data-driven analysis and decision-making.

About InfluxData

InfluxData is the creator of InfluxDB, the leading time series platform. We empower developers and organizations, such as Cisco, IBM, Lego, Siemens, and Tesla, to build transformative IoT, analytics and monitoring applications. Our technology is purpose-built to handle the massive volumes of time-stamped data produced by sensors, applications and computer infrastructure. Easy to start and scale, InfluxDB gives developers time to focus on the features and functionalities that give their apps a competitive edge. InfluxData is headquartered in San Francisco, with a workforce distributed throughout the U.S. and across Europe. For more information, visit influxdata.com and follow us @InfluxDB.

Try InfluxDB

Get InfluxDB

Contact us for a personalized demo influxdata.com/get-influxdb/