Marinet2 Bring Your Own Data (BYOD) Workshop
23rd June 2020

DOI: 10.5281/zenodo.3843078

MaRINET2

Marinet2 (meta)data 'standard'

Nikola Vasiljevic, DTU Wind Energy

@MaRINET2_EU

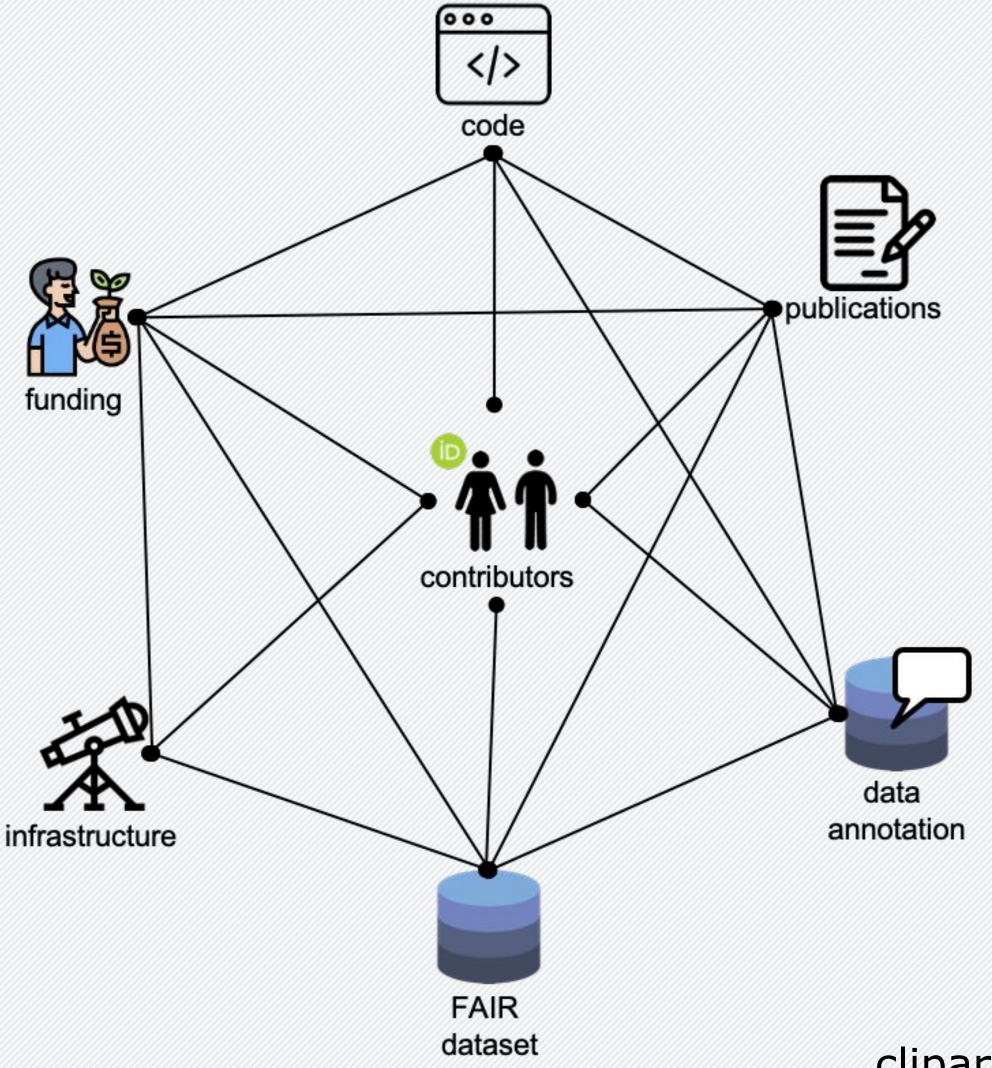
Outline

- Why ‘standardization’ of metadata and data
- Data encapsulation
- Metadata schema for data streams
- Metadata schema for datasets
- Encapsulating schemas in NetCDF

Why standardization ?

- Make your data consistent and clear
- Consistent is ensuring that the output is reliable so that related data can be identified using common terminology and format
- Clear is to ensure that the data can be easily understood by those who are not involved with the data creation process
- Allows you to build/reuse (data) tools
- Increases your scientific impact

Why metadata?



Full chain of custody

cliparts from www.flaticon.com

Data encapsulation

Definitions and icons used across slides

Data stream represents recorded values of single parameter

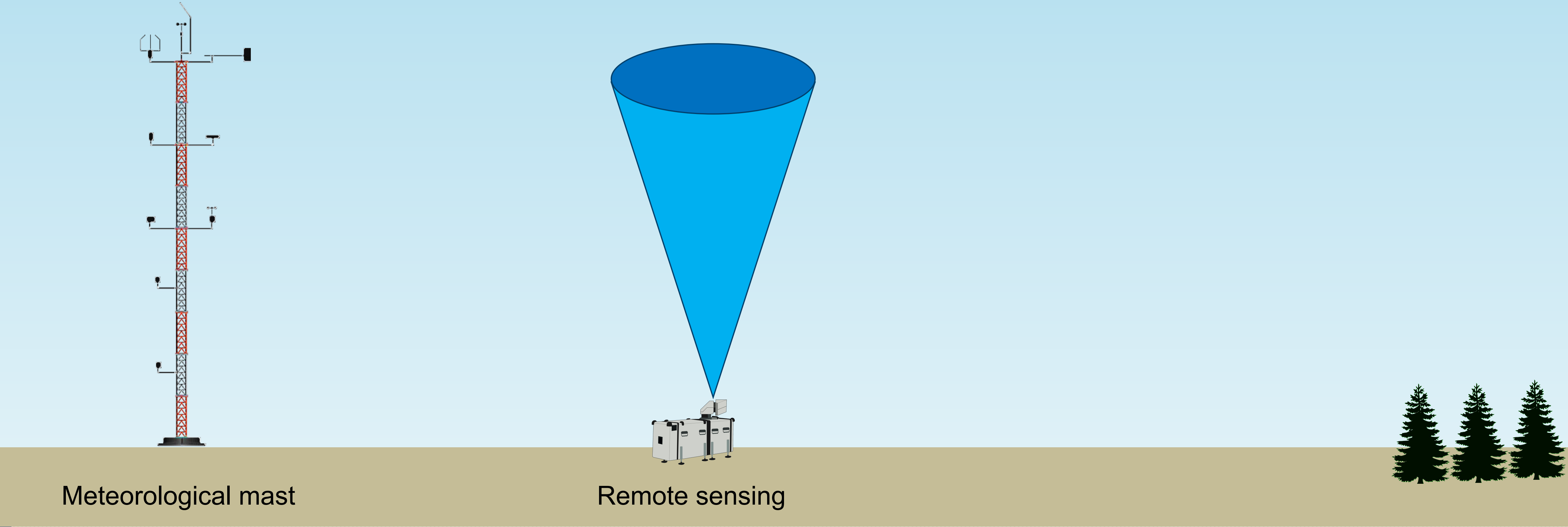
Dataset consists of one or multiple data streams

Data collection consists of one or multiple datasets

cliparts from www.flaticon.com

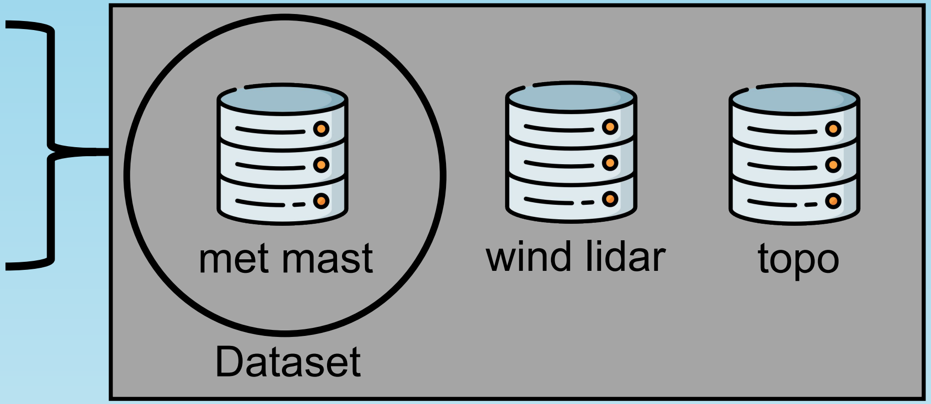
Hypothetical experimental setup

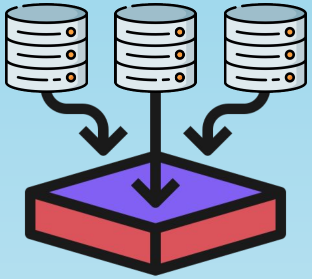
1. Meteorological mast
2. Remote sensing device
3. Topographic information

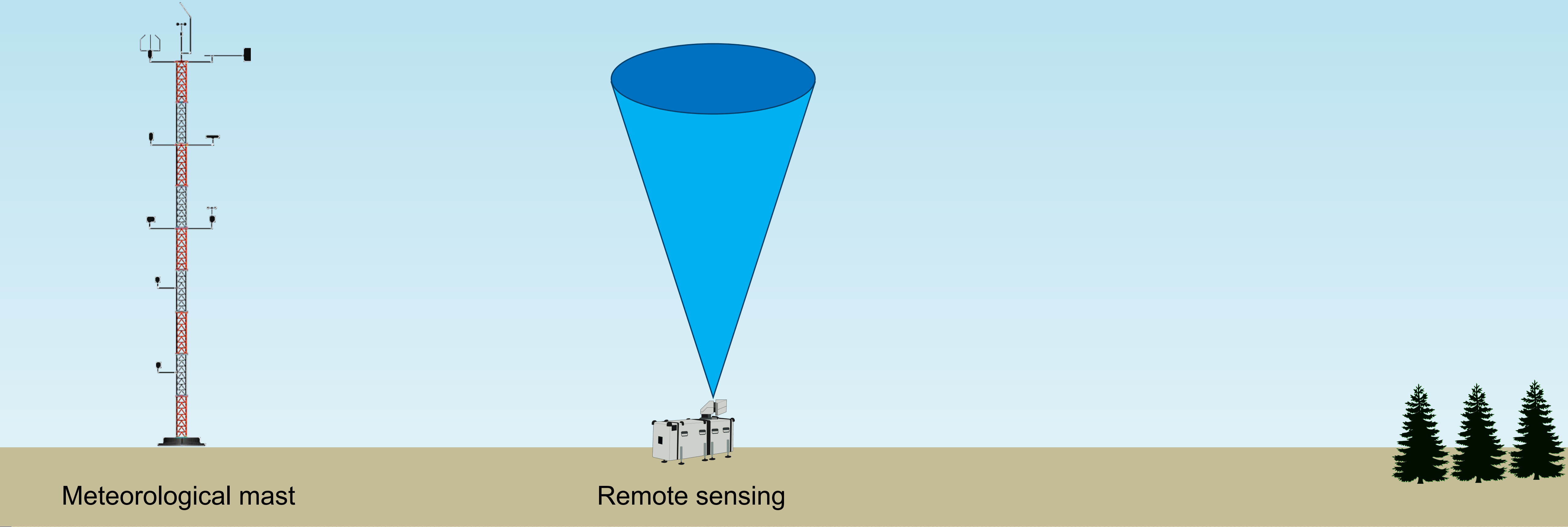


Hypothetical experimental setup

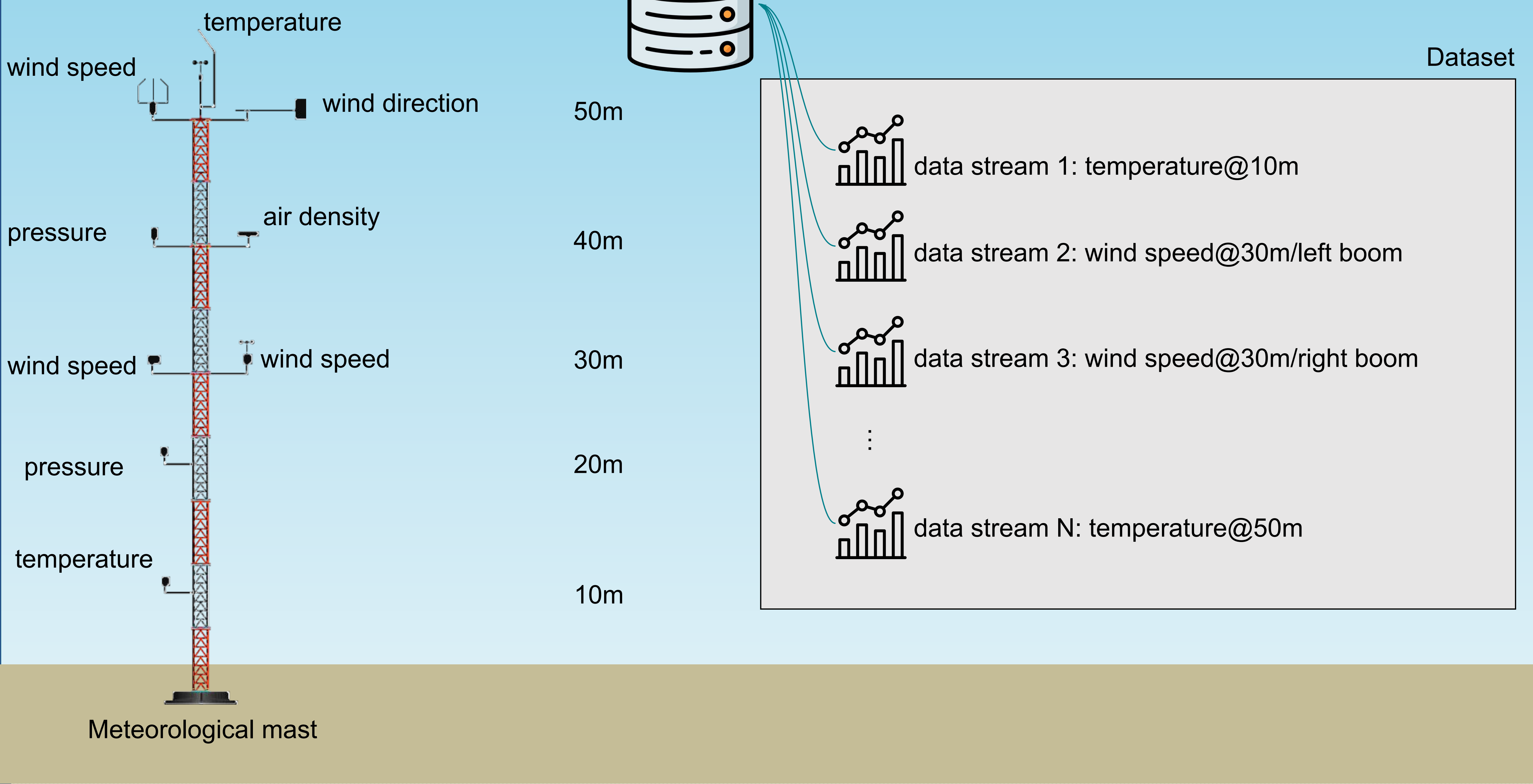
1. Meteorological mast
2. Remote sensing device
3. Topographic information







Dataset (met mast)



Dataset

data stream 1: temperature@10m

data stream 2: wind speed@30m/left boom

data stream 3: wind speed@30m/right boom

⋮

data stream N: temperature@50m

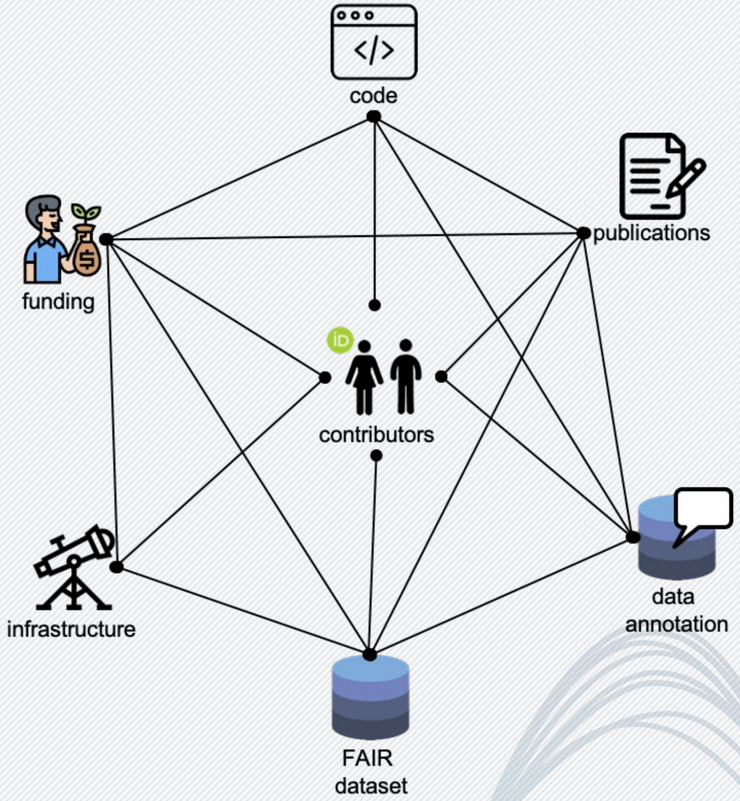
We need metadata for

- Data streams
- Data sets
- Data collections*

*(not covered in Marinet2 project)

What we want to know about data

- How it was created
- Who create it
- In what state the data are
- …



Metadata schema for datasets

Metadata grouping

- **Discovery and identification**
  *Metadata that will help end-users (**humans** and/or **machines**) to discover and identify data*
- **Publication information**
  Comprises information on the publish data
- **Used conventions**
  Informs end-users about the used conventions, such as file format, metadata schema, etc.
- **Coverage**
  Informs end-users on the spatio-temporal coverage of data, its high-level structure, etc.
- **Data provenance**
  Full chain of custody on what happened with data from their early creation until the publication

Schema: https://github.com/Marinet2/metadata-schema/blob/master/dataset_schema.yaml

Discovery and identification metadata elements

- title
- summary
- keywords
- keywords_vocabulary
- dataset_id
- dataset_id_authority
- is_part_of
- is_related_to
- infrastructure_id
- site
- creator_name
- creator_email
- creator_id
- creator_role
- contributor_name
- contributor_email
- contributor_id
- contributor_role
- project_funder
- project_name
- project_id
- project_url
- data_mode

Discovery and identification metadata elements

- **title**
- **summary**
- **keywords**
- keywords_vocabulary
- **dataset_id**
- dataset_id_authority
- is_part_of
- is_related_to
- infrastructure_id
- site
- **creator_name**
- **creator_email**
- **creator_id**
- creator_role
- contributor_name
- contributor_email
- contributor_id
- contributor_role
- project_funder
- project_name
- project_id
- project_url
- **data_mode**

Elements in **bold** are mandatory
Greyed-out elements are recommended/optional

title

- Concise title of dataset
- *Suggestion*: follow the same approach like in giving titles to journal articles

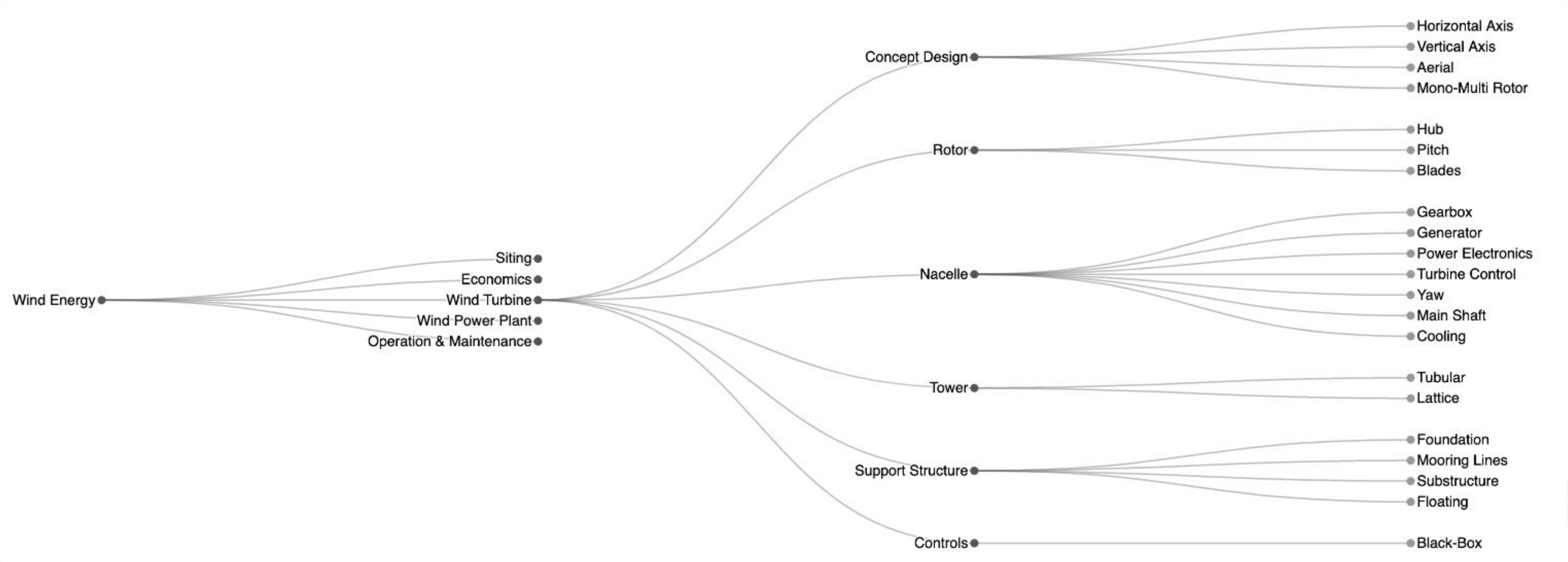
summary

- Summary (i.e., abstract) which describes dataset

keywords

- Keywords tag and enrich dataset indicating (for example) for what dataset can be used for, what parameters dataset contains, what type of instrument was used to create it, etc.
- Keywords should be provided as a list of terms: *['keyword_1', 'keyword_2', …, 'keyword N’]*
- Keywords ideally should be sourced from community-driven vocabularies, for example: *http://data.windenergy.dtu.dk/taxonomy/topics.html*

http://data.windenergy.dtu.dk/taxonomy/topics.html



keywords_vocabulary

- When some or all keywords are sourced from established vocabularies provide a list of URI's to the vocabularies
- Let say my **keywords=[‘GUST WIND SPEED’, ‘Wind mapping’]**
- The first keyword is sourced from BODC ontology with URI: http://vocab.nerc.ac.uk/collection/P09/current/
- The second keyword is sourced from Wind Energy Taxonomy of Topics with URI: http://data.windenergy.dtu.dk/taxonomy/topics.html
- Accordingly:

keywords_vocabulary=[‘http://vocab.nerc.ac.uk/collection/P09/current/ ‘, ‘http://data.windenergy.dtu.dk/taxonomy/topics.html**’]**

dataset_id

- Unique and persistent identifier of dataset
- Typically this will be DOI (Digital Object Identifier)

dataset_id_authority

- An URI of the authority body that generated **dataset_id**

- For example in case dataset_id=DOI, then:

  dataset_id_authority='https://www.doi.org/'

is_part_of & is_related_to

- If dataset is part of data catalog **is_part_of** contains URI of data catalog

- In case when interpretation of current dataset depends on other datasets, or when dataset is split to multiple files (or DBs for example) **is_related_to** contains URIs to related resources

infrastructure_id

- Unique and persistent identifier(s) of infrastructure(s) that generated dataset
- Infrastructures could be virtual (models, HPC) or physical (instruments)
- For example, WindScanner: https://www.vindenergi.dtu.dk/english/research/research-facilities/windscanner

site

- Location where data was created
- This metadata information is important for field experiments

creator_ (name, email, id, role)

- **creator_name**:
  - Ordered list of LastName FirstName of those who created the dataset
  - Order the list according to the level of contributions

- **creator_email**:
  - List of creators' email addresses which follows the order of creator_name

- **creator_id**:
  - List of creators' ORCID IDs which follows the order of creator_name
  - Your creators should have ORCID IDs!

- **creator_role**:
  - List of roles creators had in the dataset creation
  - If individual creator had multiple roles, this could be provided as list of list (array of array)
  - For roles one can use CredIT taxonomy
  - Or employ roles shown in the following slide

creator vs contributor

SIMULATION DATA

Person must participated in **all** following activities to be consider a creator:

1. Design of the simulation setup
2. Create input files
3. Run simulations
4. Collect & Store simulation data

MEASUREMENT DATA

Person must participated in the following activity to be consider a creator:

1. Design of the experiment

   and at least in **two** of the following activities:

2. Campaign installation/maintenance/decommissioning
3. Campaign monitoring
4. Data collection and preparation for data analysis

contributor_ (name, email, id, role)

- Same as **creator_** (name, email, id and role)

project_ (funder, name, id, url)

- project_funder:
  - list of project(s) funder(s)

- project_name:
  - list of project(s) that financed data creation

- project_id:
  - list of id(s) given by project funder (usually project number)

- project_url:
  - URL to the project(s) web site(s)

data_mode

- This attribute can take one of the following values: "Raw", "Provisional", "Delayed-mode" or "Mixed"

- Definition of each mode is as following:
  - **Raw** represents unprocessed data.
  - **Provisional** data means that some calibration/processing of data may have been done, but the data is not thought to be fully processed.
  - **Delayed-mode** data represents data published after all calibrations and quality control procedures have been applied on the internally recorded or best available original data. This is the best possible version of processed data.
  - **Mixed** data indicates that the dataset contains data in more than one of the above states.

Publication information metadata elements

- license
- distribution_statement
- publisher_name
- publisher_email
- publisher_url
- update_interval

Publication information metadata elements

- **license**
- distribution_statement
- **publisher_name**
- publisher_email
- publisher_url
- update_interval

Elements in **bold** are mandatory
Greyed-out elements are recommended/optional

license

- This element describes terms and conditions on data usage
- Typically we use one of Creative Commons license, such as for example CC BY 4.0
- You must provide the license since without it end-users do not have information on how they can use your data

distribution_statement

- Notation that marks dataset according to its security classification to ensure it is circulated only among the authorized recipients.
- Your institute might have ‘default’ distribution statements
- Otherwise you can use one of the following:
  - Statement A : Approved for public release; distribution is unlimited.
  - Statement B : Approved for public release; distribution is defined by license.
  - Statement C : Distribution authorized to (insert authorized entities) only; (fill in reason); (date of determination). Other requests for this document shall be referred to (insert controlling government office).
  - Statement D : Distribution authorized to (insert authorized entities) and their contractors; (fill in reason); (date of determination). Other requests for this document shall be referred to (insert controlling government office).

publisher_ (name, email, url)

- publisher_name:
  Name of entity that published data, it could be for example **Zenodo** or **SEANOE**

- publisher_email:
  Contact e-mail address of publisher

- publisher_url:

  Web site of publisher

update_interval

- In case when dataset is periodically updated provide update interval
- This is typically the case for ‘permanent’ sensor installations such as met masts

Used conventions

- metadata_schema
- format_version
- conventions

Used conventions

- metadata_schema
- format_version
- conventions

Elements in **bold** are recommended
Greyed-out elements are recommended

metadata_schema

- Contains an URL to the metadata schema used to encode information
- In our case, that is: https://fairsharing.org/bsg-s001497/

format_version

- Indicates version of data format
- In our case that would be: Marinet2 NetCDF 0.1

conventions

- Contains a list of conventions used across dataset
- For example, conventions for units, conventions for naming of variables, dimensions, etc.
- In our case, and at least **conventions=["CF-1.8"]**, where CF stands for Climate and Forecast Convention
- Since the wind energy domain does not have convention for naming of parameters and used units, we will probably build a convention prototype as we go through the data conversion `exercise`
- We will use IEC 61400-25-1 standard as an inspiration for this work

Coverage

- feature_type
- cdm_data_type
- coordinate_reference_system
- coordinate_mapping
- spatial_x_min
- spatial_x_max
- spatial_x_units
- spatial_x_resolution
- spatial_y_min
- spatial_y_max
- spatial_y_units
- spatial_y_resolution
- spatial_z_min
- spatial_z_max
- spatial_z_units
- spatial_z_resolution

- time_coverage_start
- time_coverage_end
- time_coverage_resolution
- time_coverage_duration

Coverage

- feature_type
- cdm_data_type
- coordinate_reference_system
- coordinate_mapping
- spatial_x_min
- spatial_x_max
- spatial_x_units
- spatial_x_resolution
- spatial_y_min
- spatial_y_max
- spatial_y_units
- spatial_y_resolution
- spatial_z_min
- spatial_z_max
- spatial_z_units
- spatial_z_resolution

- time_coverage_start
- time_coverage_end
- time_coverage_resolution
- time_coverage_duration

Elements in **bold** are mandatory
Greyed-out elements are recommended

feature_type

- Description of the spatio-temporal shape of the data (required for CF convention)
- It can take following values:
  - **point**
    a single data point (having no implied coordinate relationship to other points)
  - **timeSeries**
    a series of data points at the same spatial location with monotonically increasing times
  - **trajectory**
    a series of data points along a path through space with monotonically increasing times
  - **profile**
    an ordered set of data points along a vertical line at a fixed horizontal position and fixed time
  - **timeSeriesProfile**
    a series of profile features at the same horizontal position with monotonically increasing times
  - **trajectoryProfile**
    a series of profile features located at points ordered along a trajectory
- Your dataset might have a mixture of spatio-temporal shaped, if that's the case **feature_type** will be a list
- More details: http://cfconventions.org/cf-conventions/v1.8.0/cf-conventions.html#_features_and_feature_types

cdm_data_type

- Unidata CDM (common data model) data type used by THREDDS

- It takes one of the following values (definitions are not official but provisional):
  - **Grid:** points distributed over an uniform grid
  - **Image**
  - **Point**: single point measurements
  - **Radial**: points arrange on a scanned arc performed by scanning lidars or radars (e.g., PPI or RHI scan)
  - **Station**: points arrange on a vertical line (e.g., met station data)
  - **Swath**: Satellite data are typically provided as 'swaths'
  - **Trajectory**: multiple points which cannot fit in the previous structures, typical usage aerial sensors complex lidar scanning scenarios

- They are similar but not the same to **feature_type**

coordinate_reference_system

- Describes which spatial coordinate system was used to structure data
- In case it is geographical coordinate system provide its EPSG code, e.g. for lat/lon: **coordinate_reference_system**=’EPSG:4326’
- Otherwise, if you for example running a wind tunnel measurements set this to: **coordinate_reference_system**=’CUSTOM’

coordinate_mapping

- Describes relative coordinates x, y and z are mapped to absolute coordinates described by **coordinate_reference_system**
- This attribute should be provided in a form of ‘key-value’ pairs,i.e. dictionary
- For example if reference_system=”EPSG:4326”, and we are doing aerial measurements:
  coordinate_mapping=[“x”:”longitude”, “y”:”latitude”, “z”:”height”]
- In case if we are doing underwater measurements :
  coordinate_mapping=[“x”:”longitude”, “y”:”latitude”, “z”:”depth”]

spatial_ (x,y,z)_ (min,max,resolution,units)

- **spatial_x_min** - minimum value of coordinate
- **spatial_x_max** - maximum value of coordinate
- **spatial_x_resolution** - 'separation' of consecutive coordinates along a specific axis, in this case x
- **spatial_x_units** - units of coordinate values
- Providing these attributes forms basis to calculate bounding box of data

time_coverage_(start, end, resolution, duration)

- **time_coverage_start**
  - Start date of the data in UTC
  - Time must be specified as a string according to the ISO8601 standard: "YYYY-MM-DDThh:mm:ssZ".
- **time_coverage_end**
  - End date of the data in UTC
  - Time must be specified as a string according to the ISO8601 standard: "YYYY-MM-DDThh:mm:ssZ".
- **time_coverage_resolution**
  - Interval between records, i.e. sampling time
  - ISO8601 standard must be used: PnYnMnDTnHnMnS, e.g. P0Y0M0DT0H0M0.1S, corresponds to 0.1s or 10 Hz
- **time_coverage_duration**
  - Duration of the time coverage of the data
  - ISO8601 standard must be used: PnYnMnDTnHnMnS

Provenance metadata elements

- date_create
- date_update
- processing_level
- history

Provenance metadata elements

- **date_create**
- **date_update**
- processing_level
- history

Elements in **bold** are mandatory
Greyed-out elements are recommended

date_create

- Date on which the data file was created

- Must be in UTC

- Time must be specified as a string according to the ISO8601 standard: "YYYY-MM- DDThh:mm:ssZ”

date_update

- Timestamp specifying when the contents (i.e. its attributes and/or values) of the file were last changed

- Must be in UTC

- Timestamp must be a string according to the ISO8601 standard: "YYYY-MM- DDThh:mm:ssZ”

history

- Provides an audit trail for modifications to the original data (i.e., a full chain of custody)

- It should contain a separate line for each modification, with each line beginning with a timestamp, and including user name, modification name, and modification arguments.

- Example:

  history=”2020-05-26T09:14:23Z nikola Creation initial creation
  2020-05-30T23:14:23Z pawel Update filtered data”

processing_level

- Level of processing and quality control applied to data
- This attribute refines **data_mode**
- It will depend on a specific community definitions on processing levels
- For example for wind lidars based on e-WindLidar conventions:
  - LEVEL 0: Backscatter signal
  - LEVEL 1A: Individual Doppler spectra
  - LEVEL 1B: QC (i.e. filtered) Individual Doppler spectra
  - LEVEL 1C: QC Averaged Doppler spectra
  - LEVEL 2A: Estimated non-averaged radial velocity
  - LEVEL 2B: Estimated non-averaged QC radial velocity
  - LEVEL 2C: Estimated averaged QC radial velocity
  - LEVEL 3: Reconstructed wind
  - LEVEL 4: Extracted flow related parameters (e.g., wind turbine wake width)

Metadata schema for data streams

Metadata grouping

- **Discovery and identification**
  *Metadata that will help end-users (**humans** and/or **machines**) to discover and identify data stream*

- **Values and units**
  Information on stored values and values units

Schema: https://github.com/Marinet2/metadata-schema/blob/master/datastream_schema.yaml

Discovery and identification metadata elements

- short_name
- standard_name
- long_name
- alt_name
- concept_id
- concept_URI
- is_part_of
- is_related_to

Discovery and identification metadata elements

- **short_name**
- **standard_name**
- long_name
- alt_name
- concept_id
- concept_URI
- is_part_of
- is_related_to

Elements in **bold** are mandatory
Greyed-out elements are recommended

short_name

- Short name of data stream

- Typically provided in all CAPITAL letters

standard_name

- Standardized name of data stream which was agreed by community
- In case of metocean parameters standardized names are sourced from CF convention
- **For engineering domains no standard names exist, so we will build them during data conversion**

long_name

- Longer and more explanatory name of data stream comparing to **standard_name** and **short_name**

alt_name

- List of alternative names for data stream
- Optionally instead of names one can provide URIs which hold names and definitions

concept_ (id, URI)

- ID of data stream in restricted vocabulary
- Example, let say you have wind direction as one of data streams in your data set. The concept ‘wind direction’ exist in NERC vocabulary and it has id **WDIR**. Accordingly, you can set **concept_id**=’WDIR’
- Sometimes **concept_id** would match **short_name**
- **concept_URI** represents a resolvable URI which contains definition of concept
- Considering the previous example, that would be: **concept_URI**=’http://vocab.nerc.ac.uk/collection/P09/current/WDIR/’

is_part_of & is_related_to

- **is_part_of** represents PID of dataset to which datastream belongs to

- In case when interpretation of current datastream depends on other datastreams **is_related_to** contains a list of related datastreams

Values and units

- dtype
- units
- valid_min
- valid_max
- valid_range
- allowed_values
- _FillValue
- add_offset
- scale_factor
- compression_type
- compression_level

Values and units

- **dtype**
- **units**
- valid_min
- valid_max
- valid_range
- allowed_values
- **_FillValue**
- **add_offset**
- **scale_factor**
- compression_type
- compression_level

dtype

- Type of data stored in the datastream (e.g., int, float or double)

units

- Units of values

valid_min

- Minimum allowed value

valid_max

- Maximum allowed value

valid_range

- ange expressed as [valid_min, valid_max] if data streams takes continuous values

allowed_values

- List of specific values that the data stream can take in case it is discreet

uncertainty

- Absolute uncertainty of data stream values
- Example
  - Consider wind speed data streams which is produced by cup anemometer
  - Cup has certain uncertainty in measurements, say 0.1 m/s
  - Accordingly **uncertainty=0.1**

_FillValue

- In case of missing values this metadata elements defines value which will be used to fill in data stream
- Fill values must be chosen according to **dtype**
- Typically for float/double it is NaN or for int -999

add_offset and scale_factor

- add_offset:
    - Offset value to be added to all the values of the data stream

- scale_factor:
    - Scaling value to be multiplied to all the values of the variable

- Combination of **add_offset** and **scale_factor** could be used to save space by switching from float to int

compression_type and compression_level

- compression_type:
  - Indicates compression used for data stream

- compression_level:
  - Indicates level of compression according to compression_type

Encapsulating schemas in NetCDF

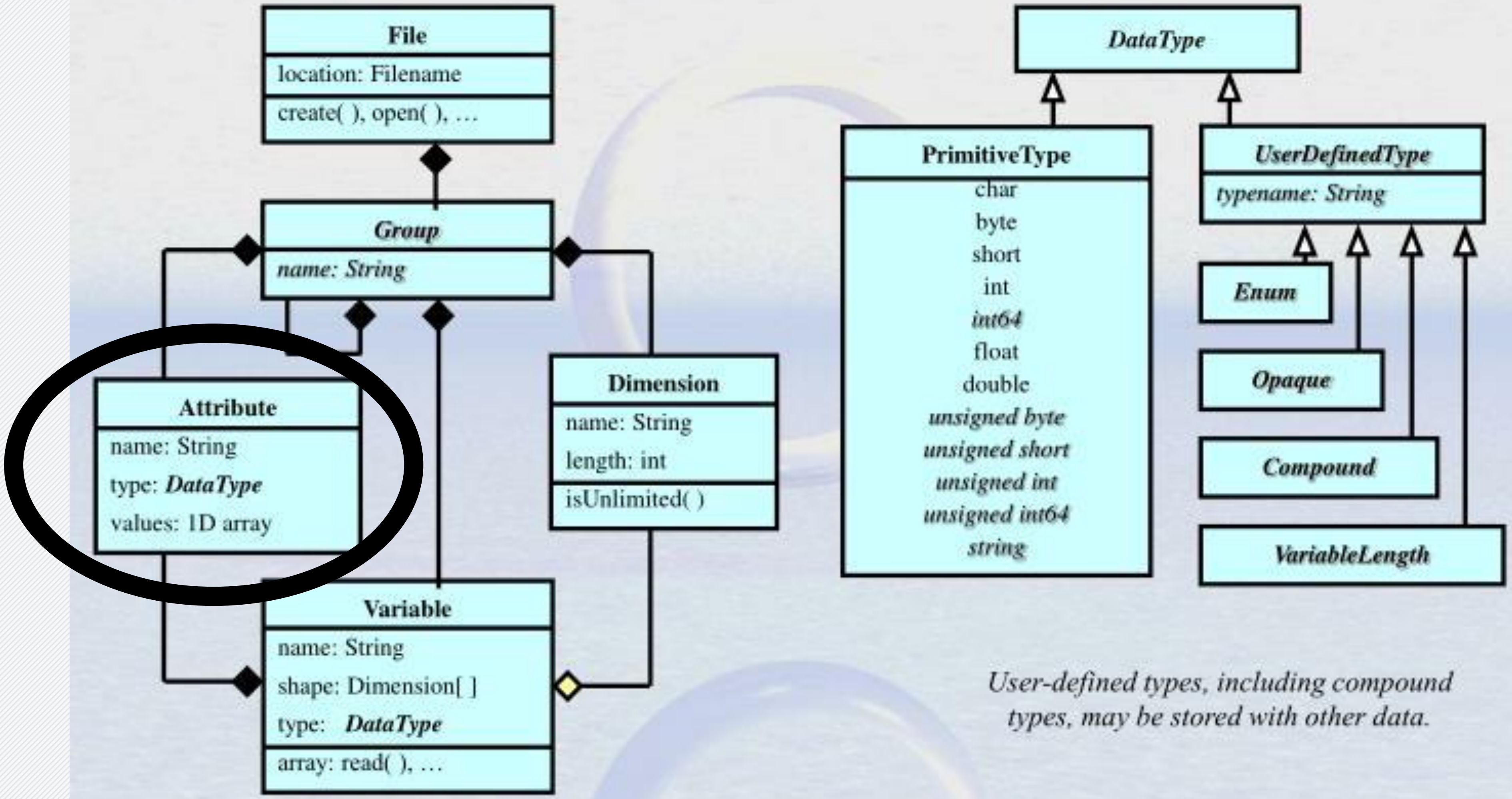
What is NetCDF?

- NetCDF (Network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data developed by Unidata

- The Unidata Program Center supports and maintains netCDF programming interfaces for C, C++, Java, and Fortran. Programming interfaces are also available for **Python**, IDL, MATLAB, R, Ruby, and Perl.

- Data in netCDF format are:
  - **Self-Describing**. A netCDF file includes information about the data it contains.
  - **Portable**. A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
  - **Scalable**. Small subsets of large datasets in various formats may be accessed efficiently through netCDF interfaces, even from remote servers.
  - **Appendable**. Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
  - **Sharable**. One writer and multiple readers may simultaneously access the same netCDF file.
  - **Archivable**. Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

NetCDF-4 Data Model



User-defined types, including compound types, may be stored with other data.

A file has a top-level unnamed group. Each group may contain one or more named subgroups, variables, dimensions, and attributes. A variable may also have attributes. Variables may share dimensions, indicating a common grid. One or more dimensions may be of unlimited length.