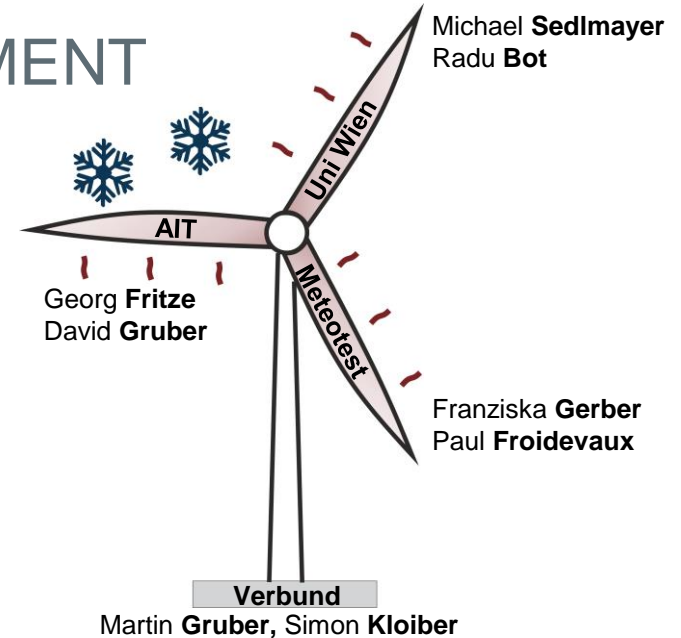


DATA MANAGEMENT AND REAL-TIME ALGORITHM DEPLOYMENT FOR ADVANCING ANTI-ICING ROTOR BLADE HEATING IN WIND TURBINES

Georg Fritze



CONTENT

Introduction SOWINDIC

Challenges

Data processing

- Execution Environment
- Retransmission Strategy
- Data Availability

Deployment

- Dataflow Architecture
- Validation Challenge

Summary & Outlook



[1]



[1]

SOWINDIC

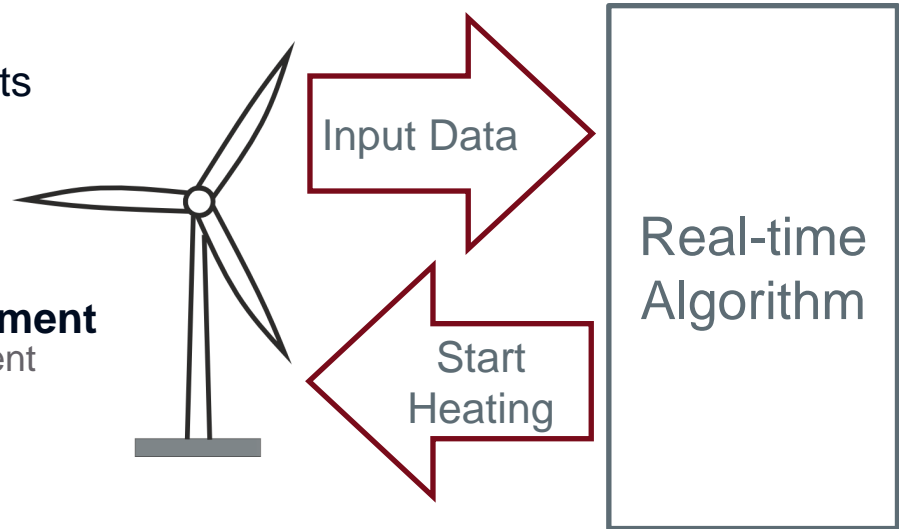
Smart **O**peration of **W**ind Turbines under **I**cing **C**onditions

Use real-time turbine data, weather data, ice detection sensors and weather forecasts to prevent icing using rotor blade heating.

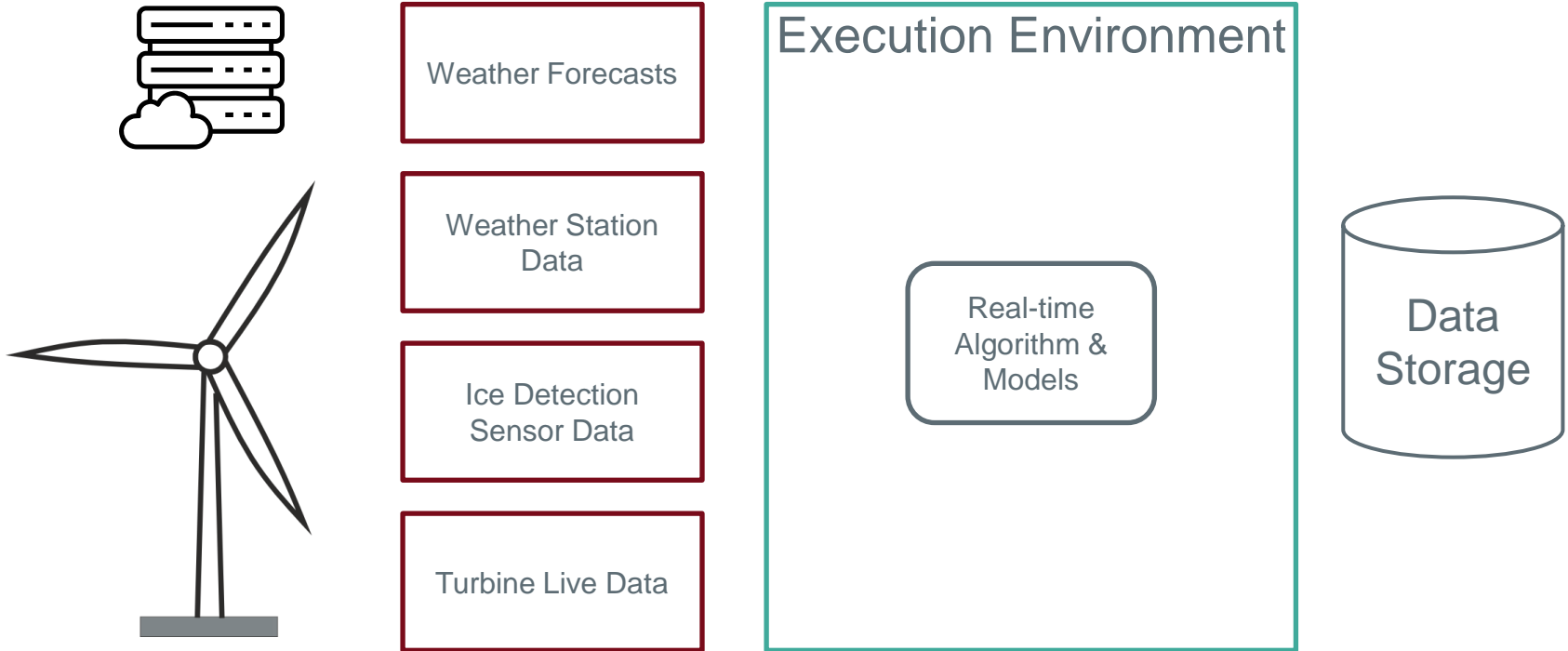
Data management & Algorithm deployment

collection
normalisation
provisioning
storage

runtime environment
validation



SYSTEM ENVIRONMENT



PRACTICAL CHALLENGES

Hardware constraints

- Blade heating supported during standstill & operation?
- How to control rotor blade heating externally?

Regulatory restrictions

- Is blade heating during operation allowed?
- Is automatic blade heating allowed at all?

DATA HANDLING CHALLENGES

Data collection

- Interface data sources with different technologies
- Normalise different data formats: "2023-11-15T10:00:04", "2023-11-15 10:00:04", 1700038804

Data pre-processing

- Data interpolation: Super/sub-sample time series with different update rates
- Provide suitable data sub-sets for different algorithms of multiple developers

Data storage

- Overlapping timestamps in weather forecasts
- Handle connection losses
- Sensor failure or replacement
- Retransmit missed data points

DATA HANDLING CHALLENGES

Data collection

- Interface data sources with different technologies
- Normalise different data formats: "2023-11-15T10:00:04", "2023-11-15 10:00:04", 1700038804

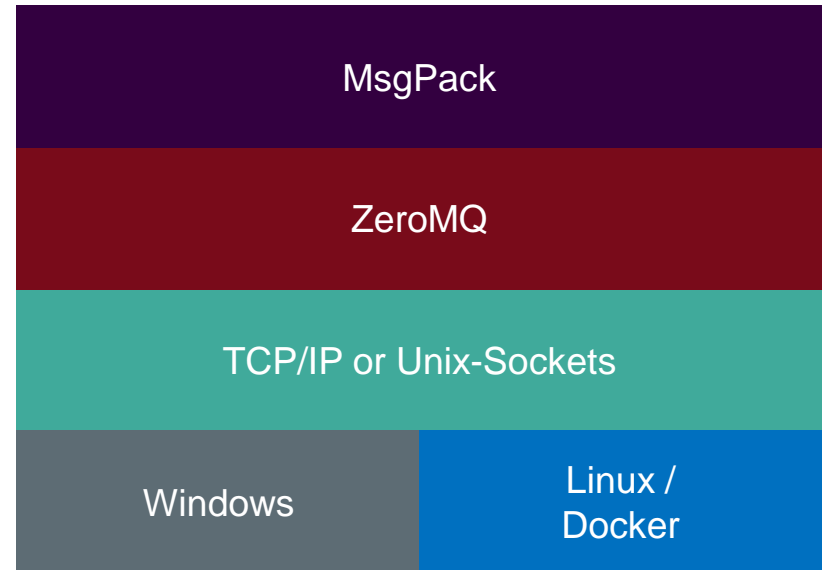
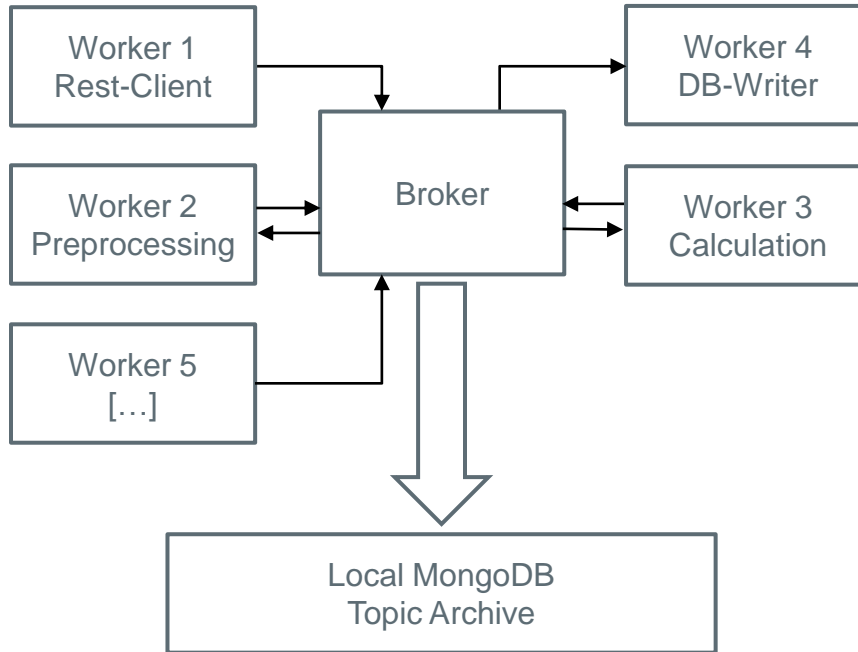
Data pre-processing

- Data interpolation: Super/sub-sample time series with different update rates
- Provide suitable data sub-sets for different algorithms of multiple developers

Data storage

- Overlapping timestamps in weather forecasts
- Handle connection losses
- Sensor failure or replacement
- Retransmit missed data points

COMMUNICATION FRAMEWORK: C++ DATA EXCHANGE SYSTEM



DATA HANDLING CHALLENGES

Data collection

- Interface data sources with different technologies
- Normalise different data formats: "2023-11-15T10:00:04", "2023-11-15 10:00:04", 1700038804

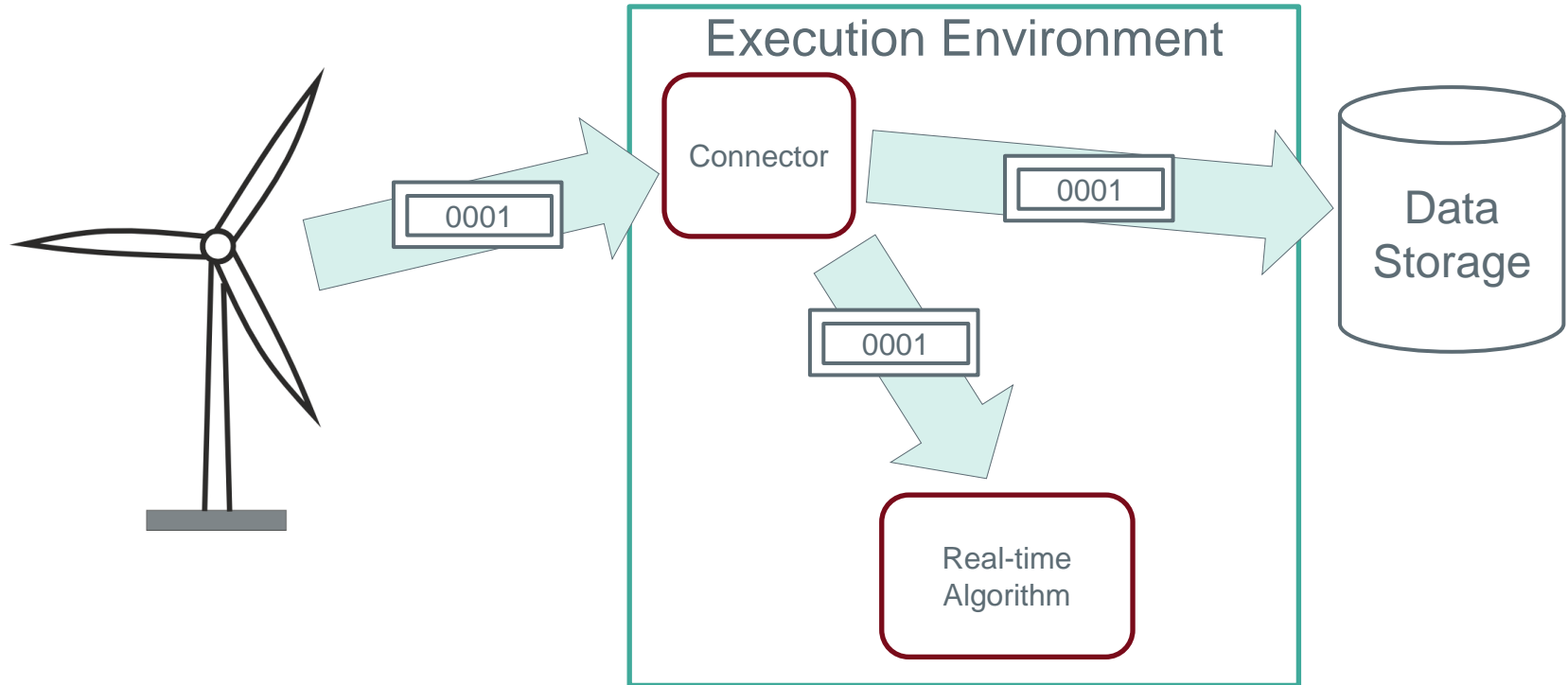
Data pre-processing

- Data interpolation: Super/sub-sample time series with different update rates
- Provide suitable data sub-sets for different algorithms of multiple developers

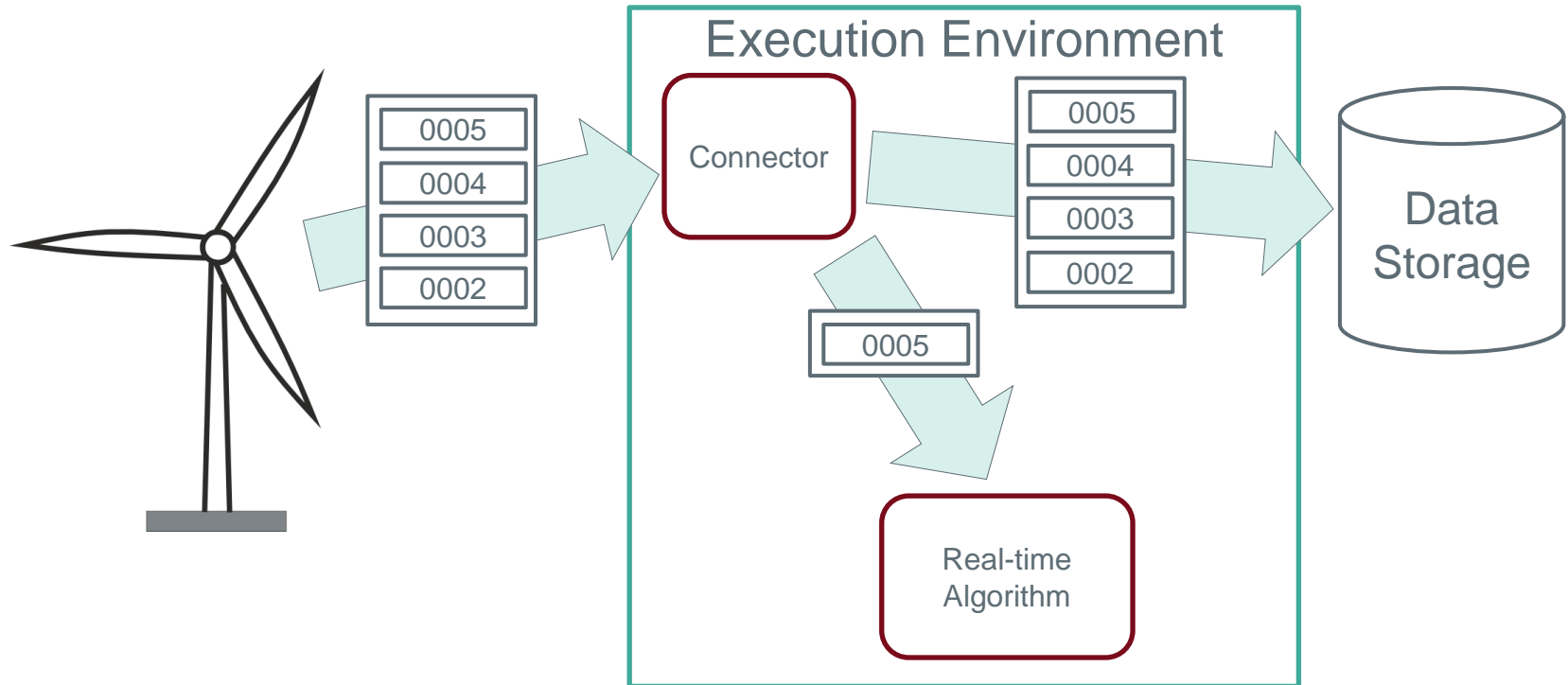
Data storage

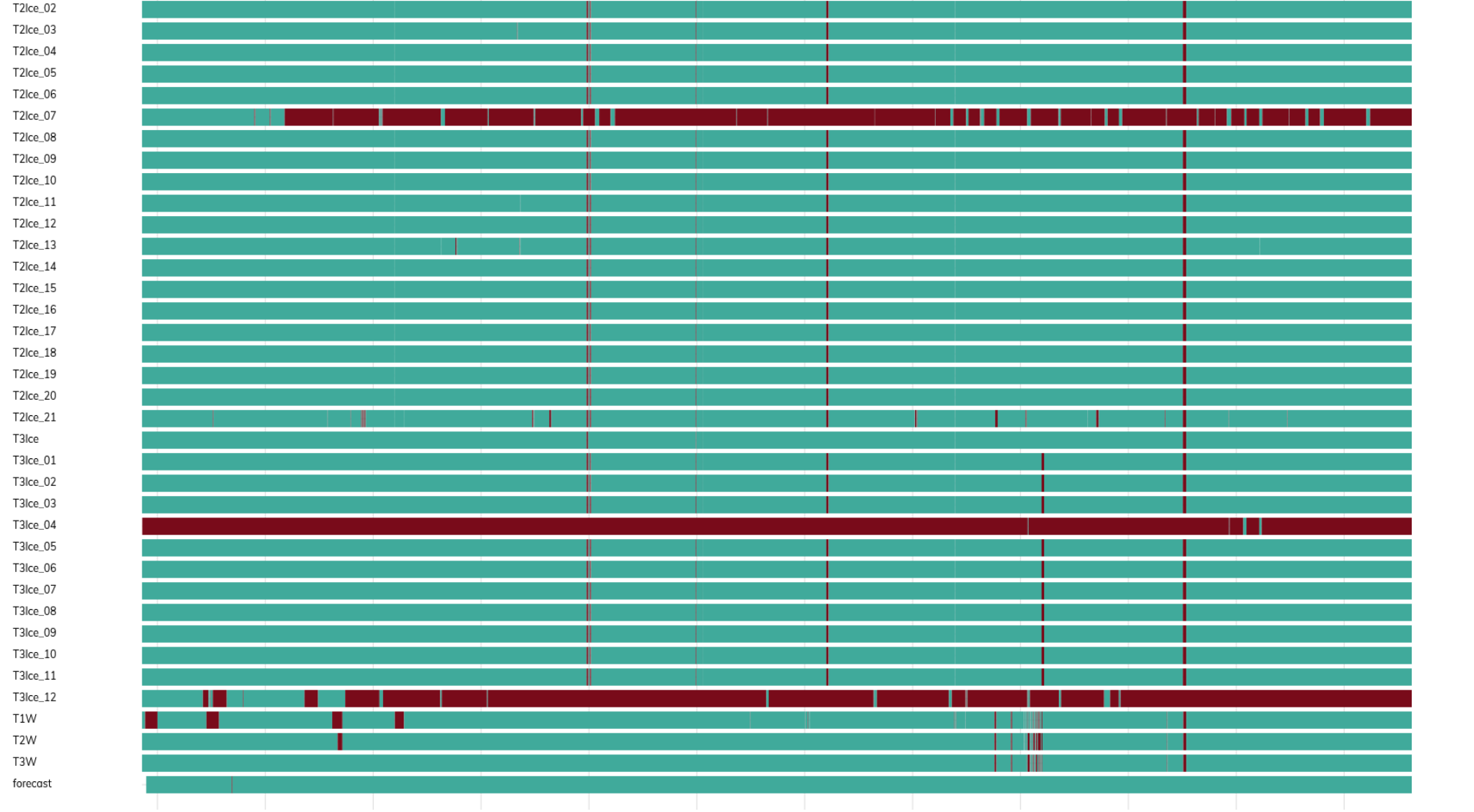
- Overlapping timestamps in weather forecasts
- Handle connection losses
- Sensor failure or replacement
- Retransmit missed data points

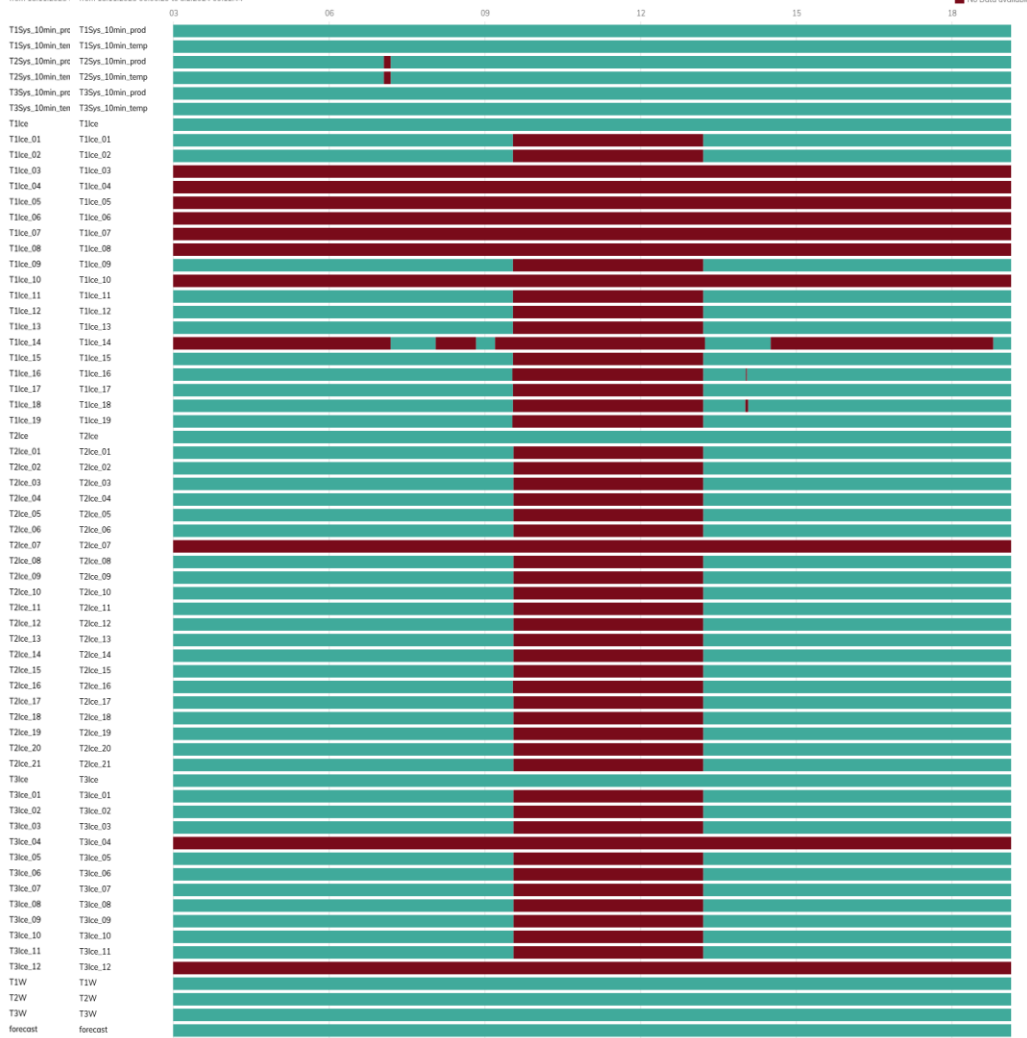
REAL-TIME DATA

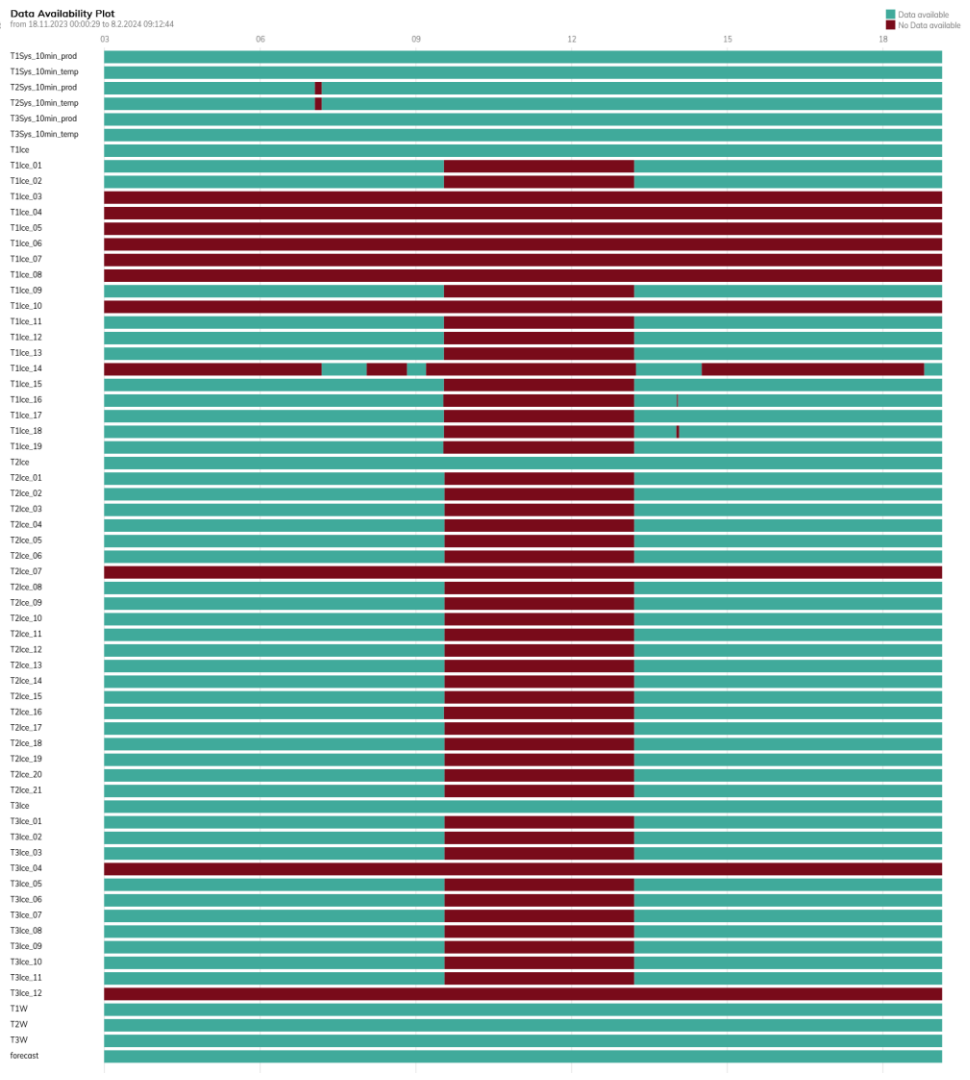
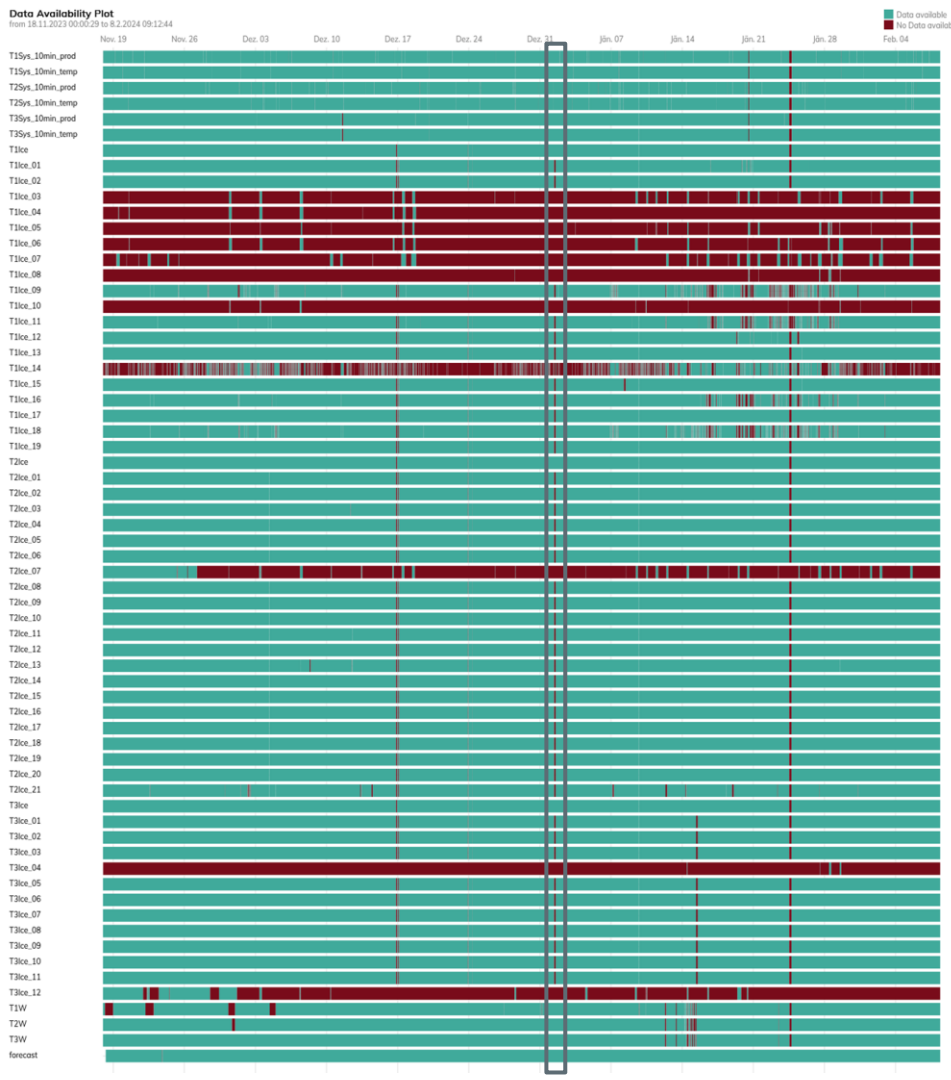


RETRANSMIT DATA





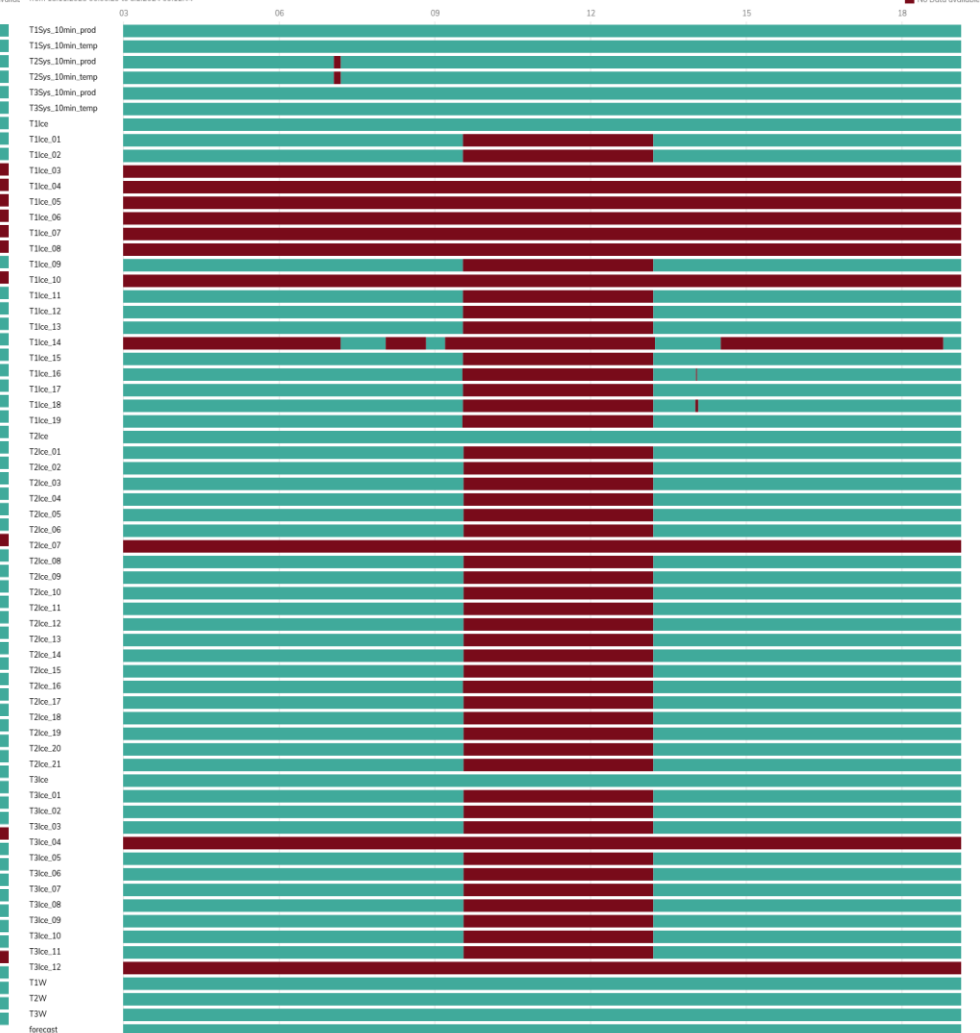




T1z Availability Plot
 from 18.11.2023 00:00:02 to 8.2.2024 09:12:42



T2z Availability Plot
 from 18.11.2023 00:00:29 to 8.2.2024 09:12:44



ON-SITE VS. REMOTE DEPLOYMENT

On-site Execution Environment (Industrial PC)



Handle connection losses for data storage



Preventive heating possible if enough sensors are online



Higher cost: one IPC for every turbine

Remote server



Handle connection losses for each sensor system

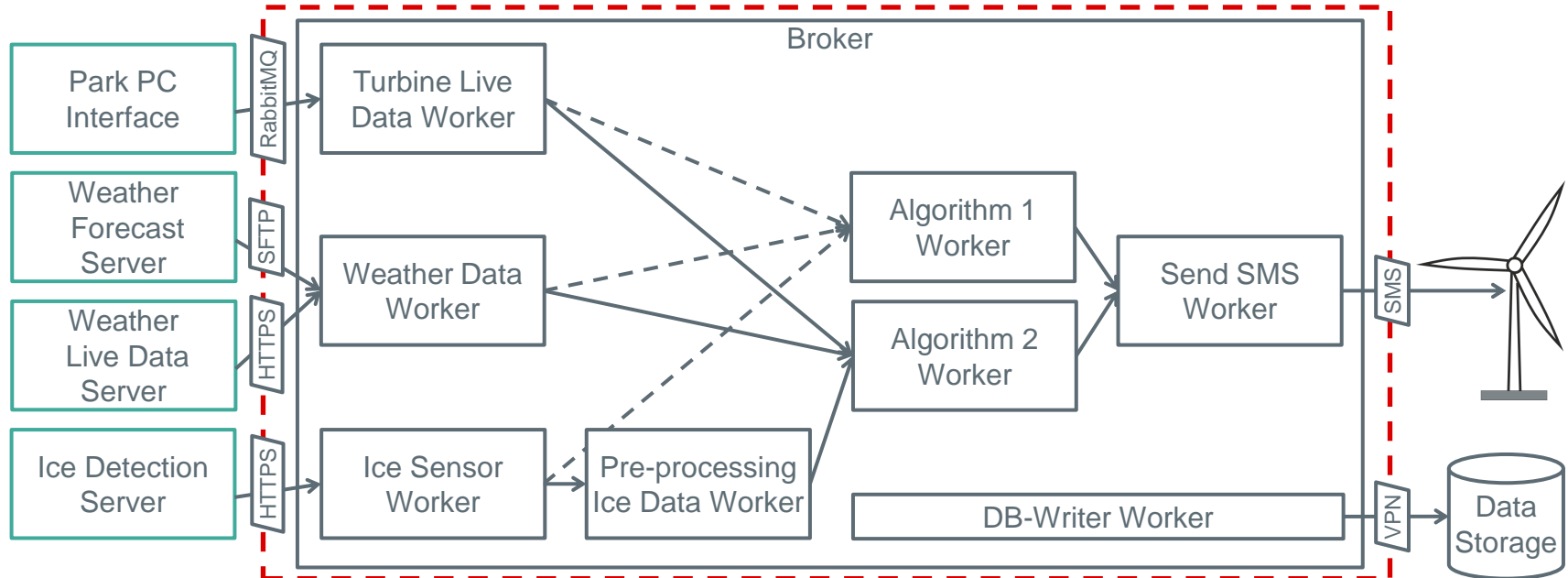


Preventive heating not possible while offline

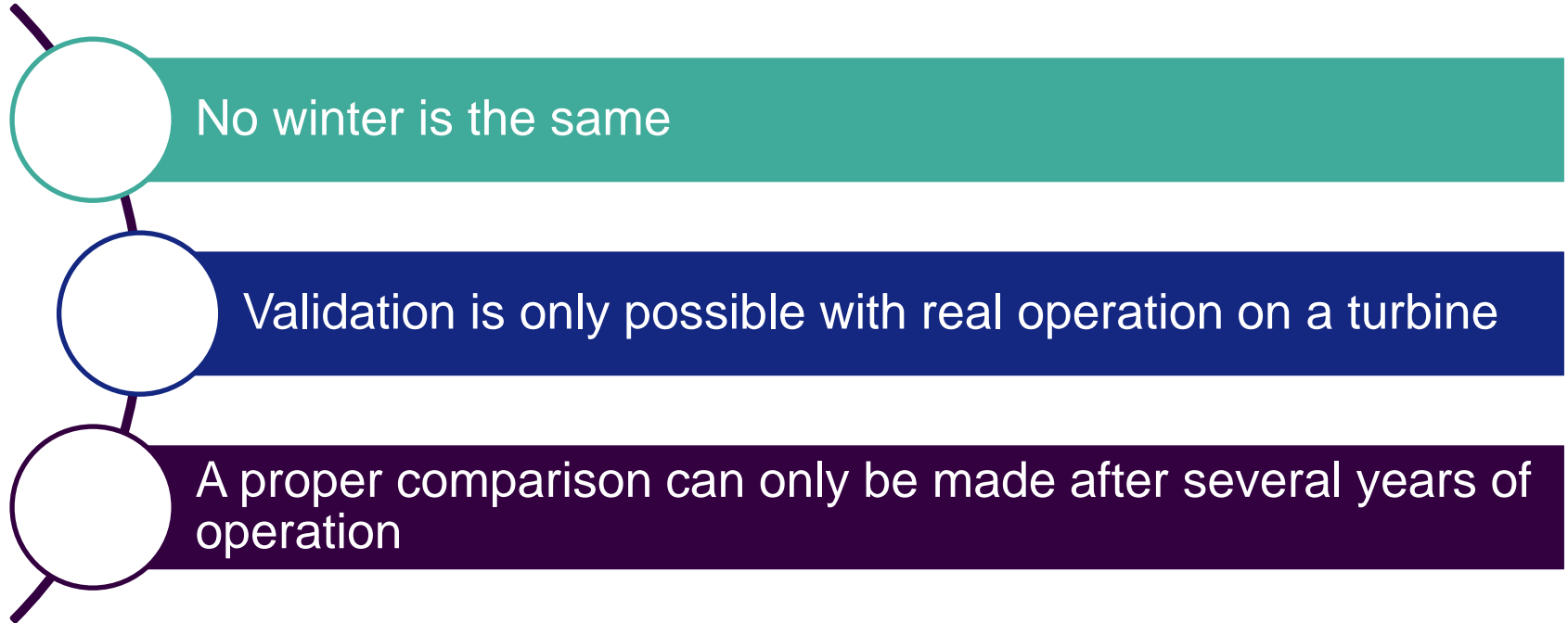


Lower cost: one server for multiple turbines

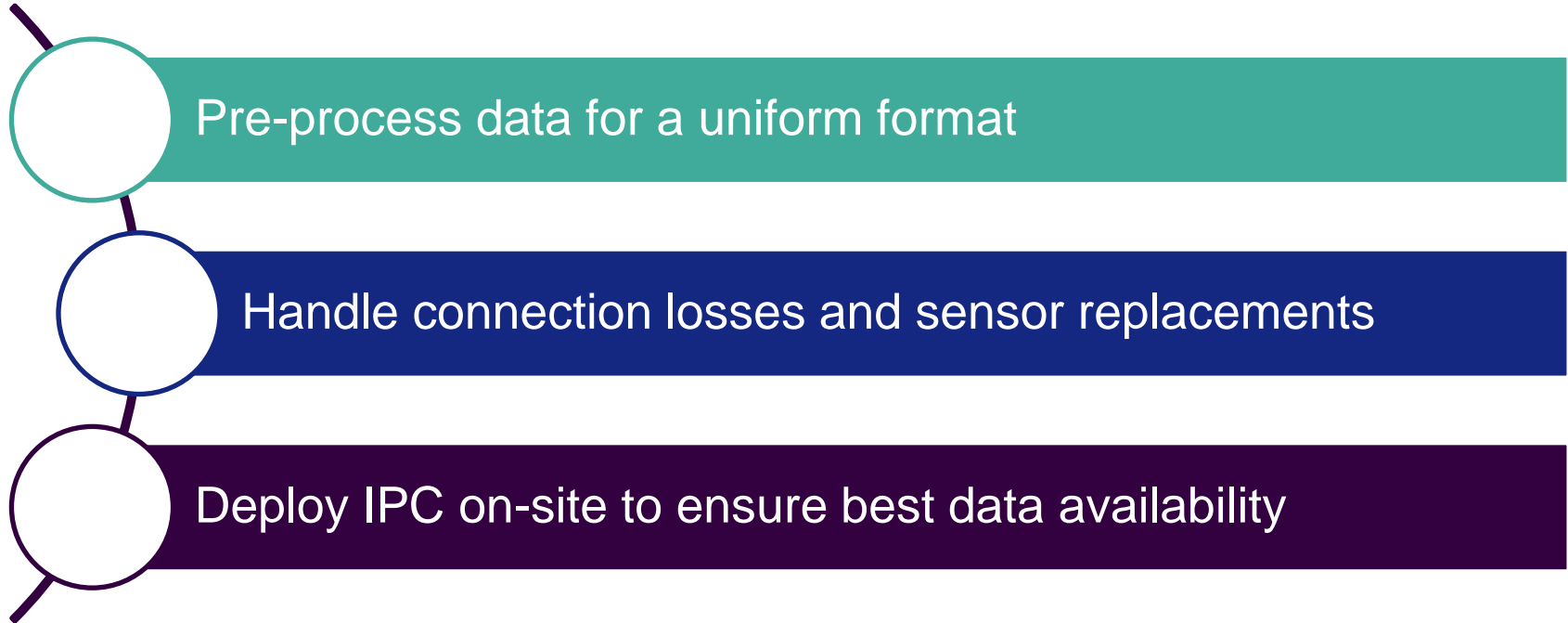
DATAFLOW ARCHITECTURE



THE VALIDATION PROBLEM



SUMMARY: DATA QUALITY



OUTLOOK

Validation

- Additional locations
- Other turbine types

Long-term tests

- Algorithm fine tuning & adaptations
- Quantification of efficiency

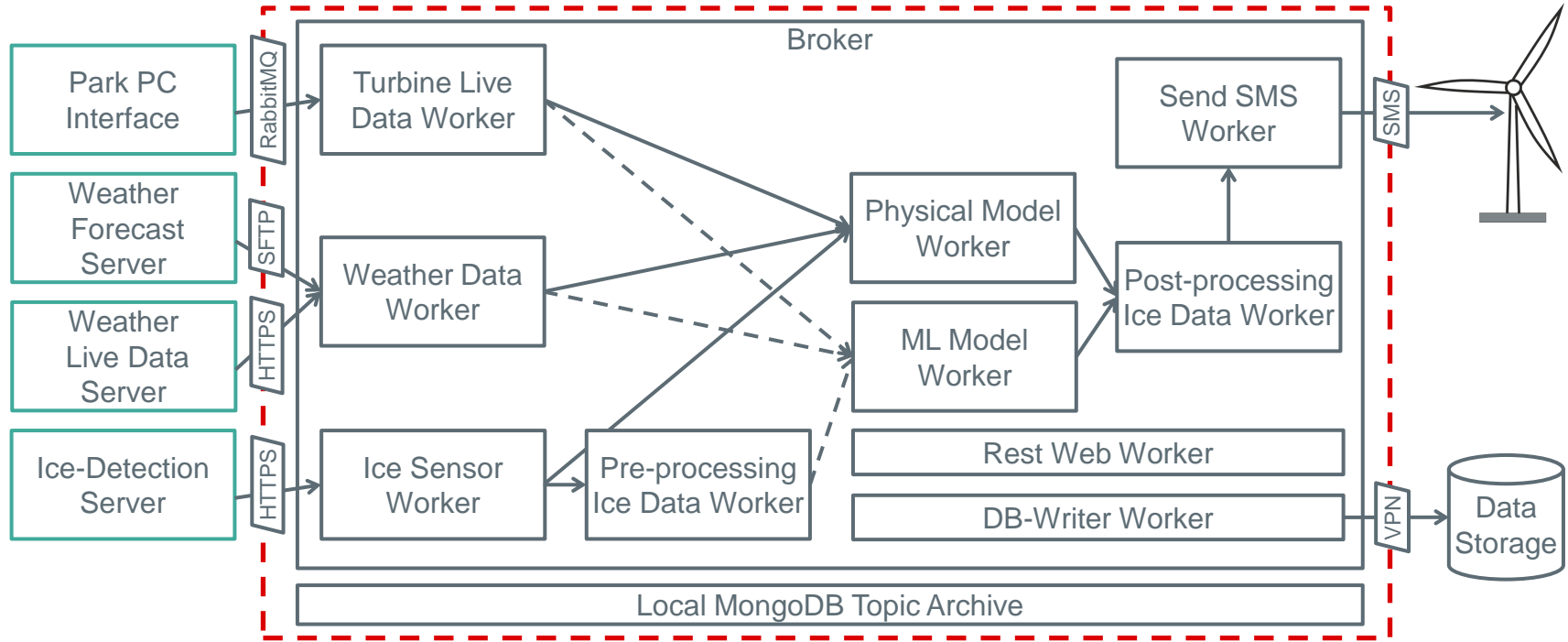
ACKNOWLEDGEMENT



This work was partially funded by the “Energy Research” programme of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (Grant No. 38160077)

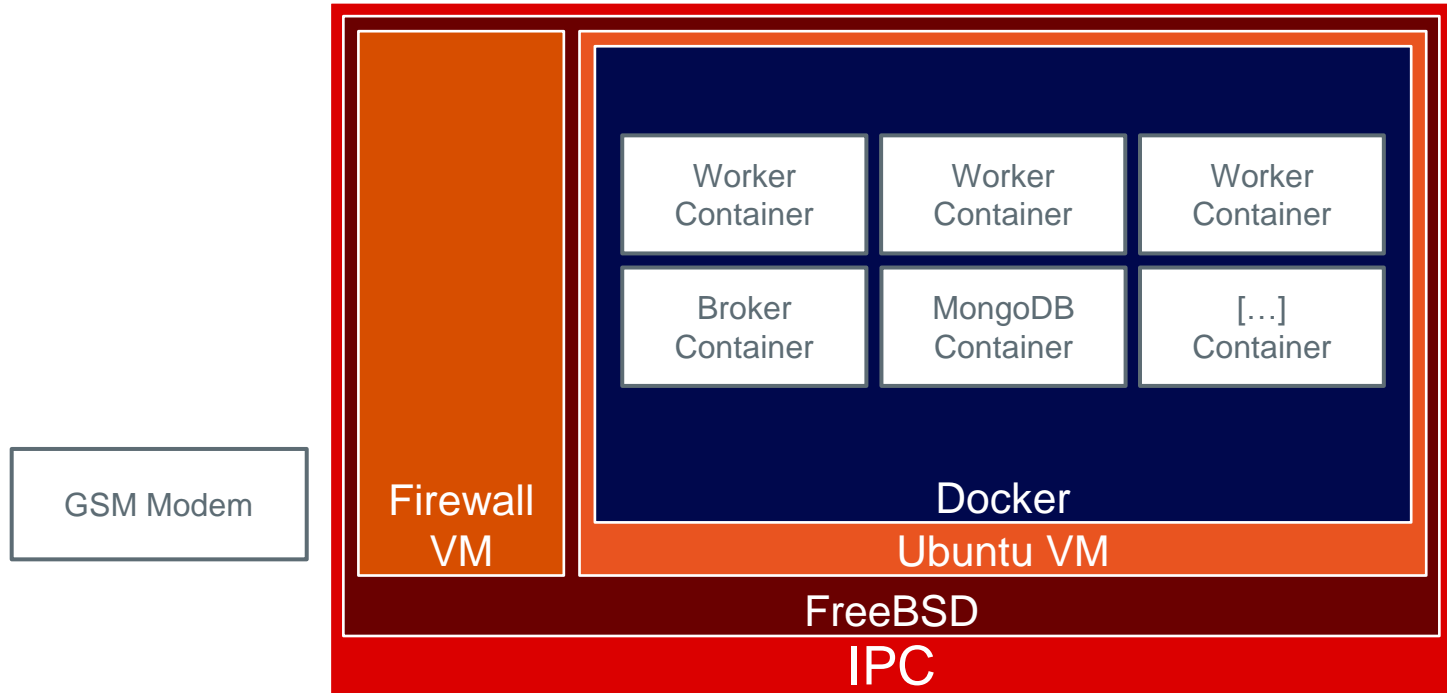
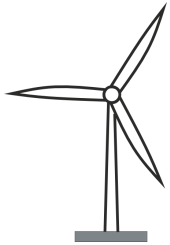


DATAFLOW ARCHITECTURE



DETAILED DATA VIEW

SYSTEM ARCHITECTURE



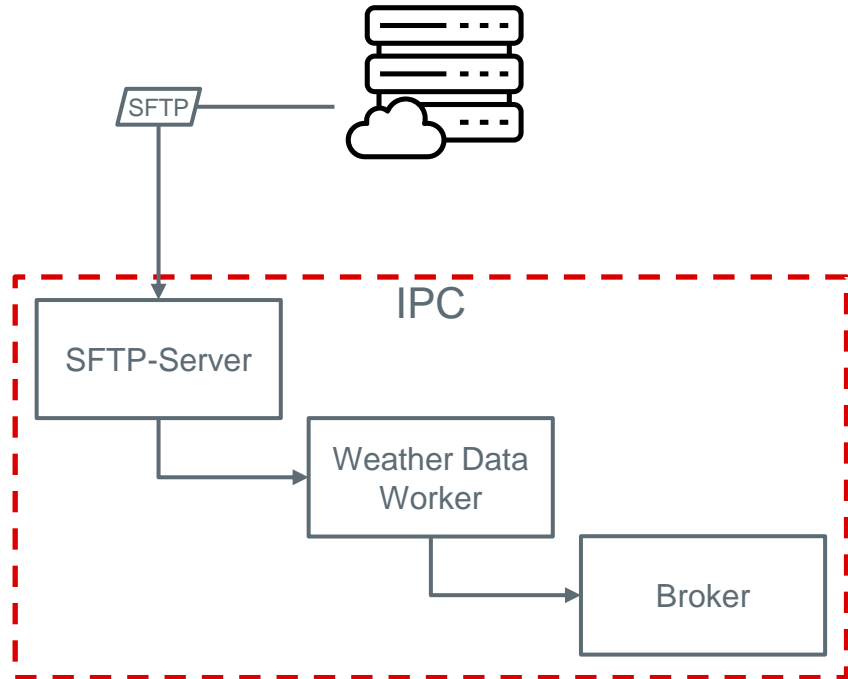
WEATHER FORECASTS

Calculate forecast for
48h every 12h

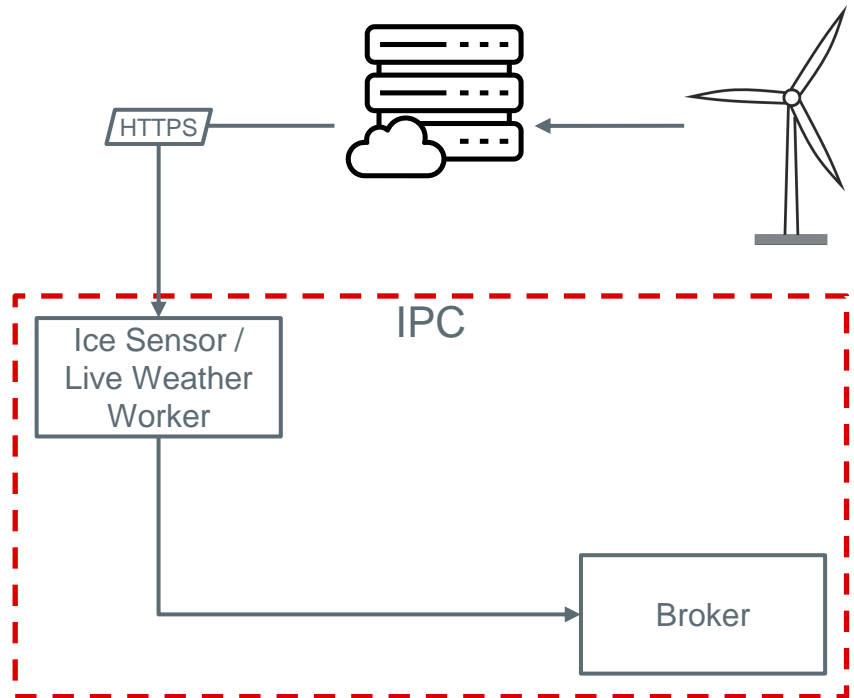
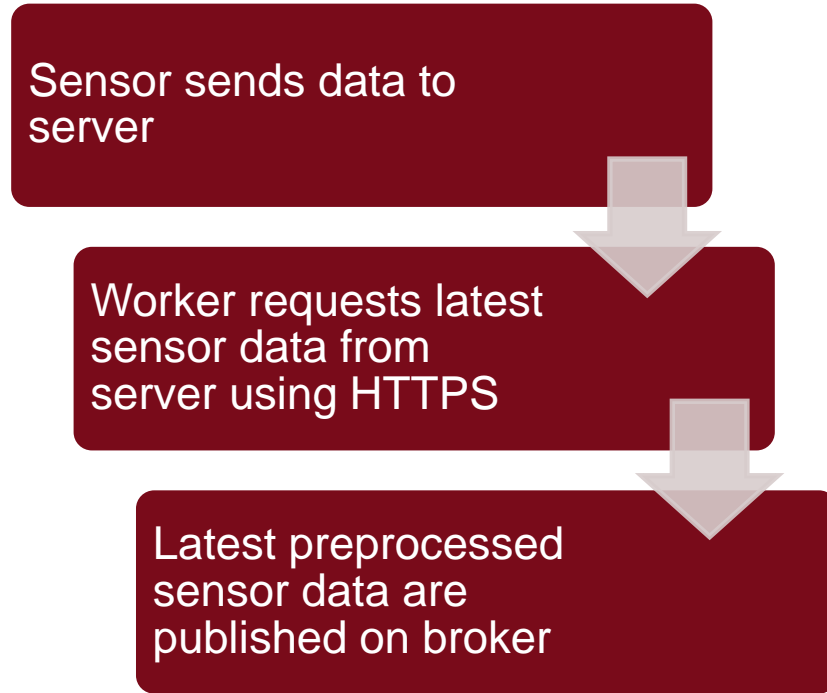
Send forecast csv files
to IPC using SFTP

Detect new csv files on
SFTP-Server

Publish parsed csv
forecast to broker



ICE-DETECTION, WEATHER STATION DATA

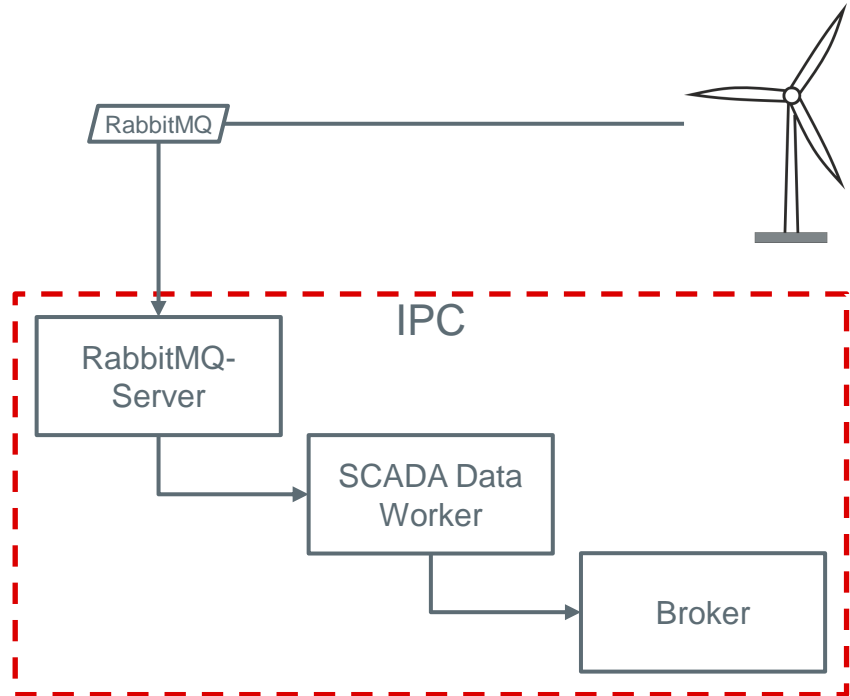


TURBINE LIVE DATA

Park PC publishes
SCADA-Data to
RabbitMQ-Server

Worker subscribes
to new data on
RabbitMQ

Publish parsed
SCADA-Data to
broker



C++ DATA EXCHANGE SYSTEM


[3]

Flexible efficient data serializer for many programming languages


[4]

ZeroMQ is a proven open-source communication library


[5]
[5]
[5]

Runs on Windows, Linux, Docker locally and over the network


[5]
[5]
[5]

Interface available in C++, Python, Matlab to implement Workers


[6]

Logging of each written topic in MongoDB



< 1ms RTT

IPC BECKHOFF CX2033



[7]

MODEM TELTONIKA RUTX14



[8]

PICTORIAL SOURCES

1. [Verbund AG](#)
2. [ffg.at](#)
3. [msgpack.org](#)
4. [zeromq.org](#)
5. [wikimedia.org](#)
6. [mongodb.com](#)
7. [beckhoff.com](#)
8. [teltonika-networks.com](#)