

# [UC]<sup>2</sup>-Datenstandard "*Stadtklima im Wandel*"

Version 1.5.2

22.11.2022

Dieter Scherer<sup>1</sup>, Ute Fehrenbach<sup>1</sup>, Tom Grassmann<sup>1</sup>, Achim Holtmann<sup>1</sup>, Fred Meier<sup>1</sup>, Katharina Scherber<sup>1</sup>, Dirk Pavlik<sup>2</sup>, Thomas Höhne<sup>3</sup>, Farah Kanani-Sühring<sup>4</sup>, Björn Maronga<sup>4</sup>, Felix Ament<sup>5</sup>, Sabine Banzhaf<sup>6</sup>, Ines Langer<sup>6</sup>, Guido Halbig<sup>7</sup>, Martin Kohler<sup>8</sup>, Ronald Queck<sup>9</sup>, Sebastian Stratbücker<sup>10</sup>, Matthias Winkler<sup>10</sup>, Robert Wegener<sup>11</sup>, Matthias Zeeman<sup>12</sup>

<sup>1</sup>Technische Universität Berlin, Fachgebiet Klimatologie

<sup>2</sup>GEO-NET Umweltconsulting GmbH

<sup>3</sup>IDU IT+Umwelt GmbH

<sup>4</sup>Leibniz Universität Hannover, Institut für Meteorologie und Klimatologie

<sup>5</sup>Universität Hamburg, Meteorologisches Institut

<sup>6</sup>Freie Universität Berlin, Institut für Meteorologie

<sup>7</sup>Deutscher Wetterdienst

<sup>8</sup>Karlsruher Institut für Technologie, Institut für Meteorologie und Klimaforschung, Troposphärenforschung

<sup>9</sup>Technische Universität Dresden, Professur für Meteorologie

<sup>10</sup>Fraunhofer-Institut für Bauphysik

<sup>11</sup>Forschungszentrum Jülich GmbH, Institut für Energie- und Klimaforschung, IEK-8: Troposphäre

<sup>12</sup>Karlsruher Institut für Technologie, Institut für Meteorologie und Klimaforschung, Institut für Atmosphärische Umweltforschung

## Arbeitsgruppe "*Datenmanagement*"

der Fördermaßnahme

**[UC]<sup>2</sup> Urban Climate Under Change  
Stadtklima im Wandel**

gefördert durch das



**Bundesministerium  
für Bildung  
und Forschung**

## Vorwort

Dieses Dokument wurde von der Arbeitsgruppe "Datenmanagement" im Rahmen der Fördermaßnahme "Stadtklima im Wandel" ("Urban Climate Under Change"; [UC]<sup>2</sup>) erstellt. Die Fördermaßnahme stellt sich der Herausforderung, Großstädten und Stadtregionen ein wissenschaftlich fundiertes, praxistaugliches Instrumentarium zur Bewältigung der mit heutigen und zukünftigen Wetter- und Klimabedingungen und Luftbelastungen einhergehenden Probleme an die Hand zu geben. Bereits heute verursachen Starkniederschläge und Stürme, Hitze- und Kältewellen, Trockenperioden und Dürren sowie Episoden mit erhöhter Luftbelastung gravierende wirtschaftliche Schäden und Gesundheitsbelastungen bis hin zu Todesfällen. In Städten besteht durch die Konzentration der Bevölkerung einerseits sowie die von urbanen Strukturen verursachten Modifikationen der atmosphärischen Prozesse andererseits ein besonders hoher diesbezüglicher Handlungsbedarf. Durch die regionalen Folgen des globalen Klimawandels werden sich diese Probleme in den nächsten Jahrzehnten weiter verstärken und zusätzliche Anstrengungen erforderlich machen.

Ein zentrales Ziel der Fördermaßnahme (Modul A) ist die Entwicklung, Validierung und Anwendung des auf dem Grobstrukturmodell (Large Eddy Simulation; LES) basierenden *PALM (Parallelized Large-Eddy Simulation Model)* gebäudeauflösenden Stadtklimamodells *PALM-4U (PALM for urban applications; gesprochen: PALM for you)* für ganze Großstädte wie Stuttgart oder Berlin. Bisher verfügbare Stadtklimamodelle sind entweder zu grobmaschig, um die gerade für die Planung von Maßnahmen zur Erhaltung und Verbesserung des Stadtklimas, zum Klimaschutz und zur Anpassung an den Klimawandel sowie zur Verbesserung der Luftreinhaltung äußerst wichtigen mikroskaligen (Gebäude und Straßenschluchten) und lokalskaligen (Stadtquartiere) Prozesse auflösen zu können, oder sie decken nur kleinere Stadtgebiete ab und können nicht an großräumige numerische Modelle gekoppelt werden, wie sie in der Wettervorhersage oder für regionale Klimaprojektionen zum Einsatz kommen.

Damit *PALM-4U* in der Lage ist, einen Beitrag zur Lösung dieser Probleme zu leisten, muss dieses zunächst hinsichtlich seiner Leistungsfähigkeit getestet und beurteilt werden. Dazu sind umfassende Daten zu Wetter, Klima und Luftqualität in Großstädten erforderlich (Modul B). Leider sind solche Daten bis heute nur begrenzt verfügbar, was insbesondere für mehrjährige oder gar multidekadische atmosphärische Langzeitbeobachtungen in Städten gilt. Daher werden im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme bereits verfügbare Daten aufbereitet und fehlende Daten über Langzeitmessungen und Intensivmesskampagnen neu erhoben. Dazu sind auch verbesserte Konzepte und Analysewerkzeuge erforderlich, deren Entwicklung ein wichtiges Ziel der Fördermaßnahme darstellt. Darüber hinaus sollen die Beobachtungsdaten auch einer eigenständigen Verwertung für spezifische Anwendungen zugeführt werden können.

Eine weitere unabdingbare Anforderung an *PALM-4U* ist seine Praxistauglichkeit. Dies bedeutet, dass die Modellergebnisse einerseits belastbare Aussagen für eine Vielzahl konkreter Anwendungen ermöglichen sollen, und andererseits die Anforderungen an die Rechnerinfrastruktur und Fachkenntnisse der potenziellen Nutzer/innen möglichst gering sind. Daher besteht ein weiteres zentrales Ziel der Fördermaßnahme (Modul C), ausgewählte Anwendungsbeispiele und Nutzerkreise direkt in die Modellentwicklung und Messdatenerhebung zu integrieren, um die Praxistauglichkeit des Stadtklimamodells und der Messkonzepte und Analysewerkzeuge sicherzustellen.

Die [UC]<sup>2</sup>-Fördermaßnahme ([www.uc2-program.org](http://www.uc2-program.org)) besteht aus vier Verbundprojekten mit insgesamt 30 Teilprojekten, welche in den drei Modulen die o.a. Aufgaben wahrnehmen: Modul A: "Model-based city planning and application in climate change" (MOSAIC); Modul B: "Three-Dimensional Observation of Atmospheric Processes in Cities" (3DO); Modul C: "Climate Models for Practice" (KliMoPrax) und "Review of practical and user serviceability of an urban climate model to foster climate proof urban development" (UseUClim).

Die Arbeitsgruppe "Datenmanagement" wurde eingerichtet, um durch eine Standardisierung der Regeln zur Benutzung der Daten ([UC]<sup>2</sup> Data Policy), der Speicherung der Daten in einem einheitlichen Format ([UC]<sup>2</sup>-Datenstandard) sowie zur Bereitstellung der Daten in einem Datenmanagementsystem (DMS) eine effiziente Zusammenarbeit innerhalb der [UC]<sup>2</sup>-Fördermaßnahme sicherzustellen.

## Inhalt

Vorwort .....	ii
Inhalt.....	iii
1 Einleitung.....	1
1.1 Zielsetzung.....	1
1.2 Konventionen .....	1
2 Globale Attribute.....	3
2.1 Allgemeine globale Attribute .....	3
2.2 Globale Attribute zum zeitlichen Bezug der Daten .....	6
2.3 Globale Attribute zum räumlichen Bezug der Daten .....	7
2.4 Das globale Attribut <b>featureType</b> .....	9
2.5 Optionale globale Attribute.....	9
3 Dimensionen, Koordinatenvariablen und Hilfskoordinatenvariablen.....	10
3.1 Zeit.....	10
3.2 Vertikale Koordinaten .....	11
3.3 Horizontale Koordinaten .....	13
3.4 <i>PALM-4U</i> -Modellgitter .....	15
3.5 Oberflächen.....	16
3.6 Intervallgrenzen von Koordinaten- und Hilfskoordinatenvariablen .....	19
4 Datenvariablen .....	20
4.1 Obligatorische Attribute.....	21
4.2 Optionale Attribute .....	22
4.3 Hilfsvariablen und Flags.....	24
4.4 Spektrale Daten .....	24
5 Unterstützte <b>featureType</b> -Varianten .....	26
5.1 Zeitreihen ( <b>featureType</b> ="timeSeries").....	26
5.2 Zeitreihen von Vertikalprofilen ( <b>featureType</b> ="timeSeriesProfile").....	27
5.3 Trajektorien ( <b>featureType</b> ="trajectory") .....	28
6 Dateizeichnungen im DMS .....	29
Anhang .....	1
A1 Beispiel "Multidimensionale Daten" .....	1
A2 Beispiel "Zeitreihen" .....	4
A3 Beispiel "Zeitreihen von Vertikalprofilen".....	7
A4 Beispiel "Trajektorien" .....	10
A5 Beispiel "Hilfsvariablen und Flags" .....	13
A6 Beispiel "Spektrale Daten" .....	16

# 1 Einleitung

## 1.1 Zielsetzung

Im Rahmen der vom *Bundesministerium für Bildung und Forschung (BMBF)* seit Juni 2016 finanzierten Fördermaßnahme „*Stadtklima im Wandel*“ („*Urban Climate Under Change*“; [UC]<sup>2</sup>); [www.uc2-program.org](http://www.uc2-program.org)) wurde ein teilprojekt- und modulübergreifendes Datenmanagementsystem (DMS) entwickelt und zur Nutzung für die Projektpartner der Fördermaßnahme [UC]<sup>2</sup> bereitgestellt.

Die Verwendung eines einheitlichen Datenformates sowie verbindlicher Konventionen für Daten und Metadaten ist Voraussetzung für eine effektive und effiziente Verwaltung aller Daten im DMS sowie für die Entwicklung und Bereitstellung innovativer Werkzeuge zur Analyse und Visualisierung der komplexen und vielfältigen Daten. Nachfolgend werden diese Konventionen erläutert und nachfolgend als [UC]<sup>2</sup>-Datenstandard bezeichnet.

Der [UC]<sup>2</sup>-Datenstandard berücksichtigt sowohl die Anforderungen des neuen Stadtklimamodells *PALM-4U*, das von den Projektpartnern des MOSAIK-Verbunds im Modul A entwickelt wird, als auch die unterschiedlichen Beobachtungsverfahren, welche von den Projektpartnern des 3DO-Verbunds im Modul B zum Einsatz kommen. Dabei muss sichergestellt werden, dass die Daten und Metadaten nicht nur vollständig sind, sondern auch zwischen den beiden Modulen ohne weitere Datenkonvertierung ausgetauscht und genutzt werden können. Nicht zuletzt sollen die Projektpartner des Moduls C (KliMoPrax und UseUClim) sowie Praxispartner ohne spezifisches Detailwissen in der Lage sein, Modell- und Beobachtungsdaten für unterschiedliche Anwendungen nutzen zu können.

Obwohl der [UC]<sup>2</sup>-Datenstandard für die spezifischen Anforderungen der [UC]<sup>2</sup>-Fördermaßnahme entwickelt wurde, sind die Festlegungen so getroffen worden, dass eine Übertragung auf andere Städte, Forschungsprojekte und Anwendungen möglich ist. Darüber hinaus können Datensätze, die dem [UC]<sup>2</sup>-Datenstandard folgen, nicht nur über das DMS, sondern auch über andere Repositorien bereitgestellt werden.

## 1.2 Konventionen

Gemäß den Anforderungen des BMBF werden in der Fördermaßnahme [UC]<sup>2</sup> alle Datensätze, die teilprojektübergreifend verwendet werden, im offenen, selbstbeschreibenden Datenformat *NetCDF* ([www.unidata.ucar.edu/software/netcdf](http://www.unidata.ucar.edu/software/netcdf)) gespeichert, da es weltweit verbreitet ist und hierfür eine Vielzahl offener, kostenloser Schnittstellen und Anwendungen existieren. In *NetCDF*-Dateien können unterschiedlichste Daten und Metadaten gespeichert werden, die mit dem [UC]<sup>2</sup>-Datenstandard definiert und verbindlich festgelegt werden.

Neben der Festlegung auf das *NetCDF*-Datenformat wird auch den Prinzipien der *NetCDF Climate and Forecast (CF) Metadata Conventions Version 1.7* ([cfconventions.org/Data/cf-conventions/cf-conventions-1.7/cf-conventions.html](http://cfconventions.org/Data/cf-conventions/cf-conventions-1.7/cf-conventions.html)) gefolgt, die somit Bestandteil des [UC]<sup>2</sup>-Datenstandards sind. Nachfolgend werden diese Konventionen mit *CF-1.7* bezeichnet. Dies heißt, dass die Definitionen von Dimensionen, Koordinaten, Datenvariablen und Attributen *CF-1.7*-konform sein müssen. Da *CF-1.7* eine Erweiterung der Konventionen des *Cooperative Ocean/Atmosphere Research Data Service (COARDS)* ([ferret.pmel.noaa.gov/Ferret/documentation/coards-netcdf-conventions](http://ferret.pmel.noaa.gov/Ferret/documentation/coards-netcdf-conventions)) darstellt, sind diese gemäß der Darlegungen in *CF-1.7* implizit im [UC]<sup>2</sup>-Datenstandard berücksichtigt. Einige Erweiterungen des *CF-1.7*, die auch im [UC]<sup>2</sup>-Datenstandard genutzt werden, verhindern jedoch eine vollständige Rückwärtskompatibilität zu den *COARDS*-Konventionen. Um größtmögliche Rückwärtskompatibilität zu frei verfügbaren Tools zu gewährleisten, die auf den *COARDS*-Konventionen basieren, werden im [UC]<sup>2</sup>-Datenstandard daher nicht alle Optionen des *CF-1.7* unterstützt.

Für eine große Anzahl an Variablen sind in *CF-1.7* einheitliche Standardnamen und Einheiten definiert ([cfconventions.org/standard-names.html](http://cfconventions.org/standard-names.html)). Für alle weiteren Variablen werden analog dazu im [UC]<sup>2</sup>-Datenstandard Festlegungen getroffen, die sich u.a. auch an den Standardisierungen der BMBF-Fördermaßnahme „*Wolken und Niederschlag im Klimasystem*“ („*High Definition Clouds and Precipitation for Advancing Climate Predictions*“; HD(CP)<sup>2</sup>; [hdcp2.eu](http://hdcp2.eu)) orientieren; siehe hierzu: *Standardized Atmospheric Measurement Data (SAMD) product standard* ([icdc.cen.uni-hamburg.de/fileadmin/user\\_upload/samd\\_docs/samd\\_product\\_standard\\_v1.0.pdf](http://icdc.cen.uni-hamburg.de/fileadmin/user_upload/samd_docs/samd_product_standard_v1.0.pdf)).

Die nachfolgende Beschreibung des [UC]<sup>2</sup>-Datenstandards liefert weder eine vollständige Beschreibung des *NetCDF*-Formats noch der darauf aufbauenden Konventionen *COARDS* und *CF-1.7*, weshalb empfohlen wird, die o.a. Dokumente bei der Erstellung und Nutzung von [UC]<sup>2</sup>-Datensätzen heranzuziehen. Darüber hinaus wird empfohlen, die Dokumentation der verwendeten *NetCDF*-Programmierschnittstelle (*NetCDF* Application Programming Interface; *NetCDF* API) mit größter Sorgfalt zu beachten. Alle Beispiele sind in der Notation der *Common Data Language* (CDL) angeführt (siehe [www.unidata.ucar.edu/software/netcdf/docs/netcdf\\_utilities\\_guide.html](http://www.unidata.ucar.edu/software/netcdf/docs/netcdf_utilities_guide.html)). Bei Verwendung anderer *NetCDF* API ist insbesondere zu beachten, dass es unterschiedliche Konventionen für Zeichenketten (Strings) oder die Indizierung multidimensionaler Feldvariablen (Arrays) gibt, weshalb beim Erstellen einer *NetCDF*-Datei immer geprüft werden muss, ob die mit einer speziellen *NetCDF* API erzeugten Dateien auch tatsächlich dem [UC]<sup>2</sup>-Datenstandard entsprechen. Dabei gilt, dass in der CDL-Notation die Indizierung von Arrays mit dem Index 0 beginnt und der letzte (am weitesten rechts stehende) Index am schnellsten variiert (row-major ordering wie z.B. in der Programmiersprache C).

**Wichtig:** In *NetCDF*-Dateien werden Attribute, welche Text enthalten, anders behandelt als Variablen mit Text. Erstere werden in Form von Zeichenketten (Strings) mit *UTF-8* Kodierung abgespeichert, während letztere in *NetCDF-3* Dateien nur als Character-Arrays (Datentyp **char**) abgespeichert werden können. Ab *NetCDF-4* gibt es zwar auch den Datentyp **string** für Variablen, der aber im [UC]<sup>2</sup>-Datenstandard nicht unterstützt wird, um die Kompatibilität mit *PALM-4U* sicherzustellen. Es wird daher festgelegt, in Character-Arrays nur *ASCII*-Zeichen zu verwenden! Alle *NetCDF*-Dateien, die im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme erstellt werden, müssen im *NetCDF-4*-Format gespeichert werden.

Das *NetCDF*-Datenformat ermöglicht das gemeinsame Speichern von Daten und Metadaten in einer Datei. Daten werden entweder als skalare Größen oder in ein- oder mehrdimensionalen Arrays gespeichert. Arrays werden dabei über Dimensionen definiert. Metadaten werden in Form von zusätzlichen Variablen sowie von Attributen gespeichert. Jede *NetCDF*-Datei enthält globale Attribute, die den gesamten Datensatz betreffen. Diese werden in Abschnitt 2 behandelt. Abschnitt 3 ist den Dimensionen, Koordinaten- und Hilfskoordinatenvariablen gewidmet, welche eine eindeutige Zuordnung der Daten in Zeit und Raum herstellen. Im Abschnitt 4 werden die Datenvariablen und ihre Attribute, welche die Daten näher beschreiben, erläutert. Die im [UC]<sup>2</sup>-Datenstandard unterstützten **featureType**-Varianten werden im Abschnitt 5 beschrieben. Im Abschnitt 6 wird dargelegt, wie im DMS Dateinamen über die Inhalte einer *NetCDF*-Datei automatisch erzeugt werden. Der Anhang enthält einige Beispiele, die in der CDL-Notation die Erstellung von *NetCDF*-Dateien erläutern.

Im [UC]<sup>2</sup>-Datenstandard wird auf vier Tabellen A1 bis A4 verwiesen, die als separate Dateien vorliegen, da sie in kürzeren Abständen aktualisiert werden müssen, um neue Variablen ([Tabelle A1](#)), Variablengruppen ([Tabelle A2](#)), Institutionen ([Tabelle A3](#)) oder Standorte ([Tabelle A4](#)) hinzufügen zu können. Dabei wird festgelegt, dass einmal veröffentlichte Tabelleneinträge weder geändert noch gelöscht werden dürfen, damit auch bereits erzeugte *NetCDF*-Dateien gültig bleiben. Die Versionsnummerierung der Tabellen erfolgt daher unabhängig von der Versionsnummerierung des [UC]<sup>2</sup>-Datenstandards. Auf den Webseiten der [UC]<sup>2</sup>-Fördermaßnahme ([www.uc2-program.org](http://www.uc2-program.org)) werden die aktuellen Versionen des [UC]<sup>2</sup>-Datenstandards und der Tabellen bereitgestellt. Diese Tabellen sind nur für die [UC]<sup>2</sup>-Fördermaßnahme verbindlich; es wird aber eine analoge Vorgehensweise empfohlen, sollte der [UC]<sup>2</sup>-Datenstandard für andere Zwecke verwendet werden.

## 2 Globale Attribute

Im [UC]<sup>2</sup>-Datenstandard werden globale Attribute vereinbart, welche in den Tabellen 2.1, 2.2 und 2.3 aufgelistet sind, obligatorisch in jeder *NetCDF*-Datei vorhanden sein müssen und in den Abschnitten [2.1](#), [2.2](#) und [2.3](#) beschrieben werden. Das globale Attribut **featureType** spielt eine besondere Rolle für die Beschreibung der Datenstrukturen und wird in Abschnitt [2.4](#) erläutert. In Abschnitt [2.5](#) und Tabelle 2.4 werden zudem optionale globale Attribute mit besonderer Bedeutung beschrieben. Im Anhang sind zudem Beispiele für die Verwendung der globalen Attribute angeführt.

Weitere globale Attribute, die im [UC]<sup>2</sup>-Datenstandard nicht verbindlich vorgeschrieben sind, können hinzugefügt werden und bleiben nach dem Upload auf das DMS unverändert erhalten. Dabei wird verlangt, sich an *CF-1.7* oder darauf aufbauende Konventionen zu halten. Dies gilt insbesondere für die Groß- und Kleinschreibung von Attributen (z.B. **Conventions** statt **conventions**).

Mit Ausnahme von Eigennamen und Ortsbezeichnungen sollen alle Angaben in Attributen in Englisch erfolgen, um eine internationale Verwendung der Daten zu erleichtern.

### 2.1 Allgemeine globale Attribute

Die in Tabelle 2.1 aufgeführten globalen Attribute charakterisieren den in einer *NetCDF*-Datei gespeicherten Datensatz.

**Tabelle 2.1** Obligatorische allgemeine globale Attribute. Mit \* markiert sind Attribute, die für die Generierung des Dateinamens (siehe Abschnitt 6) von Bedeutung sind, während \*\* Attribute anzeigen, deren Werte beim Upload einer Konsistenzprüfung unterliegen.

Name	Bezeichnung	Typ	Wert
<b>title</b>	Kurztitel der Daten	S	Text
<b>data_content</b> <sup>*,**</sup>	Bezeichnung des Inhalts	S	Text (max. 16 Zeichen; siehe Tabellen <a href="#">A1</a> , <a href="#">A2</a> )
<b>source</b>	Art der Erhebung	S	Text (Trennung mehrerer Erhebungsarten durch Semikolon)
<b>version</b> <sup>*,**</sup>	Versionsnummer	I	Ganze Zahl (zwischen 1 und 999)
<b>Conventions</b> <sup>**</sup>	NetCDF-Konventionen	S	" <b>CF-1.7</b> "
<b>dependencies</b> <sup>**</sup>	Abhängigkeiten	S	Text (Trennung mehrerer Dateinamen durch Semikolon)
<b>history</b>	Informationen zur Datenprozessierung	S	Text
<b>institution</b> <sup>**</sup>	Institution des Urhebers bzw. Herausgebers	S	Text (siehe <a href="#">Tabelle A3</a> )
<b>acronym</b> <sup>*,**</sup>	Abkürzung der Institution	S	Text (max. 12 Zeichen; siehe <a href="#">Tabelle A3</a> )
<b>author</b> <sup>**</sup>	Autor/in	S	Nachname, Vorname, E-Mail-Adresse (Trennung mehrerer Autoren durch Semikolon)
<b>contact_person</b> <sup>**</sup>	Verantwortliche Person	S	Nachname, Vorname, E-Mail-Adresse (Trennung mehrerer Kontaktpersonen durch Semikolon)
<b>references</b>	Publikationen zu Daten oder Methoden	S	Text (Trennung mehrerer Referenzen durch Semikolon)
<b>comment</b>	Beliebige zusätzliche Informationen	S	Text
<b>keywords</b>	Schlüsselwörter für Suche im DMS	S	Text (Trennung mehrerer Keywords durch Semikolon)
<b>licence</b> <sup>**</sup>	Lizenzangaben	S	Text

**Typ:** I: Ganzzahl, S: UTF-8 String

Folgende der in Tabelle 2.1 aufgeführten globalen Attribute können individuell belegt werden und unterliegen keinen weiteren Vereinbarungen: **title**, **history**, **comment**, **keywords**. Falls keine Einträge vorgesehen sind, müssen Leerstrings verwendet werden. Nachfolgend werden die übrigen in Tabelle 2.1 aufgeführten globalen Attribute erläutert.

### **data\_content**

Dieses Attribut beschreibt den Inhalt der Datei in Kurzform. Dies kann entweder die Abkürzung für eine einzelne Größe oder eine Sammelbezeichnung (Variablengruppe; engl. category) für einen aus mehreren Größen bestehenden Datensatz sein. Für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, gilt, dass die Abkürzungen und Sammelbezeichnungen die IOP-Stadt Koordinator/innen in Abstimmung mit der [UC]<sup>2</sup>-Arbeitsgruppe "Datenmanagement" festlegen. Die aktuelle Liste der Abkürzungen für einzelne Größen ist in [Tabelle A1](#) zu finden, während die Sammelbezeichnungen für Variablengruppen in [Tabelle A2](#) aufgelistet sind. Für viele Größen, für die in CF-1.7 ein Wert für das Attribut **standard\_name** (siehe Abschnitt 4.2) existiert, gibt es auch eine

vom *Intergovernmental Panel on Climate Change (IPCC)* definierte Abkürzung des *Atmospheric Model Intercomparison Project (AMIP)*, die gemäß [Tabelle A1](#) als Wert für `data_content` verwendet wird, wenn einzelne Größen in einer *NetCDF*-Datei gespeichert werden (siehe auch [www.wcrp-climate.org/images/modelling/WGCM/publications/IPCC\\_standard\\_output.pdf](http://www.wcrp-climate.org/images/modelling/WGCM/publications/IPCC_standard_output.pdf)). Für alle anderen Größen wurde in [Tabelle A1](#) eine eigene Abkürzung eingeführt. Analog dazu wurden in [Tabelle A2](#) eigene Variablengruppen gebildet, wenn es keine geeignete Definition in *CF-1.7* gibt.

#### **source**

Hier wird die Datenerhebungsmethode spezifiziert. Wenn die Originaldaten das Ergebnis einer Computersimulation sind, muss hier der Modell-Name und die Modell-Version angegeben werden. Wenn die Daten mittels Observierungen erhoben wurden, wird die Art der Observierung angegeben (z.B. "AWS", "mobile", "bicycle", "UAS", "LIDAR", "RADAR"). Es wird vorgeschlagen, Großbuchstaben nur für Abkürzungen zu verwenden.

#### **version**

Die Versionsnummer muss in aufsteigender Reihenfolge, beginnend mit **1**, vergeben werden, um sicherzustellen, dass unterschiedlich aufbereitete Datensätze eindeutig identifizierbar sind. Höhere Nummern zeigen eine aktualisierte, verbesserte Version des jeweiligen Datensatzes an. Die jeweils höchste Versionsnummer im DMS stellt den aktuell gültigen Datensatz dar.

#### **Conventions**

Hier muss der Wert "CF-1.7" als *NetCDF*-Konvention angegeben werden.

#### **dependencies**

Sollten die Daten nicht von anderen Daten abhängen, wird hier ein leerer String angegeben. Andernfalls müssen die Dateinamen gemäß Abschnitt [6](#) der jeweiligen Datensätze angegeben werden. Mehrere Dateinamen werden durch Semikolon getrennt.

#### **institution**

Im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme legt die [UC]<sup>2</sup>-Arbeitsgruppe "Datenmanagement" die vollständigen Namen der Institutionen fest. Die aktuelle Liste ist in [Tabelle A3](#) zu finden.

#### **acronym**

Im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme legt die [UC]<sup>2</sup>-Arbeitsgruppe "Datenmanagement" die Abkürzungen für die jeweiligen Institutionen fest. Die aktuelle Liste ist in [Tabelle A3](#) zu finden.

#### **author**

Wenn eine oder mehrere Personen Autor/innen des Datensatzes sind, werden sie gemäß den Formatvorgaben in [Tabelle 2.1](#) hier angeführt, ansonsten ein leerer String angegeben. Die Angabe der E-Mail-Adresse ist optional.

#### **contact\_person**

Hier muss eine oder mehrere Personen als Ansprechpartner/innen gemäß den Formatvorgaben in [Tabelle 2.1](#) angeführt werden. Die Angabe der E-Mail-Adresse ist optional.

#### **references**

Hier kann eine oder mehrere Referenzen im bibliographischen Format der Meteorologischen Zeitschrift (siehe [www.schweizerbart.de/journals/metz/instructions](http://www.schweizerbart.de/journals/metz/instructions)) sowie optional die jeweilige DOI bzw. URL zum Abstract oder Volltext angegeben werden. Ansonsten wird ein leerer String angegeben.



## licence

Dieses Attribut enthält Informationen zu lizenzrechtlichen Bestimmungen der Datenverwendung. Sollte es keine diesbezüglichen Bestimmungen geben, kann ein leerer String angegeben werden. Für Daten, die im Rahmen der Fördermaßnahme [UC]<sup>2</sup> erhoben wurden, ist eine gemeinsame Data Policy vereinbart worden und steht unter [www.uc2-program.org/uc2\\_data\\_policy.pdf](http://www.uc2-program.org/uc2_data_policy.pdf) allgemein zur Verfügung. Die Data Policy definiert sieben unterschiedliche Lizenztypen: MOSAIK Licence, 3DO Licence; KliMoPrax Licence, UseUClim Licence, [UC]<sup>2</sup> Restricted Licence, [UC]<sup>2</sup> Research Licence, [UC]<sup>2</sup> Open Licence. Sollten die Daten unter einer der o.a. Lizenztypen lizenziert werden, muss die Angabe gemäß bzw. analog zu folgendem Beispiel gemacht werden:

```
"[UC]2 Open Licence; see [UC]2 data policy available at www.uc2-program.org/uc2_data_policy.pdf".
```

**Wichtig:** Obwohl *UTF-8* das Zeichen "2" (Exponent 2) unterstützt, wird dieses hier nicht verwendet, da nicht alle *NetCDF-API* eine vollständige Unterstützung für *UTF-8* Strings bieten und den Lizenzeintrag möglicherweise fehlerhaft darstellen!

## 2.2 Globale Attribute zum zeitlichen Bezug der Daten

Die in Tabelle 2.2 aufgeführten globalen Attribute dienen dazu, die Daten zeitlich einzuordnen. Dabei werden alle Zeiten grundsätzlich bezogen auf die koordinierte Weltzeit (*UTC*) angegeben.

**Tabelle 2.2** Obligatorische globale Attribute zum zeitlichen Bezug der Daten. Mit \* markiert sind Attribute, die für die Generierung des Dateinamens (siehe Abschnitt 6) von Bedeutung sind, während \*\* Attribute anzeigt, deren Werte beim Upload einer Konsistenzprüfung unterliegen.

Name	Bezeichnung	Typ	Wert
<b>campaign</b> ***	Mess-/Modellkampagne	S	Text (max. 12 Zeichen)
<b>origin_time</b> ***	Referenzzeitpunkt ( <i>UTC</i> )	S	"YYYY-MM-DD hh:mm:ss +00"
<b>creation_time</b> **	Zeitpunkt ( <i>UTC</i> ) der Dateierstellung	S	"YYYY-MM-DD hh:mm:ss +00"

**Typ:** S: *UTF-8* String

### campaign

Dieses Attribut stellt den zeitlichen Bezug der Daten zu Mess- oder Modellkampagnen her. Die Benennung der Messkampagnen in [UC]<sup>2</sup> erfolgt nach einem festgelegten Schlüssel. Daten aus Langzeitbeobachtungen (long-term observations; LTO) erhalten den Wert "**LTO**". Daten der Intensivmesskampagnen (intense observation periods; IOP) werden mit dem Wert "**IOP**" und einer Zahl mit zwei Ziffern bezeichnet, z.B. "**IOP01**". Daten der Sondermesskampagnen (special observation periods; SOP) werden mit dem Wert "**SOP**" und einer Zahl mit zwei Ziffern bezeichnet, z.B. "**SOP01**", analog zu den IOPs.

Datensätze aus Windkanalexperimenten oder Feldmessungen, welche im Rahmen der [UC]<sup>2</sup> Fördermaßnahme als Referenzdaten für die vereinbarte Modellvalidierung verwendet werden sollen, erhalten den Wert **VALRxx**, wobei **xx** eine Zahl mit zwei Ziffern darstellt, die von der Arbeitsgruppe "*Modellevaluierung*" festgelegt wird. Analog dazu werden die Ergebnisse der Modellsimulationen für die Validierungsläufe mit **VALMxx** bezeichnet. Werden bereits simulierte Validierungsläufe neu simuliert, erhalten diese Versionsnummern:

**VALMxxvy** (xx=01-99, Version="v", Versionsnummer y=1-9)

Für Windkanaldaten, die anderweitig genutzt werden sollen, wird empfohlen, diese mit dem Wert **WTE** zu kennzeichnen.

Sollten Datensätze keinen Bezug zu einer Kampagne haben, kann ein anderer String angegeben werden, wobei dieser nur die ASCII-Zeichen **A-Z**, **a-z**, "-", ".", "\_" sowie die Ziffern **0** bis **9** enthalten darf. Leerzeichen oder ein Leerstring sind nicht zulässig, da der Wert von **campaign** für die Generierung des Dateinamens (siehe Abschnitt 6) erforderlich ist. Für *PALM-4U*-Simulationen ohne speziellen Bezug zu einer Kampagne wird empfohlen, "**PALM-4U**" als Bezeichnung zu verwenden.

**Wichtig:** Bei der Generierung des Dateinamens (siehe Abschnitt 6) werden Minuszeichen "-", welche zur Trennung der Dateinamensbestandteile benötigt werden, durch Unterstriche "\_" ersetzt; der Attributwert selbst bleibt aber unverändert erhalten!

#### **origin\_time**

Hier wird der Referenzzeitpunkt in *UTC* für die Angabe der relativen Zeiten in den Daten (siehe Abschnitt 3.1) spezifiziert. Die Angabe der Zeitzone *UTC* durch "+00" ist verbindlich. Für Daten ohne expliziten Zeitbezug, z.B. bei digitalen Höhenmodellen oder digitalen Karten zu Landbedeckung oder anderen Eigenschaften von Oberflächen, soll hier der Zeitpunkt der Datenerhebung bzw. der Kartennachführung angegeben werden.

#### **creation\_time**

Analog zu **origin\_time** wird hier der Zeitpunkt der Dateierstellung in *UTC* spezifiziert. Die Angabe der Zeitzone *UTC* durch "+00" ist verbindlich.

### 2.3 Globale Attribute zum räumlichen Bezug der Daten

Die in Tabelle 2.3 aufgeführten globalen Attribute dienen dazu, die Daten räumlich einzuordnen.

**Tabelle 2.3** Obligatorische globale Attribute zum räumlichen Bezug der Daten. Mit \* markiert sind Attribute, die für die Generierung des Dateinamens (siehe Abschnitt 6) von Bedeutung sind, während \*\* Attribute anzeigt, deren Werte beim Upload einer Konsistenzprüfung unterliegen.

Name	Bezeichnung	Typ	Wert
<b>location**</b>	Ort, auf den sich die Daten beziehen	S	Name der Region, Stadt oder Kommune
<b>site**</b>	Messstandort bzw. Modelldomäne	S	Text (max. 12 Zeichen)
<b>origin_x</b>	x-Koordinate des Referenzpunktes	F	Rechtswert (= Ostwert) als <i>UTM</i> -Koordinate in Meter
<b>origin_y</b>	y-Koordinate des Referenzpunktes	F	Hochwert (= Nordwert) als <i>UTM</i> -Koordinate in Meter
<b>origin_lon</b>	Geographische Länge des Referenzpunktes	F	In Dezimalgrad Ost
<b>origin_lat</b>	Geographische Breite des Referenzpunktes	F	In Dezimalgrad Nord
<b>origin_z</b>	Referenzhöhe	F	In Meter
<b>rotation_angle</b>	Rotationswinkel des Koordinatensystems	F	In Grad (0.0 - 359.99)

**Typ:** F: Gleitkommazahl, S: *UTF-8* String

#### **location**

Dieses Attribut bezeichnet den Ort (Regionen, Städte oder Kommunen), auf welchen sich der Datensatz bezieht. Für deutsche Städte und Landkreise können hierzu auch amtliche KFZ-Kennzeichen (z.B. "B" für Berlin, "HH" für Hamburg, "S" für Stuttgart) verwendet werden. Für alle Daten, welche

im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, werden die Werte von der [UC]<sup>2</sup>-Arbeitsgruppe "Datenmanagement" festgelegt. Die aktuelle Liste ist in [Tabelle A4](#) zu finden.

#### **site**

Dieses Attribut bezeichnet das Gebiet, in welchem Observierungen stattgefunden haben, oder enthält den Namen einer Modell-Domäne in Kurzform. Ein Gebiet kann dabei ein einzelner Standort einer Messstation sein oder mehrere nahe beieinander gelegene Standorte umfassen. Im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme legen die IOP-Stadt Koordinator/innen in Abstimmung mit der [UC]<sup>2</sup>-Arbeitsgruppe "Datenmanagement" Bezeichnungen für die Gebiete fest. Die aktuelle Liste ist in [Tabelle A4](#) zu finden.

#### **origin\_x**

Dieses Attribut enthält die x-Koordinate (in Meter) des räumlichen Referenzpunktes, der den Ursprung des metrischen Koordinatensystems bildet (siehe Abschnitt [3.3](#)). Als metrisches Koordinatensystem wird im [UC]<sup>2</sup>-Datenstandard verbindlich das *Universale Transversale Mercatorsystem (UTM)* verwendet. Die Parameter des Koordinatenreferenzsystems (coordinate reference system; crs) für die UTM-Rechts- und -Hochwerte werden in der Koordinatenreferenzsystemvariablen (grid mapping variable) **crs** abgelegt (siehe Abschnitt [3.3](#)).

**Wichtig:** Für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, werden UTM-Rechts- und Hochwerte mit Bezug auf das Referenzsystem [ETRS89](#) mit dem [GRS80](#)-Ellipsoid angegeben!

#### **origin\_y**

Dieses Attribut enthält die y-Koordinate des räumlichen Referenzpunktes, der den Ursprung des metrischen Koordinatensystems bildet (siehe Beschreibung zu **origin\_x** sowie Abschnitt [3.3](#)).

#### **origin\_lon**

Hier wird die geographische Länge des räumlichen Referenzpunktes in Dezimalgrad östlicher Länge angegeben. Im [UC]<sup>2</sup>-Datenstandard ist festgelegt, dass Referenzsystem und Ellipsoid für die Angabe der geographischen Länge und Breite identisch sein müssen mit denen des UTM-Koordinatensystems (siehe Abschnitt [3.3](#)).

**Wichtig:** Für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, werden geographische Längen und Breiten mit Bezug auf das Referenzsystem [ETRS89](#) mit dem [GRS80](#)-Ellipsoid angegeben!

#### **origin\_lat**

Hier wird die geographische Breite des räumlichen Referenzpunktes in Dezimalgrad nördlicher Breite angegeben (siehe Beschreibung zu **origin\_lon** sowie Abschnitt [3.3](#)).

#### **origin\_z**

Hier wird die Höhe des räumlichen Referenzpunktes in Meter, bezogen auf ein vertikales Referenzsystem (vertical reference system; VRS), angegeben (siehe Abschnitt [3.2](#)). Im Falle von Beobachtungsdaten oder bei Verwendung einer **featureType**-Variante wird **origin\_z** einheitlich auf den Wert 0.0 gesetzt.

#### **rotation\_angle**

Dieses Attribut enthält den Rotationswinkel des metrischen Koordinatensystems in Grad. Der Winkel wird dabei von Nord ausgehend im Uhrzeigersinn abgebildet. Ist das Koordinatensystem nicht rotiert, d.h. wenn die y-Achse nach Norden zeigt, wird der Wert 0 angegeben. Eine Ausrichtung der y-Achse nach Osten, Süden bzw. Westen wird mit einem Rotationswinkel von 90, 180 bzw. 270 angezeigt.

## 2.4 Das globale Attribut **featureType**

Das *NetCDF*-Datenformat bietet die Möglichkeit, Daten in skalaren, ein- oder multidimensionalen Variablen zu speichern. Mit dem globalen Attribut **featureType** kann gemäß *CF-1.7* festgelegt werden, wie die Zeit- und Raumbezüge für unterschiedliche Arten der zeitlichen und räumlichen Organisation der Daten über Dimensionen, Koordinaten-, Hilfskoordinaten- und Datenvariablen strukturiert werden.

Liegen die Daten in einem regulären oder irregulären Gitter für ein metrisches Koordinatensystem vor, dann werden diese als multidimensionale Variablen gespeichert, deren Dimensionen durch **time**, **z**, **y** und **x** gegeben und durch entsprechende Koordinatenvariablen gleichen Namens spezifiziert sind.

**Wichtig:** Für multidimensionale Daten darf **featureType** nicht definiert sein!

Die Dimensionen müssen bei der Definition einer Variablen zwingend in folgender Reihenfolge aufgeführt werden: Zeit (**time**), Höhe über dem Referenzpunkt (**z**), Abstand vom Referenzpunkt in y-Richtung (**y**), Abstand vom Referenzpunkt in x-Richtung (**x**), d.h. die Daten sind so in der *NetCDF*-Datei gespeichert, dass der Index für die Zeit am langsamsten variiert. Nur durch diese Reihenfolge ist sichergestellt, dass die Daten auch *COARDS*-kompatibel und mit *PALM-4U* konform sind. Zur Verdeutlichung sei die Definition einer Datenvariable **ta** (**time**, **z**, **y**, **x**) angeführt, welche für jeden Zeitpunkt **time**[**m**] einer Zeitperiode sowie für jeden Punkt (**x**[**i**],**y**[**j**],**z**[**k**]) eines dreidimensionalen metrischen Gitters die Lufttemperatur enthält. Dabei ist der Wert der Dimension **time** die Anzahl der Zeitpunkte und die Werte der Dimensionen **z**, **y**, **x** die jeweilige Anzahl der Gitterpunkte. Für den Index **m** gilt, dass er Werte zwischen 0 und **time**-1 annehmen kann. Analog dazu können die Indizes **i**, **j** und **k** Werte von 0 bis **x**-1, 0 bis **y**-1 und 0 bis **z**-1 annehmen.

Sobald Datensätze nicht als multidimensionale Daten gespeichert sollen, muss das globale Attribut **featureType** gemäß *CF-1.7* angegeben werden.

**Wichtig:** In einer *NetCDF*-Datei können immer nur Daten mit gleichem **featureType** enthalten sein!

Da es die Möglichkeit gibt, Daten eines bestimmten **featureType** auch als multidimensionale Daten oder unter Verwendung einer anderen **featureType**-Variante zu speichern, werden im [UC]<sup>2</sup>-Datenstandard nicht alle in *CF-1.7* definierten **featureType**-Varianten unterstützt. Die zulässigen **featureType**-Varianten "**timeSeries**", "**timeSeriesProfile**" und "**trajectory**" werden in den Abschnitten [5.1](#), [5.2](#) und [5.3](#) genauer erläutert.

## 2.5 Optionale globale Attribute

Die in Tabelle 2.4 aufgeführten optionale globalen Attribute dienen der weiteren Beschreibung des in einer *NetCDF*-Datei gespeicherten Datensatzes.

**Tabelle 2.4** Optionale globale Attribute. Mit \* markiert sind Attribute, die für die Generierung des Dateinamens (siehe Abschnitt [6](#)) von Bedeutung sind, während \*\* Attribute anzeigt, deren Werte beim Upload einer Konsistenzprüfung unterliegen.

Name	Bezeichnung	Typ	Wert
<b>data_specifier</b> **	Spezifikation der Daten	S	Text (max. 16 Zeichen)

Typ: S: UTF-8 String

### **data\_specifier**

Dieses Attribut kann dafür verwendet werden, den Inhalt der gespeicherten Datensätze genauer zu spezifizieren. Dabei dürfen nur die Zeichen **A-Z**, **a-z**, die Ziffern **0** bis **9** und "**\_**" verwendet werden. Wenn dieses Attribut definiert wird, wird der Wert Bestandteil des automatisch generierten Dateinamens (siehe dazu Abschnitt [6](#)). Somit kann das Attribut dafür verwendet werden, die Dateinamen zweier Dateien, die ansonsten denselben Dateinamen hätten, unterscheidbar zu machen.

### 3 Dimensionen, Koordinatenvariablen und Hilfskoordinatenvariablen

Jede Datenvariable (siehe Abschnitt 4) wird in Zeit und Raum durch die Angabe von Zeitpunkten, von vertikalen Koordinaten (Höhen bzw. Tiefen) sowie von horizontalen Koordinaten verortet und ist somit über diese Angaben genau lokalisierbar. Als räumliche oder zeitliche Koordinatenvariable wird eine numerische, eindimensionale Variable bezeichnet, deren Name identisch mit ihrer Dimension ist und monoton auf- oder absteigende Werte zu Zeiten bzw. räumliche Koordinaten enthält.

**Wichtig:** Koordinatenvariablen dürfen keine Fehlwerte haben!

Als Hilfskoordinatenvariable werden Variablen bezeichnet, die Koordinaten enthalten, aber keine Koordinatenvariablen in o.a. Sinne sind (z.B. mehrdimensionale Arrays). Neben räumlichen oder zeitlichen Koordinaten- und Hilfskoordinatenvariablen gibt es weitere Hilfskoordinatenvariablen zur eindeutigen Zuordnung der Daten (z.B. für Stationsnamen oder Trajektorienamen; siehe Abschnitte 5.1 und 5.2). Im Gegensatz zu Koordinatenvariablen dürfen Hilfskoordinatenvariablen Fehlwerte haben, weshalb das Attribut `_FillValue` verpflichtend ist und den Wert `-9999` bzw. `-9999.0` haben muss.

**Wichtig:** Obwohl CF-1.7 es zulässt, verbietet der [UC]<sup>2</sup>-Datenstandard skalare Koordinaten- und Hilfskoordinatenvariablen, um die Kompatibilität mit den COARDS-Konventionen zu gewährleisten; d.h. die Werte müssen immer als Arrays eingegeben werden, auch wenn diese nur ein Element haben!

Je nach `featureType` unterscheiden sich dabei die Definitionen der Dimensionen, Koordinatenvariablen und Hilfskoordinatenvariablen gemäß CF-1.7 (siehe Beispiele im Anhang).

Bei multidimensionalen Daten mit einfachen Zeit- und Raumbezügen, bei denen das globale Attribut `featureType` nicht definiert werden darf (siehe Abschnitt 2.4), stellen die Dimensionen `time`, `z`, `y`, und `x` sowie die gleichnamigen eindimensionalen Koordinatenvariablen `time(time)`, `z(z)`, `y(y)`, und `x(x)` die Zeit- und Raumbezüge direkt her. Ansonsten werden je nach `featureType` andere Dimensionen definiert (siehe Abschnitte 5.1, 5.2 und 5.3) und die Variablen `time`, `z`, `y`, und `x` als ein- oder mehrdimensionale Koordinaten- bzw. Hilfskoordinatenvariablen der Dimensionen `ntime`, `nx`, `ny`, `nz`, `station` oder `traj` definiert.

In den nachfolgenden Abschnitten werden die entsprechenden Konventionen des [UC]<sup>2</sup>-Datenstandards in Bezug auf die Definition, Attribute und Werte der Koordinaten- und Hilfskoordinatenvariablen erläutert.

#### 3.1 Zeit

Alle Zeitangaben werden in der Koordinaten- bzw. Hilfskoordinatenvariable `time` relativ zu einem Referenzzeitpunkt in Sekunden angegeben (Abbildung 3.1). Die Attribute `long_name` und `standard_name` müssen den Wert "time" haben. Das Attribut `axis` wird auf "T" gesetzt.



**Abbildung 3.1** Definition von Zeiten in der Koordinaten- bzw. Hilfskoordinatenvariable `time` (rot markiert) im [UC]<sup>2</sup>-Datenstandard. Alle absoluten Zeitangaben (hier `origin_time`; grün markiert) beziehen sich auf den proleptischen Gregorianischen Kalender und die koordinierte Weltzeit (UTC).

In `NetCDF`-Dateien ist es zwar erlaubt, die Dimension `time=UNLIMITED` zu definieren, was das spätere Hinzufügen von neueren Daten ermöglicht. Da der Upload von Daten auf das DMS die Vollständigkeit der Daten voraussetzt, wird diese Option im [UC]<sup>2</sup>-Datenstandard aber nicht unterstützt.

Der Referenzzeitpunkt ist als absolute Zeitangabe im globalen Attribut **origin\_time** gespeichert (siehe Abschnitt [2.2](#)). Alle absoluten Zeitangaben beziehen sich dabei auf den proleptischen gregorianischen Kalender und die Zeitzone *UTC*.

Die Zeitintervalle zeitlich aggregierter Größen werden in der Variablen **time\_bounds** spezifiziert. Dabei gelten die in Abschnitt [3.6](#) beschriebenen Regeln für den bounds-Mechanismus. Weiterhin wird festgelegt, dass Zeitintervalle nach unten offen und nach oben abgeschlossen sind. Alle Zeitangaben in **time** sind grundsätzlich auf das Ende eines Zeitintervalls bezogen. Für grafische Darstellungen zeitlich aggregierter Werte kann es zwar sinnvoll sein, sich auf die Mitte eines Zeitintervalls zu beziehen, was aber vom jeweiligen Grafikprogramm eigenständig erledigt werden muss.

Die Aggregierungsmethoden der Daten werden mit dem Attribut **cell\_methods** spezifiziert. Dabei kann die Angabe der Aggregierungsmethode auch weggelassen werden, sofern es sich um punktuelle Werte ("**point**") handelt. Es wird empfohlen, die in *CF-1.7* erläuterten Angaben in **cell\_methods** umfassend zu nutzen. Bei zeitlich punktuellen Daten werden das Attribut **bounds** sowie die Variable **time\_bounds** weggelassen.

Sollten in einer Datei nur Werte für einen Zeitpunkt bzw. für ein Zeitintervall (z.B. Tagesmittelwerte) vorhanden sein, muss dennoch die Variable **time** als Array mit einem Element definiert werden.

Es ist prinzipiell möglich, die Variable **time** auch als **float** oder **double** zu definieren, um Bruchteile von Sekunden zu ermöglichen. Es wird jedoch empfohlen, diese Möglichkeit nur dann zu nutzen, wenn die Verwendung von ganzen Zahlen (**int** oder **long**) nicht ausreicht.

**Wichtig:** Zeitangaben von *PALM-4U*-Simulationen werden grundsätzlich als Gleitkommazahlen vom Datentyp **float** repräsentiert. Daten, die direkt in *PALM-4U* eingelesen werden sollen, müssen daher den Datentyp **float** für die Variable **time** verwenden!

Im [UC]<sup>2</sup>-Datenstandard werden weitere Konventionen für die Speicherung von Beobachtungsdaten definiert, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert wurden. Für Beobachtungsdaten der Intensivmesskampagnen (IOP) gilt, dass pro Datei nur die Daten eines Tages enthalten sein dürfen. Der Beginn des jeweiligen Tages wird in **origin\_time** als Referenzzeitpunkt festgelegt. Somit enthält die Variable **time** Werte bis maximal **86400**. Daten von Langzeitbeobachtungen (LTO) dürfen nur Werte eines Monats enthalten. Der Monatsbeginn wird in **origin\_time** als Referenzzeitpunkt festgelegt, und die Variable **time** enthält somit Werte bis maximal **31\*86400**. Der Zeitschritt für LTO-Daten, welche über das DMS bereitgestellt werden sollen, wird zudem auf mindestens 1800 Sekunden (30 Minuten) festgelegt.

**Wichtig:** Die o.a. Festlegungen gelten nicht für Datensätze, die für Modellvalidierungszwecke verwendet werden und sich jeweils auf den gesamten jeweiligen Validierungszeitraum erstrecken!

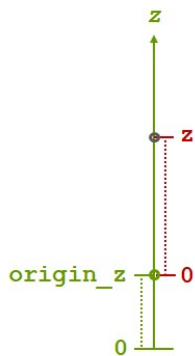
## 3.2 Vertikale Koordinaten

Angaben zu vertikalen Koordinaten (Höhen bzw. Tiefen) werden in der Koordinaten- bzw. Hilfskoordinatenvariable **z** als relative Werte in Bezug auf eine Referenzhöhe (**origin\_z**) in Meter aufgeführt (Abbildung 3.2). Das Attribut **long\_name** muss den Wert "**height above origin**" haben. Das Attribut **axis** wird auf "**Z**" gesetzt.

**Wichtig:** Das Attribut **standard\_name** muss den Wert "**height\_above\_mean\_sea\_level**" haben, wenn **origin\_z** den Wert 0 hat, und darf in allen anderen Fällen nicht definiert werden! Im Falle von Beobachtungsdaten oder bei Verwendung einer **featureType**-Variante wird die Referenzhöhe in **origin\_z** einheitlich auf den Wert 0 gesetzt, und somit muss **standard\_name** den Wert "**height\_above\_mean\_sea\_level**" haben!

Die Referenzhöhe **origin\_z** ist ein globales Attribut (siehe Abschnitt [2.3](#)) und wird in Meter, bezogen auf ein vertikales Referenzsystem, angegeben. Positive Werte für **z** sind Höhen oberhalb der

Referenzhöhe. Das Attribut **positive** muss daher immer auf den Wert "up" gesetzt werden. Negative Werte für **z** sind somit immer unterhalb der Referenzhöhe gelegen.



**Abbildung 3.2** Definition der vertikalen Koordinaten- bzw. Hilfskoordinatenvariable **z** (rot markiert) im [UC]<sup>2</sup>-Datenstandard. Absolute Höhenangaben (hier: **origin\_z**; grün markiert) beziehen sich im Falle von Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, auf das *Deutsche Höhenreferenzsystem DHHN2016*. Die vertikale Achse ist immer nach oben gerichtet.

Das vertikale Referenzsystem wird mit der Variable **vrs** spezifiziert. Dabei muss das in **vrs** spezifizierte vertikale Referenzsystem sich auf einen mittleren Meeresspiegel beziehen. Das Attribut **long\_name** hat den Wert "vertical reference system", und das Attribut **system\_name** den Wert "DHHN2016". Für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, wird als vertikales Referenzsystem das *Deutsche Höhenreferenzsystem DHHN2016* (siehe [Integrierter Raumbezug 2016](#)) verwendet und die Referenzhöhe **origin\_z** in Meter über Normalhöhen-Null angegeben.

**Wichtig:** Das Attribut **standard\_name** darf bei **vrs** nicht definiert werden!

Die Höhenintervalle vertikal aggregierter Größen werden in der Variablen **z\_bounds** spezifiziert. Dabei gelten die in Abschnitt 3.6 beschriebenen Regeln zum bounds-Mechanismus. Weiterhin wird festgelegt, dass Höhenintervalle nach unten offen und nach oben abgeschlossen sind. Alle Höhenangaben in **z** sind grundsätzlich auf die Mitte eines Höhenintervalls bezogen.

Die Aggregierungsmethoden der Daten werden mit dem Attribut **cell\_methods** spezifiziert. Dabei kann die Angabe der Aggregierungsmethode auch weggelassen werden, sofern es sich um punktuelle Werte ("point") handelt. Es wird empfohlen, die in CF-1.7 erläuterten Angaben in **cell\_methods** umfassend zu nutzen. Bei vertikal punktuellen Daten werden das Attribut **bounds** sowie die Variable **z\_bounds** weggelassen.

Sollten in einer Datei nur Werte für eine Höhe bzw. für ein Höhenintervall vorhanden sein, muss dennoch die Variable **z** als Array mit einem Element definiert werden.

Im Falle von Zeitreihendaten (**featureType="timeSeries"**; siehe Abschnitt 5.1) oder Zeitreihen von Vertikalprofilen (**featureType="timeSeriesProfile"**; siehe Abschnitt 5.2) muss die Stationshöhe (Höhe der Oberfläche) in der Variablen **station\_h** angegeben werden. Die Höhe über Grund bzw. die Tiefe unter Grund ergibt sich somit aus der Differenz von **z** und **station\_h**.

Bei Trajektorien (**featureType="trajectory"**) muss eine Variable namens **height** gemäß Abschnitt 5.3 verwendet werden, um für jeden Ort die Höhe über Grund (**height**  $\geq$  0) bzw. die Tiefe unter Grund (**height**  $<$  0) anzugeben.

Bei der Angabe von **station\_h** und **height** ist folgendes zu beachten: Bei Dateien des featureType **timeSeries** oder **timeSeriesProfile** gibt die Variable **station\_h** die Höhe der Oberfläche über **origin\_z** (**origin\_z** = 0) an. Die Oberfläche kann auch eine Dachfläche sein. Bei Messungen über Dach ist also **station\_h** = Geländehöhe + Dachhöhe, bei Messungen über unbebautem Boden

ist **station\_h** = Geländehöhe. Die Höhe der Messpunkte über **origin\_z** wird in der Variable **z** angegeben. Der Abstand der Messung von der Oberfläche ist also **z** minus **station\_h**. Bei einer Messung 5 m über dem Dach eines 10 m hohen Gebäudes auf einer Geländehöhe von 30 m ü. NN wäre **station\_h** = 40 m und **z** = 45 m.

Bei Dateien des **featureType trajectory** wird die Variable **height** benutzt, um die Höhe der Messung über der Oberfläche anzugeben. Bei einer Rucksackmessung könnte **height** also z.B. konstant bei 1.5 m liegen. Bei einer geflogenen Trajektorie könnte **height** z.B. bei 0 m starten, bis 200 m steigen und wieder bei 0 m enden. Die Variable **z** gibt wieder die Höhe der Messpunkte über **origin\_z** (**origin\_z** = 0) an. Die Differenz aus **z** und **height** ergibt die Höhe der Oberfläche. Wenn im Beispiel der geflogenen Trajektorie die Geländehöhe konstant bei 30 m ü. NN läge, würde **z** zwischen 30 und 230 m variieren.

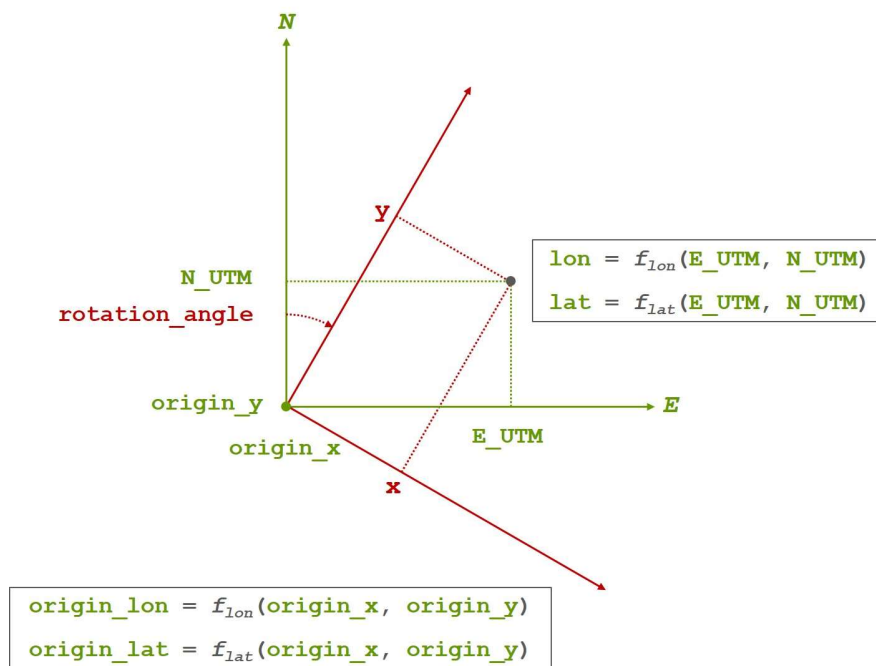
### 3.3 Horizontale Koordinaten

Jede *NetCDF*-Datei muss gemäß *CF-1.7* Koordinaten- bzw. Hilfskoordinatenvariablen enthalten, welche geographische Längen und Breiten der Daten spezifizieren. Hierzu werden die Variablen **lon** und **lat** verwendet (siehe Beispiele im [Anhang](#)). Im [UC]<sup>2</sup>-Datenstandard werden horizontale Koordinaten aber grundsätzlich über metrische Koordinaten **x** und **y** repräsentiert, um die Kompatibilität mit *PALM-4U* zu gewährleisten und Anwendern die räumliche Zuordnung der Daten zu erleichtern. Das Attribut **long\_name** muss den Wert "**distance to origin in x-direction**" bzw. "**distance to origin in y-direction**" haben.

<b>Wichtig:</b> Das Attribut <b>standard_name</b> darf bei <b>x</b> und <b>y</b> nicht angegeben werden!
--

*PALM-4U* verwendet ein horizontales, reguläres, metrisches Gitter, das aus Gründen der Optimierung der Modell-Performanz optional rotiert sein kann. Aus diesem Grund beziehen sich alle Modell- und Beobachtungsdaten auf ein nach Abbildung 3.3 definiertes horizontales, metrisches Koordinatensystem, dessen Ursprung in einem *UTM*-Koordinatensystem über die globalen Attribute **origin\_x** und **origin\_y** festgelegt ist. Der Rotationswinkel wird im globalen Attribut **rotation\_angle** in Grad spezifiziert; die Werte liegen zwischen 0 und 359.99. Ein nichtrotiertes, horizontales metrisches Koordinatensystem ist durch den Wert 0 in **rotation\_angle** gekennzeichnet und bedeutet eine Ausrichtung der x-Achse nach *UTM*-Gitter-Ost sowie eine Ausrichtung der y-Achse nach *UTM*-Gitter-Nord. Der Wert 90 bedeutet, dass die x-Achse nach *UTM*-Gitter-Süd und die y-Achse nach *UTM*-Gitter-Ost ausgerichtet sind.





**Abbildung 3.3** Definition eines rotierten, horizontalen, metrischen Koordinatensystems mit den Koordinaten  $\mathbf{x}$  und  $\mathbf{y}$  (rot markiert) im [UC]<sup>2</sup>-Datenstandard. Alle UTM-Koordinaten ( $\mathbf{E\_UTM}$ ,  $\mathbf{N\_UTM}$ ,  $\mathbf{origin\_x}$  und  $\mathbf{origin\_y}$ ; grün markiert) sowie die entsprechenden geographischen Längen und Breiten ( $\mathbf{lon}$  und  $\mathbf{lat}$ ,  $\mathbf{origin\_lon}$  und  $\mathbf{origin\_lat}$ ; grün markiert) beziehen sich bei Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert wurden, auf das Datum des *Europäischen Terrestrischen Referenzsystems 1989 (ETRS89)* mit dem Ellipsoid des *Geodätischen Referenzsystems 1980 (GRS80)*. Die Funktionen  $f_{lon}$  und  $f_{lat}$  zur Umrechnung der UTM-Koordinaten in geographische Längen und Breiten sind in frei verfügbaren Programmbibliotheken wie PROJ.4 enthalten.

**Wichtig:** Im Falle von Beobachtungsdaten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert wurden, wird **rotation\_angle** einheitlich auf den Wert 0 gesetzt!

Mittels der Gleichungen Gl. 3.1 und 3.2 können aus den Koordinaten  $\mathbf{x}$  und  $\mathbf{y}$  sowie der globalen Attribute  $\mathbf{origin\_x}$ ,  $\mathbf{origin\_y}$  und  $\mathbf{rotation\_angle}$  die UTM-Koordinaten als Rechtswerte ( $\mathbf{E\_UTM}$ ) und Hochwerte ( $\mathbf{N\_UTM}$ ) berechnet werden:

$$\mathbf{E\_UTM} = \mathbf{origin\_x} + \cos(\mathbf{rotation\_angle}) \cdot \mathbf{x} + \sin(\mathbf{rotation\_angle}) \cdot \mathbf{y} \quad (\text{Gl. 3.1})$$

$$\mathbf{N\_UTM} = \mathbf{origin\_y} - \sin(\mathbf{rotation\_angle}) \cdot \mathbf{x} + \cos(\mathbf{rotation\_angle}) \cdot \mathbf{y} \quad (\text{Gl. 3.2})$$

Der [UC]<sup>2</sup>-Datenstandard verlangt, die UTM-Koordinaten  $\mathbf{E\_UTM}$  und  $\mathbf{N\_UTM}$  als Hilfskoordinatenvariablen zu speichern, um Umrechnungen gemäß Gl. 3.1 und 3.2 und die damit verbundenen Unsicherheiten (Rundungsprobleme etc.) zu vermeiden und darüber hinaus weitere Informationen über die zugrundeliegende UTM-Kartenprojektion (siehe unten) abzuspeichern. Dabei müssen die Standardnamen gemäß den Beispielen im [Anhang](#) verwendet werden.

In einem nichtrotierten, horizontalen Koordinatensystem sind die Rechts- und Hochwerte eindimensionale Variablen  $\mathbf{E\_UTM}(\mathbf{x})$  und  $\mathbf{N\_UTM}(\mathbf{y})$ , sofern kein **featureType** verwendet wird. Bei rotierten, horizontalen Koordinatensystemen müssen diese als zweidimensionale Variablen  $\mathbf{E\_UTM}(\mathbf{y}, \mathbf{x})$  und  $\mathbf{N\_UTM}(\mathbf{y}, \mathbf{x})$  definiert werden.

Bei den **featureType**-Varianten sind auch  $\mathbf{x}$  und  $\mathbf{y}$  Hilfskoordinatenvariablen. Bei "**timeSeries**" und "**timeSeriesProfile**" sind die horizontalen Hilfskoordinatenvariablen eindimensional über die Dimension **station** ( $\mathbf{x}(\mathbf{station})$ ,  $\mathbf{y}(\mathbf{station})$ ,  $\mathbf{E\_UTM}(\mathbf{station})$  und  $\mathbf{N\_UTM}(\mathbf{station})$ ) definiert. Bei "**trajectory**" sind die Hilfskoordinatenvariablen

zweidimensional über die Dimensionen **traj** und **ntime** (**x(traj,ntime)**, **y(traj,ntime)**, **E\_UTM(traj,ntime)** und **N\_UTM(traj,ntime)**) definiert (siehe Beispiele im [Anhang](#)).

Gemäß *CF-1.7* müssen alle horizontalen Koordinaten auch als geographische Längen und Breiten angegeben werden. Die Umrechnung kann u.a. mit frei verfügbaren Tools erfolgen (z.B. [proj.org/apps/cs2cs.html](http://proj.org/apps/cs2cs.html) oder [trac.osgeo.org/osgeo4w/](http://trac.osgeo.org/osgeo4w/)). Diese Hilfskoordinatenvariablen müssen analog zu **E\_UTM** und **N\_UTM** als ein- oder zweidimensionale Variablen mit den Namen **lon** und **lat** (z.B. als **lon(y,x)** und **lat(y,x)** oder **lon(station)** und **lat(station)**) definiert und mit den umgerechneten geographischen Längen und Breiten belegt werden. Dabei müssen die Standardnamen gemäß den Beispielen im [Anhang](#) verwendet werden.

Die Parameter des Koordinatenreferenzsystems (coordinate reference system; **crs**) für die *UTM*-Rechts- und Hochwerte werden in der Koordinatenreferenzsystemvariablen (grid mapping variable) **crs** abgelegt. Für Deutschland gibt es drei *UTM*-Zonen (31, 32 und 33), welche sich auf die Zentralmeridiane 3, 9 und 15 Grad beziehen (*EPSG*-Projektionen 25831, 25832 und 25833; siehe z.B. [spatialreference.org/ref/epsg/25831/](http://spatialreference.org/ref/epsg/25831/)). Für Stadtregionen in anderen Ländern sind die entsprechenden *UTM*-Zonen auszuwählen.

**Wichtig:** Das Attribut **standard\_name** darf bei **crs** nicht angegeben werden!

Mittels der Attribute **coordinates** und **grid\_mapping** werden dann den Datenvariablen die entsprechenden absoluten Koordinaten und das zugrunde liegende Koordinatenreferenzsystem zugewiesen (siehe auch Abschnitt [4.1](#)). Somit kann jeder Modellgitterpunkt bzw. jeder Beobachtungsstandort sowohl relativ zum Ursprung des Koordinatensystems (definiert durch **origin\_x** und **origin\_y**) über die Koordinatenvariablen **x** und **y** als auch absolut über die Hilfskoordinatenvariablen für die geographischen Längen **lon** und Breiten **lat** sowie die *UTM*-Rechtswerte **E\_UTM** und Hochwerte **N\_UTM** eindeutig verortet werden.

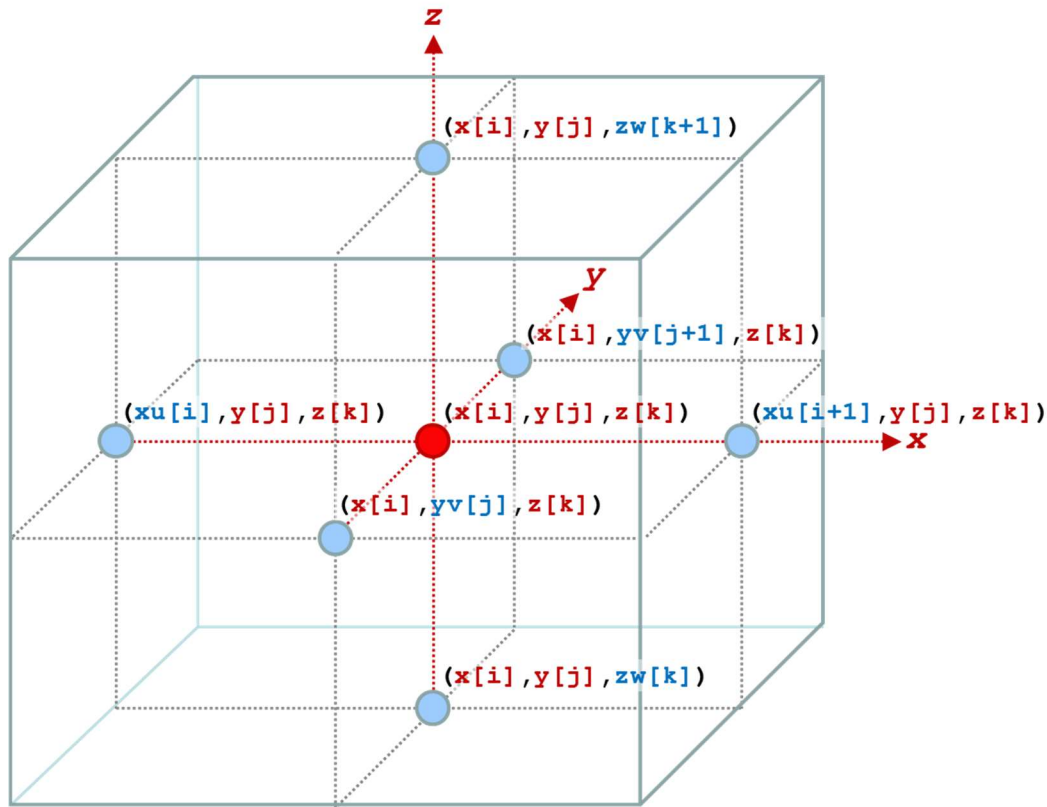
### 3.4 PALM-4U-Modellgitter

*PALM-4U* verwendet ein für numerische Modelle optimiertes, räumlich dreidimensionales, metrisches *Arakawa-C* Modellgitter (siehe Abbildung 3.4). Das Modellgitter mit den Dimensionen und Koordinatenvariablen **x**, **y** und **z** umfasst  **$n_x \cdot n_y \cdot n_z$**  Volumenelemente mit den Mittelpunkten (**x[i],y[j],z[k]**), an welchen die Werte der intensiven thermodynamischen Zustandsvariablen (potenzielle Temperatur etc.) und die Konzentrationen der Luftinhalstoffe (spezifische Feuchte, Konzentrationen von Luftschadstoffen etc.) vorliegen. An den Mittelpunkten der sechs das Volumenelement begrenzenden Oberflächen werden die Flussgrößen (Wind, sensible und latente Wärmeflussdichten etc.) gespeichert. Dazu werden die weiteren Dimensionen **xu**, **yv**, **zw** sowie die gleichnamigen Koordinatenvariablen **xu(xu)**, **yv(yv)**, **zw(zw)** benötigt, die jeweils um eine halbe Gitterweite in x-, y- und z-Richtung versetzt sind. Dabei sind die Dimensionen **xu**, **yv**, **zw** um jeweils ein Element größer als die entsprechenden Dimensionen **x**, **y**, **z**.

Analog zu **E\_UTM**, **N\_UTM**, **lon** und **lat** werden für die Koordinatenvariablen **xu** und **yv** weitere Hilfskoordinatenvariablen benötigt:

Blaue Punkte auf der x-Achse: **Eu\_UTM**, **Nu\_UTM**, **lonu** und **latu**.

Blaue Punkte auf der y-Achse: **Ev\_UTM**, **Nv\_UTM**, **lonv** und **latv**.



**Abbildung 3.4** Definition eines räumlich dreidimensionalen, metrischen Modellgitters für *PALM-4U* (Arakawa-C Grid) mit den Koordinatenvariablen  $\mathbf{x}$ ,  $\mathbf{y}$  und  $\mathbf{z}$  (rot markiert) und den jeweils um eine halbe Gitterweite versetzten zusätzlichen Koordinatenvariablen  $\mathbf{xu}$ ,  $\mathbf{yv}$  und  $\mathbf{zw}$  (blau markiert). Der rote Punkt liegt dabei im Zentrum eines Volumenelements, während die blauen Punkte die Mittelpunkte der sechs das Volumenelement begrenzenden Flächenelemente darstellen.

### 3.5 Oberflächen

Um Daten für Oberflächen (z.B. Wände und Dächer von Gebäuden) verarbeiten zu können, sind weitere Angaben erforderlich. Dabei können vier Fälle unterschieden werden, die jeweils spezifische Anforderungen an die zusätzlich erforderlichen Angaben stellen.

**Wichtig:** Sollten Daten einen expliziten Zeitbezug aufweisen, dann muss dieser in der Koordinatenvariablen **time** abgebildet werden. Die Zeitdimension muss dabei als erste Dimension der Datenvariablen verwendet werden.

#### Digitale Gelände- und Oberflächenmodelle

Zur quantitativen Beschreibung der Topographie von Oberflächen werden digitale Geländemodelle (DGM) bzw. Digitale Oberflächenmodelle (DOM) weitverbreitet eingesetzt. Es handelt sich hierbei um zweidimensionale Gitterdaten (Rasterdaten), bei denen die Geländehöhe bzw. die Höhe der Oberfläche als Datenvariablen in der generellen Form  $\mathbf{zt}(\mathbf{y}, \mathbf{x})$  bzw.  $\mathbf{zt}(\mathbf{time}, \mathbf{y}, \mathbf{x})$  vorliegen. Dabei gelten die Höhenwerte entweder nur für die Gitterpunkte (d.h. ohne Angabe von **cell\_methods** für die horizontalen Dimensionen) oder sind Mittelwerte (**cell\_methods="area: mean"**) von Flächenelementen (Pixel).

Bei reliefierter Topographie ist es sinnvoll, die Orientierung der Flächenelemente durch zwei Winkel anzugeben. Bei DGM werden hierzu meist die Exposition und die Hangneigung angegeben. Verallgemeinert kann die Orientierung einer Oberfläche durch den Azimut- und den Zenitwinkel der Flächennormalen spezifiziert werden. Hierfür sind die Datenvariablen **azimuth** und **zenith** vorgesehen, die im Falle von DGM bzw. DOM als zwei- bzw. dreidimensionale Felder **azimuth(y, x)**

bzw. `azimuth(time,y,x)` und `zenith(y,x)` bzw. `zenith(time,y,x)` definiert werden. Die Werte von `zenith` werden in Dezimalgrad angegeben und liegen zwischen 0 (Fläche ist nach oben orientiert) und 180 (Fläche ist nach unten orientiert). Die Datenvariable `azimuth` beschreibt die horizontale Ausrichtung der Flächennormalen in Dezimalgrad. Die Werte liegen zwischen 0 und 359, wobei analog zur Angabe von Windrichtungen die folgende Konvention gilt: Nord=0; Ost=90; Süd=180; West=270. Mit dieser Festlegung entspricht bei DGM somit der Winkel `azimuth` der Exposition und der Winkel `zenith` der Hangneigung. Bei einer nach oben oder unten orientierten Oberfläche ist der Wert von `azimuth` undefiniert und erhält den Pseudowert 0. Wenn die Variablen `zenith` und `azimuth` fehlen, wird von einheitlich nach oben ausgerichteten Oberflächen ausgegangen (sogenannte ‚Legotopographie‘).

### Gebäude im PALM-4U-Modellgitter

Beim PALM-4U-Modellgitter (Abschnitt 3.4) kann jedes Volumenelement mit dem Mittelpunkt  $(x[i],y[j],z[k])$  Teil eines Gebäudes sein. Da PALM-4U auch überhängende Gebäudeteile zulässt, kann somit jedes der sechs ein Volumenelement begrenzendes Flächenelement eine Grenzfläche zwischen Gebäude und Atmosphäre darstellen. Die Flächennormale einer Grenzfläche ist dabei stets nach außen, d.h. entweder in die Außen- oder Innenraumatmosphäre hinein orientiert.

Da in den meisten Fällen die Anzahl der tatsächlichen Gebäudeflächen wesentlich kleiner ist als die Anzahl der theoretisch möglichen Flächenelemente, ist es aus Speicherplatzgründen effizient, eine weitere Dimension `s` sowie die gleichnamige Koordinatenvariable `s` (surface) einzuführen, welche die Anzahl der Gebäudeflächen spezifiziert. Für jede der `s` Gebäudeflächen werden somit die Koordinaten `xs(s)`, `ys(s)` und `zs(s)`, die Hilfskoordinatenvariablen `Es_UTM(s)`, `Ns_UTM(s)`, `lons(s)` und `lats(s)` sowie die Winkel `zeniths(s)` und `azimuths(s)` benötigt. Dazu kommen noch weitere Datenvariablen, welche die Gebäudeflächen charakterisieren.

### Pixelbasierte Oberflächen

Speziell bei Thermal-IR-Bildern, die in Schrägsichtgeometrie aufgenommen wurden, liegen Daten als Flächenintegrale (Mittelwerte; `cell_methods="area: mean"`) für Bildelemente (Pixel) vor, welche in zweidimensionalen Variablen abgespeichert sind. Dazu müssen zwei zusätzliche Dimensionen `row` und `col` sowie die gleichnamigen Koordinatenvariablen eingeführt werden, welche die Bildgröße und die Indizes der Pixel spezifizieren. Sofern die Koordinaten  $(x[i,j],y[i,j],z[i,j])$  der Mittelpunkte der Pixel vorliegen, d.h. die Georeferenz bekannt ist, müssen die Daten zusammen mit den Koordinaten in einer NetCDF-Datei gespeichert werden.

**Wichtig:** In zukünftigen Versionen des [UC]<sup>2</sup>-Datenstandards müssen die Koordinaten `x`, `y`, und `z` angegeben werden!

Die Hilfskoordinatenvariablen `x(row,col)`, `y(row,col)`, `z(row,col)`, `lon(row,col)`, `lat(row,col)`, `E_UTM(row,col)` und `N_UTM(row,col)` sowie die Datenvariablen, z.B. `ts(row,col)`, sind somit zweidimensionale Felder.

**Wichtig:** Bei der Definition der Dimensionen `row` und `col` muss beachtet werden, dass der Index für `col` am schnellsten variiert und somit die Anzahl der Bildspalten (columns) spezifiziert!

Es wird empfohlen, zu jedem Pixel zusätzlich auch die Orientierung mittels der Datenvariablen `azimuth(row,col)` und `zenith(row,col)` anzugeben.

Sollten Bildzeitreihen zeitlich stationärer Oberflächen vorliegen, können diese in analoger Weise unter zusätzlicher Verwendung der Dimension `time` abgespeichert werden. Die Zeitdimension muss dabei wie immer als erste Dimension verwendet werden, z.B. `ts(time,row,col)`. Falls sich die Lage bzw. Orientierung der Oberflächen zeitlich ändert, werden alle räumlichen Hilfskoordinatenvariablen ebenfalls dreidimensionale Variablen (`x(time,row,col)` etc.).

Alternativ könnten pixelbasierte Oberflächendaten stationärer Oberflächen (sowohl Einzelbilder als auch Bildzeitreihen) unter Verwendung der **featureType**-Variante "**timeSeries**" abgespeichert werden (siehe Abschnitt 5.1). Da jedes Pixel aber in diesem Fall als separate Station angeführt werden muss, ist diese Alternative bei einer größeren Pixel-Anzahl ineffizient und wird daher nicht empfohlen.

### Beliebig im Raum orientierte Oberflächen

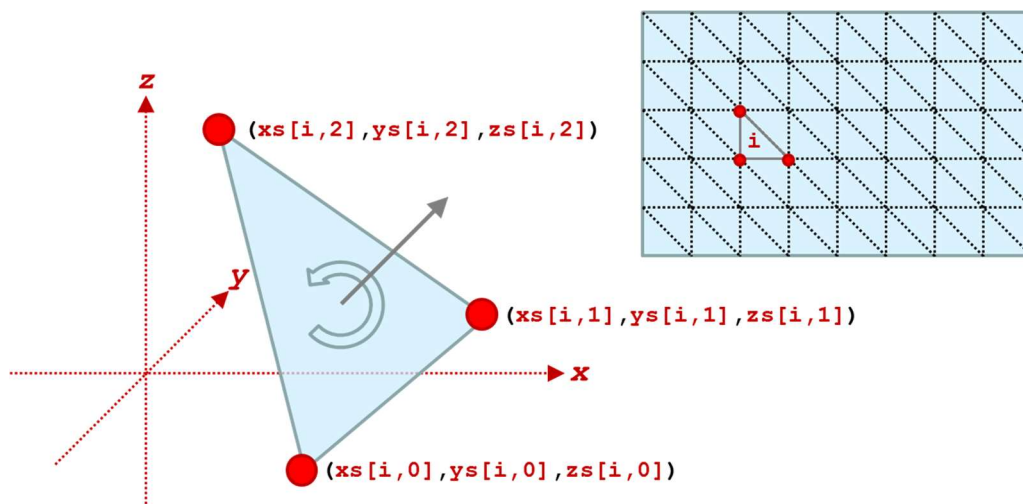
Bei den o.a. Fällen sind die Eckpunkte der orientierten Flächen durch zwei- bzw. dreidimensionale Gitter definiert. Bei gewissen Beobachtungsdaten, z.B. Thermalbildzeitreihen von beliebig orientierten Gebäudeflächen und Vegetationsbeständen, oder bei realitätsnahen 3D-Gebäudemodellen aus GIS-Daten ist es aber erforderlich, die Orientierung von Oberflächen in allgemeiner Form zu beschreiben und dadurch komplexe, realitätsnahe Repräsentationen von Flächen zu ermöglichen.

Wie im Falle von Gebäuden im *PALM-4U*-Modellgitter werden eine weitere Dimension **s** sowie die gleichnamige Koordinatenvariable benötigt, welche die Anzahl der Oberflächenelemente spezifiziert.

Komplexe Objekte (z.B. Gebäude mit Schrägdächern, die derzeit nicht in einem *PALM-4U*-Modellgitter dargestellt werden können), werden in elementare, miteinander verbundene Dreiecke zerlegt (Abbildung 3.5). Dabei werden die Volumina durch ihre begrenzenden Oberflächen repräsentiert. Jede dieser Oberflächen wird, je nach Anzahl der Eckpunkte, in eine Vielzahl von Dreiecken (Abbildung 3.5) zerlegt, deren Anzahl und Größe so gewählt werden kann, dass die Oberflächen mit hinreichender räumlicher Differenzierung dargestellt werden können.

Ein Dreieck ist durch die Angabe von drei Punkten sowohl bzgl. seiner Lage und Ausdehnung als auch seiner Orientierung eindeutig definiert. Die zusätzliche Dimension **npt** hat den festen Wert 3.

Die Reihenfolge der drei Punkte legt die Orientierung des Dreiecks fest (siehe Abbildung 3.5). Flächennormalen sind dabei in die Atmosphäre gerichtet, weshalb **azimuth** und **zenith** der Flächennormalen nicht erforderlich sind (zur Vereinfachung der Datennutzung können diese aber dennoch angegeben werden). Für jede der **s** Oberflächen werden somit die Hilfskoordinatenvariablen **xs(s,npt)**, **ys(s,npt)** und **zs(s,npt)** sowie **Es\_UTM(s,npt)**, **Ns\_UTM(s,npt)**, **lons(s,npt)** und **lats(s,npt)** benötigt. Dazu kommen pro Flächenelement noch weitere Datenvariablen (z.B. bauphysikalische Größen, Oberflächentemperatur etc.), welche die Gebäudeflächen charakterisieren bzw. gemessene oder simulierte Werte sind.



**Abbildung 3.5** Definition eines orientierten Dreiecks (Indexwert **i**) durch drei Punkte mit den Hilfskoordinatenvariablen **xs**, **ys** und **zs**, deren Reihenfolge des zweiten Index (Indexwerte 0 bis 2) die Orientierung der Flächennormalen in einem rechtshändigen System definieren. Dabei gilt, dass die Flächennormale eines Dreiecks in die Atmosphäre gerichtet ist (Außenwände: Außenraumatmosfera; Innenwände: Innenraumatmosfera). Komplexe bzw. ausgedehnte

Oberflächen können durch Zerlegung in eine Vielzahl von Dreiecken mit hinreichender räumlicher Differenzierung dargestellt werden.

### 3.6 Intervallgrenzen von Koordinaten- und Hilfskoordinatenvariablen

Im Falle von Koordinaten- und Hilfskoordinatenvariablen, die sich auf Intervalle beziehen, werden diese Intervalle gemäß *CF-1.7* mittels des `bounds`-Mechanismus angegeben. Dazu erhält die Koordinaten- oder Hilfskoordinatenvariable das Attribut `bounds`, welches den Namen der Variable enthält, die die Intervallgrenzen beschreibt. Diese erhält den Namen `xyz_bounds`, wobei `xyz` der Name der entsprechenden Koordinaten- oder Hilfskoordinatenvariable ist. Die Variable `xyz_bounds` hat zusätzlich zu den Dimensionen der Variable `xyz` eine weitere Dimension `nv` mit dem Wert 2, welche die am schnellsten variierende Dimension sein muss, also ganz rechts steht. `xyz_bounds` muss in demselben Datentyp wie `xyz` definiert sein. Die Variable `xyz_bounds` darf keine Attribute besitzen, da sie automatisch die Werte der Attribute von `xyz` (z.B. `units` oder `_FillValue`) übernimmt.

## 4 Datenvariablen

Als Datenvariable werden im [UC]<sup>2</sup>-Datenstandard Variablen bezeichnet, die keine Koordinaten-, Hilfskoordinatenvariablen oder Hilfsvariablen (siehe Abschnitt 4.3) sind und keine Referenzsysteme (**vrs**, **crs**) beschreiben. Eine Datenvariable enthält somit Daten und variablenspezifische Metadaten von Größen, die durch Beobachtungen oder Modellsimulationen gewonnen wurden.

Im [UC]<sup>2</sup>-Datenstandard werden verbindliche Festlegungen für die Bezeichnungen von Datenvariablen getroffen, wie dies auch im Datenstandard des SAMD der BMBF-Fördermaßnahme HD(CP)<sup>2</sup> erfolgt. Da PALM-4U feste Variablenbezeichnungen für Input- und Output-Variablen verwendet, sind diese zu verwenden, damit eine automatisierte Verarbeitung von Mess- und Modelldaten möglich ist. Darüber hinaus wird festgelegt, Datenvariablen wie folgt zu benennen:

a) **<variable>**

b) **<variable>\_<method>**

Dabei ist **<variable>** der entsprechende Wert für die jeweilige Größe gemäß [Tabelle A1](#). Mit **<method>** wird die zeitliche Aggregierungsmethode gemäß CF-1.7 spezifiziert. Dabei muss der Wert von **<method>** eine Abkürzung (siehe Tabelle 4.1) einer der zulässigen Methoden des Attributs **cell\_methods** sein. Eine Ausnahme davon bildet der Wert **skew** (skewness; drittes Moment einer Verteilung), für den es in CF-1.7 derzeit keine Aggregierungsmethode gibt.

**Tabelle 4.1** Zulässige Bezeichnungen (**<method>**) für zeitliche Aggregierungsmethoden (**cell\_methods**) gemäß CF-1.7 zur Benennung von Datenvariablen.

<b>&lt;method&gt;</b>	<b>cell_methods gemäß CF-1.7</b>	<b>Angabe der Einheit in units</b>
max	"maximum"	Größe (z.B. "m s-1")
maxabs	"maximum_absolute_value"	Größe (z.B. "m s-1")
med	"median"	Größe (z.B. "m s-1")
mid	"mid_range"	Größe (z.B. "m s-1")
min	"minimum"	Größe (z.B. "m s-1")
minabs	"minimum_absolute_value"	Größe (z.B. "m s-1")
meanabs	"mean_absolute_value"	Größe (z.B. "m s-1")
meanupd	"mean_of_upper_decile"	Größe (z.B. "m s-1")
mode	"mode"	Größe (z.B. "m s-1")
range	"range"	Größe (z.B. "m s-1")
sigma	"standard_deviation"	Größe (z.B. "m s-1")
sumsqr	"sum_of_squares"	Quadrat der Größe (z.B. "m <sup>2</sup> s-2")
var	"variance"	Quadrat der Größe (z.B. "m <sup>2</sup> s-2")
skew		dimensionslos ("1")

Variante a) wird eingesetzt, falls eine Variable Werte enthält, die instantan (**cell\_methods: "point"**), zeitlich gemittelt (**cell\_methods: "mean"**) oder zeitlich akkumuliert (**cell\_methods: "sum"**) sind. Ansonsten wird Variante b) verwendet. Somit können mehrere mit unterschiedlichen Methoden zeitlich aggregierte Größen in einer *NetCDF*-Datei gespeichert werden.

**Wichtig:** Enthält eine *NetCDF*-Datei nur eine einzige Datenvariable, dann muss die Bezeichnung der Variablen gemäß [Tabelle A1](#) auch als Wert für das globale Attribut `data_content` verwendet werden, d.h. ohne `<method>`!

Variablenspezifische Metadaten werden durch Variablenattribute angegeben. Im [UC]<sup>2</sup>-Datenstandard wird zwischen obligatorischen und optionalen Attributen für Datenvariablen unterschieden. Die obligatorischen Attribute werden beim Upload auf das DMS in Bezug auf ihre Konsistenz überprüft.

## 4.1 Obligatorische Attribute

In Tabelle 4.2 sind die obligatorischen Attribute für Datenvariablen aufgelistet.

**Tabelle 4.2** Obligatorische Attribute für Datenvariablen.

Name	Erläuterung	Typ	Wert
<code>long_name</code>	Langname der Größe	S	Gemäß <a href="#">Tabelle A1</a>
<code>units</code>	Einheit der Größe	S	Text; <i>UDUNITS</i> -konforme Einheit
<code>_FillValue</code>	Füllwert bei fehlenden oder ungültigen Daten	N	-9999 bzw. -9999.0
<code>coordinates</code>	Alle Koordinaten- und Hilfskoordinatenvariablen	S	Z.B. "lon lat E_UTM N_UTM x y z time"
<code>grid_mapping</code>	Koordinatenreferenz-systemvariable	S	"crs"

**Typ:** N: numerisch (Ganz- oder Gleitkommazahl), S: *UTF-8* String

### `long_name`

Der Langname dient der näheren Erläuterung einer Variablen und wird von einigen Programmen für automatisch generierte Abbildungen verwendet. Im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme wird jeder Größe durch die AG "Datenmanagement" der Wert für `long_name` fest zugeordnet. Die aktuelle Liste der definierten Variablen und die entsprechenden Werte für `long_name` befinden sich in [Tabelle A1](#).

### `units`

Gemäß *CF-1.7* können für Größen unterschiedliche Einheiten verwendet werden, sofern diese *UDUNITS*-konform sind und somit ineinander umgerechnet werden können (siehe [cfconventions.org/standard-names.html](http://cfconventions.org/standard-names.html)). In [Tabelle A1](#) gibt es für jede im [UC]<sup>2</sup>-Datenstandard definierte Variable einen vorgeschlagenen Wert für `units`. Optional kann zusätzlich eine alternative Einheit im optionalen Attribut `units_alt` angegeben werden (siehe Abschnitt [4.2](#)).

### `_FillValue`

Fehlwerte oder ungültige Werte (z.B. Werte außerhalb eines gültigen Wertebereichs) werden mit dem angegebenen „Füllwert“ belegt, der einheitlich auf den Wert -9999 bzw. -9999.0 festgesetzt wird. Das Attribut `_FillValue` muss denselben Datentyp wie die zugehörige Datenvariable haben!

### `coordinates`

Dieses Attribut enthält die Namen aller Koordinaten- und Hilfskoordinatenvariablen, auf die sich die Datenvariable bezieht. Je nachdem, ob ein *PALM-4U*-Modellgitter verwendet wird, ob orientierte Oberflächen vorliegen, oder ob eine `featureType`-Variante verwendet wird, unterscheidet sich der Wert dieses Attributes. Die Reihenfolge oder die Anzahl an Leerzeichen spielen keine Rolle.



Nachfolgend sind einige Beispiele angeführt:

```
"lon lat E_UTM N_UTM x y z time"
"lonu latu Eu_UTM Nu_UTM xu y z time"
"lonv latv Ev_UTM Nv_UTM x yv z time"
"lon lat E_UTM N_UTM x y zw time"
"lons lats Es_UTM Ns_UTM xs ys zs time"
"lon lat E_UTM N_UTM x y z time station_name"
"lon lat E_UTM N_UTM x y z time traj_name"
"lon lat E_UTM N_UTM x y z time traj_name bands_pm".
```

### grid\_mapping

Dieses Attribut enthält die Zuordnung der horizontalen Hilfskoordinatenvariablen zur Koordinatenreferenzsystemvariablen **crs** (siehe Beispiele im [Anhang](#)).

## 4.2 Optionale Attribute

In *CF-1.7* ist eine Vielzahl von weiteren Attributen aufgeführt, welche die Eigenschaften von Variablen näher beschreiben. Diese können optional verwendet werden bzw. sind erforderlich, wenn weitere Eigenschaften der Daten beschrieben werden müssen. Letzteres gilt insbesondere, wenn es sich um zeitlich oder räumlich aggregierte Daten handelt, bei denen die Aggregierungsmethode gemäß *CF-1.7* mit dem Attribut **cell\_methods** spezifiziert werden muss, da ansonsten Grafik- oder Analyseprogramme möglicherweise annehmen, dass die Daten zeitlich oder räumlich punktuell gültig seien. Eine Ausnahme davon gilt für zeitliche Aggregierungsmethoden (z.B. **skew**), für die es derzeit keine in *CF-1.7* definierten Werte für **cell\_methods** gibt.

Tabelle 4.3 führt die im [UC]<sup>2</sup>-Datenstandard zusätzlich festgelegten optionalen Attribute für Variablen auf. Es wird dringend empfohlen, alle der in Tabelle 4.3 angeführten optionalen Attribute zu verwenden, wenn dies sinnvoll und anwendbar ist. Insbesondere ist die Angabe zu den Messinstrumenten für alle Beobachtungsdaten erforderlich. Darüber hinaus können weitere optionale Attribute unter Beachtung des *CF-1.7* verwendet werden.

**Tabelle 4.3** Optionale Attribute für Datenvariablen. Mit \* markiert ist das Attribut **standard\_name**, das nur dann definiert werden darf, wenn es für die Variable einen *CF-1.7* Standardnamen gibt.

Name	Erläuterung	Typ	Wert
<b>standard_name*</b>	Standardname der Größe	S	Gemäß <a href="#">Tabelle A1</a> (das Attribut darf nicht definiert werden, wenn es keinen <i>CF-1.7</i> -konformen Standardnamen gibt!)
<b>units_alt</b>	Alternative Einheit	S	Text
<b>uncertainty_rel</b>	Relative Unsicherheit	F	Angabe in Prozent des jeweiligen Wertes
<b>uncertainty_abs</b>	Absolute Unsicherheit	N	Angabe in der Einheit der Größe
<b>processing_level</b>	Prozessierungslevel	I	Ganzzahl zwischen 0 und 3
<b>processing_info</b>	Prozessierungsmethode	S	Text
<b>instrument_name</b>	Bezeichnung des Messinstruments	S	Text
<b>instrument_nr</b>	Interne Nummer des Messinstruments	S	Text: Ziffern und Buchstaben
<b>instrument_sn</b>	Seriennummer des Messinstruments	S	Text: Ziffern und Buchstaben

**Typ:** I: Ganzzahl, N: numerisch (Ganz- oder Gleitkommazahl), S: *UTF-8* String

### **standard\_name**

Dieses Attribut ist in *CF-1.7* für eine Vielzahl von Größen festgelegt (siehe [cfconventions.org/standard-names.html](http://cfconventions.org/standard-names.html)) und muss, wenn dort vorhanden, verwendet werden. Ansonsten darf das Attribut nicht definiert werden. Die aktuelle Liste der definierten Variablen und die entsprechenden Werte für **standard\_name** befinden sich in [Tabelle A1](#).

### **units\_alt**

Mit diesem Attribut kann eine alternative Einheit in Ergänzung zur *UDUNITS*-konformen Einheit (siehe obligatorisches Attribut **units** in Abschnitt [4.1](#)) angegeben werden. Dabei ist zu beachten, dass die Einheiten in **units** und **units\_alt** sich auf identische Zahlenwerte der Daten beziehen müssen. Als Beispiel sei angeführt, dass die Einheit "**s<sup>-1</sup>**" alternativ auch als "**Hz**" angegeben werden kann.

### **uncertainty\_rel**

Mit diesem Attribut wird die Unsicherheit der Werte in relativer Form (in Prozent) angegeben. Ein Wert von **20.0** (= 20 %) bedeutet beispielsweise, dass die Unsicherheit mindestens ein Fünftel des jeweiligen Wertes beträgt. Wird das Attribut weggelassen, wird **uncertainty\_rel** auf den Wert **0.0** gesetzt und die Unsicherheit durch den Wert von **uncertainty\_abs** festgelegt. Ist die über **uncertainty\_rel** berechnete Unsicherheit kleiner als der Wert in **uncertainty\_abs**, wird letzterer Wert zur Bestimmung der Unsicherheit verwendet. Fehlen sowohl **uncertainty\_rel** als auch **uncertainty\_abs**, dann werden alle Werte als exakt betrachtet.

### **uncertainty\_abs**

Mit diesem Attribut wird die Unsicherheit der Werte in absoluter Form angegeben. Somit ist die absolute Unsicherheit unabhängig vom jeweiligen Wert konstant. Wird das Attribut weggelassen, wird **uncertainty\_abs** auf den Wert **0** (bzw. **0.0**) gesetzt und die Unsicherheit durch den Wert im Attribut **uncertainty\_rel** festgelegt. Ist die über **uncertainty\_rel** berechnete Unsicherheit kleiner als der Wert in **uncertainty\_abs**, wird letzterer Wert zur Bestimmung der Unsicherheit verwendet. Fehlen sowohl **uncertainty\_rel** als auch **uncertainty\_abs**, dann werden alle Werte als exakt betrachtet.

<b>Wichtig:</b> Dieses Attribut muss denselben Datentyp wie die Variable haben!
---

### **processing\_level**

Dieses Attribut enthält einen Zahlencode, der folgende Bedeutung hat: **0**: Rohdaten; **1**: kalibrierte Daten; **2**: automatisch gefilterte bzw. weiterverarbeitete Daten; **3**: manuell gefilterte bzw. weiterverarbeitete Daten. Für Observierungsdaten, die über das DMS bereitgestellt werden sollen, gilt, dass diese qualitätsgeprüft und somit mindestens kalibriert sein sollen. Sinnvollerweise sollten aber alle Daten auch mit automatischen Verfahren gefiltert bzw. weiterverarbeitet sein, z.B. durch Prüfung des gültigen Wertebereichs einer Größe und Markieren aller Werte außerhalb dieses Bereichs mit dem in **\_FillValue** spezifizierten Wert.

### **processing\_info**

Mit diesem Attribut können ergänzende Angaben zur Datenprozessierung gemacht werden. Beispielsweise können Informationen zur Datenfilterung oder anderen Methoden der Qualitätssicherung als Freitext hinterlegt werden.

### **instrument\_name**

Da jede Messgröße mit unterschiedlichen Instrumenten erhoben werden kann, erfolgt hier die Angabe des Namens oder der Bezeichnung des Messinstruments (bzw. des Sensors). Für Modellsimulationen entfällt dieses Attribut.

### **instrument\_number**

Mehrere Messinstrumente gleichen Typs können z.B. beliebig nummeriert oder mit einer Kombination aus Ziffern und Buchstaben bezeichnet werden, um sie zu unterscheiden. Dabei darf die Nummerierung aus Ziffern, Buchstaben oder Kombinationen davon bestehen. Für Modellsimulationen entfällt dieses Attribut.

### **instrument\_sn**

Zur Identifikation des Messinstruments ist die Angabe der Seriennummer als Ziffern, Buchstaben oder Kombinationen davon erforderlich. Für Modellsimulationen entfällt dieses Attribut.

## 4.3 Hilfsvariablen und Flags

In *CF-1.7* gibt es die Möglichkeit, Hilfsvariablen mit zusätzlichen Daten (ancillary data) für eine Datenvariable zu definieren und diese mit dieser zu verknüpfen. Dies ist beispielsweise dann sinnvoll, wenn jeder Wert der Datenvariablen eigene zusätzliche Daten zugewiesen bekommen soll. Pro Datenvariable können eine oder mehrere Hilfsvariablen definiert werden. Umgekehrt kann eine Hilfsvariable mehreren Datenvariablen zugeordnet werden.

Hilfsvariablen müssen die gleichen Dimensionen wie die Datenvariable haben, können aber unterschiedliche Datentypen besitzen. Die Verknüpfung der Datenvariablen mit den Hilfsvariablen erfolgt über das Attribut **ancillary\_variables**, in welchem die Datenvariable die Namen aller ihr zugeordneten Hilfsvariablen auflistet. Mehrere Hilfsvariablen werden durch Leerzeichen getrennt.

Ein spezieller Fall von Hilfsvariablen sind sogenannte Flags. Hilfsvariablen, die Flags enthalten, müssen gemäß *CF-1.7* das Attribut **flag\_meanings** sowie eines der beiden Attribute **flag\_values** und **flag\_masks** verwenden. Obwohl *CF-1.7* auch die gleichzeitige Verwendung von **flag\_values** und **flag\_masks** erlaubt, wird dies nicht empfohlen, da alternativ auch zwei Flag-Variablen genutzt werden können. Flags können auch dazu genutzt werden, bei Fehlern in der Datenvariable den Grund für das Fehlen von Daten anzugeben. Dazu müssen entsprechende **flag\_meanings** angelegt werden. Bei unbekanntem Grund kann beispielsweise auch „NA“ in den **flag\_meanings** enthalten sein. Daher sollen Flags auch nicht selbst das Attribut **\_FillValue** besitzen.

Für Hilfsvariablen legt der [UC]<sup>2</sup>-Datenstandard eine Namenskonvention fest. Einerseits kann jede Datenvariable auch als Hilfsvariable für andere Datenvariablen fungieren. Andererseits können zusätzliche Hilfsvariablen eingeführt werden. Diese werden wie folgt benannt: **ancillary\_xyz**. Dabei kann der Wert für **xyz** frei gewählt werden. Im Anhang [A5](#) ist ein Beispiel für Daten mit Hilfsvariablen und Flags angeführt.

## 4.4 Spektrale Daten

Spektrale Daten sind Daten, für die zu jedem Zeitpunkt und an jedem Ort mehrere Werte für eine Variable vorliegen. Ein häufig vorkommendes Beispiel für spektrale Daten sind Satellitendaten, in denen für jedes Pixel Strahldichten oder Reflektanzen für unterschiedliche Wellenlängenbereiche (Spektralkanäle bzw. spektrale Bänder) vorliegen. Ein weiteres Beispiel sind gleichzeitige Messungen von Anzahl- oder Massenkonzentrationen von Aerosolpartikeln für unterschiedliche Größenklassen mit Aerosolspektrometern.

Anstatt die Werte von spektralen Daten in separaten Variablen abzulegen, können diese in einer einzigen Datenvariablen zusammengefasst werden. Dazu werden eine weitere Dimension und eine gleichnamige Koordinatenvariable benötigt, welche im [UC]<sup>2</sup>-Datenstandard den Namen **bands\_xyz** erhalten. Dabei darf der Wert für **xyz** beliebig gewählt werden, was es erlaubt, unterschiedliche Spektren in einer einzigen *NetCDF*-Datei gemeinsam abzuspeichern. So könnten z.B. die Dimensionen und Koordinatenvariablen **bands\_landsat\_tm** und **bands\_pm** verwendet werden, sollten Satellitendaten des Landsat Thematic Mapper (TM) Sensors zusammen mit Aerosolspektren für Feinstaub in einer Datei zusammengefasst werden.

Die Koordinatenvariable **bands\_xyz** kann gemäß *CF-1.7* unterschiedliche numerische Datentypen (**int**, **float**, etc.) annehmen und enthält Werte, welche die spektralen Bänder charakterisieren. Beispielsweise können die Werte ganzzahlige Nummern der spektralen Bänder, mittlere Wellenlängen der Wellenlängenintervalle oder mittlere Durchmesser der Größenklassen sein. Im Falle einer kontinuierlichen geophysikalischen Variable können, wie z.B. bei der Variable **time**, mittels des Attributs **bounds** in einer weiteren Variable die Intervallgrenzen festgelegt werden. Dabei gelten die in Abschnitt [3.6](#) beschriebenen Regeln zum bounds-Mechanismus.

Sollten weitere Informationen zu den spektralen Kanälen vorliegen (zum Beispiel Nummern oder Bezeichnungen von Spektralkanälen), können diese als Hilfskoordinatenvariable mit der Dimension **bands\_xyz** definiert werden. Solche Hilfskoordinatenvariablen müssen dabei mit **bands\_xyz\_abc** bezeichnet werden, wobei der Wert für **abc** frei gewählt werden kann. Auch sie können über das Attribut **bounds** auf eine Variable mit Intervallgrenzen verweisen, welche dann entsprechend mit **bands\_xyz\_abc\_bounds** bezeichnet wird. Dabei gelten die in Abschnitt [3.6](#) beschriebenen Regeln zum bounds-Mechanismus.

**Wichtig:** Wie generell erforderlich, müssen alle Koordinaten- und Hilfskoordinatenvariablen im Attribut **coordinates** der jeweiligen Datenvariablen angegeben werden!

Datenvariablen werden schließlich als multidimensionale Array aller Koordinatenvariablen definiert, wobei die Dimension **bands\_xyz** gemäß *CF-1.7* ganz links (am langsamsten variierend) stehen muss. Im Anhang [A6](#) ist ein Beispiel für spektrale Daten angeführt.

## 5 Unterstützte `featureType`-Varianten

Wie in Abschnitt [2.4](#) beschrieben, definiert *CF-1.7* unterschiedliche `featureType`-Varianten, um spezifische, diskrete Sampling-Geometrien zu unterstützen. Nachfolgend werden die im [UC]<sup>2</sup>-Datenstandard unterstützten drei `featureType`-Varianten beschrieben.

### 5.1 Zeitreihen (`featureType="timeSeries"`)

Die Verwendung von Zeitreihen ist vorgesehen, wenn innerhalb eines Gebiets, das durch die globalen Attribute `location` und `site` spezifiziert ist, an einem oder mehreren Orten Zeitreihendaten vorliegen, die in einer *NetCDF*-Datei abgespeichert werden sollen.

Die Dimension `station` spezifiziert die Anzahl der Orte mit unterschiedlichen horizontalen oder vertikalen Koordinaten. Die Bezeichnung `station` wurde dabei in Anlehnung an *CF-1.7* gewählt, kann aber in allgemeiner Weise auch dazu verwendet werden, um beispielsweise mehrere Zeitreihen, die an einem Ort in unterschiedlichen Höhen vorliegen, voneinander zu unterscheiden. Außerdem muss es sich bei den Daten nicht ausschließlich um Daten von Observierungen handeln, sondern können auch die Ergebnisse numerischer Modellsimulationen an sogenannten virtuellen Stationen sein.

**Wichtig:** Für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, wird festgelegt, dass der `featureType` `"timeSeries"` auch dann verwendet werden muss, wenn es nur eine einzige (virtuelle) Station gibt! Die Dimension `station` hat dann den Wert 1!

Die Hilfskoordinatenvariable `station_name(station,max_name_len)`, welche die Stationsnamen enthält, muss gemäß dem Beispiel im Anhang [A2](#) definiert werden. Die Dimension `max_name_len` ist dabei auf 32 Zeichen festgelegt und limitiert somit die Anzahl der Zeichen des längsten gültigen Namens. Die Stationsnamen können frei gewählt werden, wobei nur ASCII-Zeichen verwendet werden dürfen. Alle Stationsnamen müssen gemäß *CF-1.7* unterschiedlich sein. Das Attribut `long_name` muss den Wert `"station name"`, das Attribut `standard_name` den Wert `"platform_name"` und das Attribut `cf_role` den Wert `"timeseries_id"` haben.

Jede Datenvariable muss im Attribut `coordinates` neben den räumlichen und zeitlichen Hilfskoordinatenvariablen die zusätzliche Hilfskoordinatenvariable `station_name` anführen, um die Daten den Stationen eindeutig zuzuordnen. Außerdem muss eine Variable `station_h(station)` definiert werden, welche für jede Station die jeweilige Höhe der Oberfläche enthält (siehe Abschnitt [3.2](#) und Anhang [A2](#)). Dabei kann die Oberfläche entweder die Gelände- oder eine Dachfläche sein.

Die Dimension `ntime` definiert die größte Anzahl von Zeitpunkten bzw. Zeitintervallen, die in einer der Zeitreihen vorhanden ist. Die Hilfskoordinatenvariable `time` ist ein zweidimensionales Array `time(station,ntime)`, so dass für jede Zeitreihe die Zeiten unabhängig voneinander definiert werden können. Bei unterschiedlich langen Zeitreihen müssen die Zeiten der kürzeren Zeitreihen mit Fehlwerten aufgefüllt werden.

**Wichtig:** Bei zeitlich aggregierten Größen hat die Variable `time_bounds(station,ntime,nv)`, welche die Grenzen der Zeitintervalle spezifiziert, neben `station` und `ntime` die zusätzliche Dimension `nv` mit dem Wert 2 (siehe Anhang [A2](#) Beispiel "Zeitreihen")!

Punktdaten können als Zeitreihen betrachtet werden, welche sich nur auf einen Zeitpunkt bzw. ein Zeitintervall beziehen. Die Dimension `ntime` hat dann den Wert 1. Daher wird im [UC]<sup>2</sup>-Datenstandard der `featureType` `"point"` nicht unterstützt.

**Wichtig:** Zeitreihen sind ideal, um Daten entlang von Profilen abzuspeichern, die nicht ausschließlich vertikal ausgerichtet sind und sich daher in jeder Höhe `z` auch die Werte von `x` oder `y` ändern können!

## 5.2 Zeitreihen von Vertikalprofilen (`featureType="timeSeriesProfile"`)

Die Verwendung von Zeitreihen von Vertikalprofilen ist vorgesehen, wenn für ein Gebiet, das durch die globalen Attribute `location` und `site` spezifiziert ist, an einem oder mehreren Orten Zeitreihen von Vertikalprofilen vorliegen, die in einer *NetCDF*-Datei abgespeichert werden sollen. Dabei sind alle Werte eines Vertikalprofils jeweils für den gleichen Zeitpunkt bzw. das gleiche Zeitintervall gültig.

Die Dimension `station` spezifiziert die Anzahl der Orte mit unterschiedlichen horizontalen oder vertikalen Koordinaten. Die Bezeichnung `station` wurde dabei in Anlehnung an *CF-1.7* gewählt, kann aber in allgemeiner Weise auch dazu verwendet werden, um beispielsweise mehrere Zeitreihen von Vertikalprofilen, die an einem Ort an unterschiedlichen Höhen vorliegen (z.B. von zwei unterschiedlichen, höhenversetzten Profiler-Systemen erhoben wurden), voneinander zu unterscheiden. Außerdem muss es sich bei den Daten nicht ausschließlich um Daten von Observierungen handeln, sondern können auch die Ergebnisse numerischer Modellsimulationen an sogenannten virtuellen Stationen sein.

**Wichtig:** Für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, wird festgelegt, dass der `featureType "timeSeriesProfile"` auch dann verwendet werden muss, wenn es nur eine einzige Station gibt! Die Dimension `station` hat dann den Wert **1**!

Die Hilfskoordinatenvariable `station_name(station,max_name_len)`, welche die Stationsnamen enthält, muss gemäß dem Beispiel im Anhang [A3](#) definiert werden. Die Dimension `max_name_len` ist dabei auf 32 Zeichen festgelegt und limitiert somit die Anzahl der Zeichen des längsten gültigen Namens. Die Stationsnamen können frei gewählt werden, wobei nur ASCII-Zeichen verwendet werden dürfen. Alle Stationsnamen müssen gemäß *CF-1.7* unterschiedlich sein. Das Attribut `long_name` muss den Wert "`station name`", das Attribut `standard_name` den Wert "`platform_name`" und das Attribut `cf_role` den Wert "`timeseries_id`" haben.

Jede Datenvariable muss im Attribut `coordinates` neben den raumzeitlichen Koordinatenvariablen die zusätzliche Hilfskoordinatenvariable `station_name` anführen, um die Daten den Stationen eindeutig zuzuordnen. Außerdem muss eine Variable `station_h(station)` definiert werden, welche für jede Station die jeweilige Höhe der Oberfläche enthält (siehe Abschnitt [3.2](#) und Anhang [A3](#)). Dabei kann die Oberfläche entweder die Erdoberfläche oder eine Dachfläche sein.

Die Dimension `ntime` definiert die größte Anzahl von Zeitpunkten bzw. Zeitintervallen, die in einer der Zeitreihen von Vertikalprofilen vorhanden ist. Die Hilfskoordinatenvariable `time` ist ein zweidimensionales Array `time(station,ntime)`, so dass für jede Zeitreihe von Vertikalprofilen die Zeiten an den einzelnen Stationen unabhängig voneinander definiert werden können. Bei unterschiedlich langen Zeitreihen müssen die kürzeren Zeitreihen mit Fehlwerten aufgefüllt werden.

**Wichtig:** Bei zeitlich aggregierten Größen hat die Variable `time_bounds(station,ntime,nv)`, welche die Grenzen der Zeitintervalle spezifiziert, neben `station` und `ntime` die zusätzliche Dimension `nv` mit dem Wert **2**!

Die Dimension `nz` definiert die größte Anzahl von Höhen bzw. Höhenintervallen, die in einer der Zeitreihen von Vertikalprofilen vorhanden ist. Die Hilfskoordinatenvariable `z` ist ein dreidimensionales Array `z(station,ntime,nz)`, so dass für jede Zeitreihe von Vertikalprofilen die Höhen an den unterschiedlichen Stationen und Zeiten unabhängig voneinander definiert werden können. Bei Zeitreihen von Vertikalprofilen mit unterschiedlicher Anzahl von Höhen bzw. Höhenintervallen müssen nicht vorhandene Werte mit Fehlwerten aufgefüllt werden.

Vertikalprofildaten, die nur zu einem Zeitpunkt vorliegen, können als Zeitreihen von Vertikalprofilen betrachtet werden, welche sich nur auf einen Zeitpunkt bzw. ein Zeitintervall beziehen. Die Dimension `ntime` hat dann den Wert **1**. Daher wird im [UC]<sup>2</sup>-Datenstandard der `featureType "profile"` nicht unterstützt.

### 5.3 Trajektorien (`featureType="trajectory"`)

Die Verwendung von Trajektorien ist vorgesehen, wenn es für ein Gebiet, das durch die globalen Attribute `location` und `site` spezifiziert ist, eine oder mehrere Reihen von Orten gibt, an welchen für jeweils aufeinanderfolgende Zeitpunkte bzw. Zeitintervalle Daten einer oder mehrerer Größen vorliegen, die in einer *NetCDF*-Datei abgespeichert werden sollen. Trajektorien können beliebig im Raum orientiert sein.

**Wichtig:** Im Gegensatz zu einem Profil, für welches die Daten in unterschiedlichen Höhen zeitgleich vorliegen, sind die Daten entlang einer Trajektorie zu unterschiedlichen, streng monoton aufeinander folgenden Zeitpunkten bzw. Zeitintervallen gültig, können aber an identischen Orten vorliegen!

Die Dimension `traj` spezifiziert die Anzahl der Trajektorien. Mit diesem `featureType` können auch Zeitreihen von Trajektorien, d.h. wiederholte Messungen bzw. Simulationen entlang der gleichen Orte, gespeichert werden. Jede Wiederholung stellt in diesem Fall eine eigene Trajektorie dar.

**Wichtig:** Für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert werden, wird festgelegt, dass der `featureType` "`trajectory`" auch dann verwendet werden muss, wenn es nur eine einzige Trajektorie gibt! Die Dimension `traj` hat dann den Wert 1!

Die Hilfskoordinatenvariable `traj_name(traj, max_name_len)`, welche die Trajektoriennamen enthält, muss gemäß dem Beispiel im Anhang [A4](#) definiert werden. Die Dimension `max_name_len` ist dabei auf 32 Zeichen festgelegt und limitiert somit die Anzahl der Zeichen des längsten gültigen Namens. Die Trajektoriennamen können frei gewählt werden, wobei nur ASCII-Zeichen verwendet werden dürfen. Alle Trajektoriennamen müssen gemäß *CF-1.7* unterschiedlich sein. Das Attribut `long_name` muss den Wert "`trajectory name`", das Attribut `standard_name` den Wert "`platform_name`" und das Attribut `cf_role` den Wert "`trajectory_id`" haben.

Außerdem muss eine Datenvariable `height` gemäß dem Beispiel im Anhang [A4](#) definiert werden, welche für jeden Ort die jeweilige Höhe über bzw. unter der Oberfläche enthält (siehe Abschnitt [3.2](#)). Dabei kann `height` entweder eine skalare Variable (einheitliche Höhe über bzw. unter der Oberfläche) oder ein Array mit gleicher Struktur wie `z` sein. Die Höhe der Oberfläche ergibt sich somit aus der Differenz von `z` und `height`.

Die Dimension `ntime` definiert die größte Anzahl von Zeitpunkten bzw. Zeitintervallen, die in einer der Trajektorien vorhanden ist. Die Hilfskoordinatenvariable `time` ist ein zweidimensionales Array `time(traj, ntime)`, so dass für jede Trajektorie die Anzahl der Zeiten unabhängig voneinander definiert werden kann. Bei Trajektorien mit unterschiedlicher Anzahl von Zeiten müssen nicht vorhandene Werte mit Fehlwerten aufgefüllt werden.

**Wichtig:** Im Falle zeitlich aggregierter Größen hat die Variable `time_bounds(traj, ntime, nv)`, welche die Grenzen der Zeitintervalle spezifiziert, neben `traj` und `ntime` die zusätzliche Dimension `nv` mit dem Wert 2!

## 6 Dateibezeichnungen im DMS

Die Bereitstellung von Daten erfolgt im DMS über *NetCDF*-Dateien, die einer vorgegebenen Namenskonvention unterliegen. Jeder Partner, der Daten über das DMS bereitstellt, kann Dateien vor dem Hochladen (Upload) auf das DMS beliebig benennen (es wird jedoch die u.a. Benennung empfohlen). Beim Hochladen auf das DMS werden die Dateien nach erfolgreicher Prüfung der Konsistenz der Dateiinhalte gemäß folgendem Schlüssel auf Basis der in der Datei gespeicherten Werte der globalen Attribute umbenannt:

```
<campaign>-<location>-<site>-<acronym>-<data_content>-<origin_date>-<version>.nc
```

Falls das optionale globale Attribut **data\_specifier** definiert wurde, wird dessen Wert hinter dem von **data\_content** zusätzlich in den Dateinamen aufgenommen:

```
<campaign>-<location>-<site>-<acronym>-<data_content>-<data_specifier>-  
<origin_date>-<version>.nc
```

Beispielsweise könnte ein Dateiname folgendermaßen aussehen:

```
IOP01-B-rothab1lawn-TUBklima-meteo-20170117-001.nc
```

Dabei gelten folgende Konvertierungsregeln:

1. Die Werte für die Dateinameanteile **<campaign>**, **<location>**, **<site>**, **<acronym>** und **<data\_content>** werden direkt aus den Werten der entsprechenden globalen Attribute gebildet. Dabei werden Minuszeichen "-" in Attributen bei der Generierung der Dateinamen in Unterstriche "\_" konvertiert, da Minuszeichen zur Trennung der Dateinameanteile benötigt werden.
2. Der Wert für **<origin\_date>** wird aus dem globalen Attribut **origin\_time** abgeleitet und im Format "YYYYMMDD" ("YYYY" = Jahr mit vier Ziffern; "MM" = Monat mit zwei Ziffern; "DD" = Tag mit zwei Ziffern) dargestellt.
3. Der Wert für **<version>** ergibt sich aus der Versionsnummer (1-999), die im gleichnamigen globalen Attribut enthalten ist und mit drei Ziffern ("001" bis "999") dargestellt wird.

Vom DMS aufgelistete bzw. heruntergeladene Dateien (Download) lassen somit aufgrund der Dateinamen in einfacher, leicht lesbarer Weise Rückschlüsse auf die darin enthaltenen Datensätze zu.

Bei der Zusammenstellung eines Datensatzes, welcher in einer *NetCDF*-Datei auf das DMS hochgeladen werden soll, muss berücksichtigt werden, dass alle Daten, welche zum gleichen Dateinamen führen würden, in einer Datei zusammengefasst werden müssen. Um die Unterschiedlichkeit von Dateinamen zu gewährleisten, kann das optionale globale Attribut **data\_specifier** (siehe Abschnitt [2.5](#)) verwendet werden. Wird dieses Attribut nicht verwendet, sind bei der Doppelung von Dateinamen vor allem zwei potenzielle Problembereiche zu beachten:

1. Wenn eine **<site>** einer **<location>** mehrere Einzelstationen umfasst, die von einer Institution (identifiziert über **<acronym>**) während eines Zeitraumes (identifiziert über **<campaign>**) gleichzeitig (identifiziert über **<origin\_date>**) betrieben wurde, müssen alle betroffenen Stationsdaten mit gleichem **<data\_content>** in derselben Datei gespeichert werden. Dies hat zur Folge, dass eine entsprechende **featureType**-Variante (z.B. "timeSeries"; siehe Abschnitt [5.1](#)) verwendet werden muss.
2. Daten für unterschiedliche Größen können entweder in separaten *NetCDF*-Dateien einzeln abgespeichert oder in einer *NetCDF*-Datei als Datensatz zusammengefasst werden. Die jeweiligen Werte für das globale Attribut **data\_content** sind in den Tabellen [A1](#) und [A2](#) zu finden. Bei der Verwendung von Sammelbezeichnungen wie "meteo" oder "air" (siehe [Tabelle A2](#)) müssen alle Variablen, die zu dieser Variablengruppe gehören, in derselben *NetCDF*-Datei gespeichert werden.



**Wichtig:** Sollte eine *NetCDF*-Datei mit einem unvollständigen Datensatz bereits auf das DMS hochgeladen worden sein, so dass die Ergänzung des Datensatzes zu einem Dateinamenskonflikt führen würde, muss der gesamte Datensatz neu erstellt und mit einer höheren Versionsnummer auf das DMS hochgeladen werden!

Bei der Zusammenstellung der Datensätze sollte zudem beachtet werden, dass das Abspeichern einzelner Größen in separaten *NetCDF*-Dateien zwar die Identifizierbarkeit der Daten über den Dateinamen mittels `<data_content>` verbessert, aber auch die Anzahl der *NetCDF*-Dateien erhöht. Außerdem wird dringend davon abgeraten, die gleichen Daten in unterschiedlichen *NetCDF*-Dateien (z.B. sowohl einzeln als in einer Gruppe, oder mehrfach in unterschiedlichen Gruppen) bereitzustellen.

## Anhang

### A1 Beispiel "Multidimensionale Daten"

```
// Global attributes:

:title           = "Multidimensional data example";
:data_content    = "ta";
:source         = "model";
:version        = 1s;
:Conventions    = "CF-1.7";
:institution    = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBklima";
:author         = "Scherer, Dieter, dieter.scherer@tu-berlin.de;
                  Holtmann, Achim, achim.holtmann@tu-berlin.de";
:contact_person = "Meier, Fred, fred.meier@tu-berlin.de;
                  Fehrenbach, Ute, ute.fehrenbach@tu-berlin.de";

:dependencies   = "";
:history        = "";
:references     = "";
:comment        = "demo data for hourly instantaneous values";
:keywords       = "air temperature; Berlin; gridded data";
:licence        = "[UC]2 Open Licence; see [UC]2 data policy available at
                  www.uc2-program.org/uc2_data_policy.pdf";

:campaign       = "IOP01";
:origin_time    = "2017-01-17 00:00:00 +00";
:creation_time  = "2020-05-07 15:15:05 +00";
:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.;
:origin_y       = 5813054.;
:origin_lon     = 13.3136143016031;
:origin_lat     = 52.4556312383161;
:origin_z       = 42.;
:rotation_angle = 0.f;

dimensions:

    time = 24;
    z    = 20;
    y    = 50;
    x    = 40;

variables:

    int time(time);
        time:long_name      = "time";
        time:standard_name = "time";
        time:units          = "seconds since 2017-01-17 00:00:00 +00";
        time:calendar       = "proleptic_gregorian";
        time:axis           = "T";

    int vrs;
        vrs:long_name      = "vertical reference system";
        vrs:system_name   = "DHHN2016";

    float z(z);
        z:long_name       = "height above origin";
        z:units           = "m";
        z:axis            = "Z";
        z:positive        = "up";

    int crs;
        crs:long_name      = "coordinate reference system";
        crs:grid_mapping_name = "transverse_mercator";
        crs:semi_major_axis = 6378137.f;
        crs:inverse_flattening = 298.257222101;
```

```

    crs:longitude_of_prime_meridian      = 0.f;
    crs:longitude_of_central_meridian    = 15.f;
    crs:scale_factor_at_central_meridian = 0.9996f;
    crs:latitude_of_projection_origin    = 0.f;
    crs:false_easting                    = 500000.f;
    crs:false_northing                   = 0.f;
    crs:units                             = "m";
    crs:epsg_code                         = "EPSG:25833";

float y(y);
    y:long_name = "distance to origin in y-direction";
    y:units     = "m";
    y:axis      = "Y";

float x(x);
    x:long_name = "distance to origin in x-direction";
    x:units     = "m";
    x:axis      = "X";

double N_UTM(y);
    N_UTM:long_name      = "northing";
    N_UTM:standard_name = "projection_y_coordinate";
    N_UTM:units          = "m";

double E_UTM(x);
    E_UTM:long_name      = "easting";
    E_UTM:standard_name = "projection_x_coordinate";
    E_UTM:units          = "m";

double lat(y, x);
    lat:long_name      = "latitude";
    lat:standard_name = "latitude";
    lat:units          = "degrees_north";

double lon(y, x);
    lon:long_name      = "longitude";
    lon:standard_name = "longitude";
    lon:units          = "degrees_east";

float ta(time, z, y, x);
    ta:long_name      = "air temperature";
    ta:standard_name = "air_temperature";
    ta:units          = "K";
    ta:_FillValue     = -9999.f;
    ta:coordinates    = "lon lat E_UTM N_UTM x y z time";
    ta:grid_mapping   = "crs";

data:
    time = 3600, 7200, ..., 82800, 86400;

    vrs = _; // no value assigned

    z = 0.00000, 1.00000, ..., 18.0000, 19.0000;

    crs = _; // no value assigned

    y = 0.00000, 1.00000, ..., 48.0000, 49.0000;

    x = 0.00000, 1.00000, ..., 38.0000, 39.0000;

    N_UTM = 5813054.0, 5813055.0, ..., 5813102.0, 5813103.0;

    E_UTM = 385412.00, 385413.00, ..., 385450.00, 385451.00;

    lat = 52.455631, 52.455631, ..., 52.455639, 52.455639,
          52.455640, 52.455640, ..., 52.455648, 52.455648,
          ..., ..., ..., ...,
          52.456063, 52.456063, ..., 52.456071, 52.456071,
          52.456072, 52.456072, ..., 52.456080, 52.456080;

```

```
lon = 13.313614, 13.313629, ..., 13.314173, 13.314188,  
      13.313614, 13.313629, ..., 13.314173, 13.314188,  
      ..., ..., ..., ...,  
      13.313598, 13.313613, ..., 13.314157, 13.314172,  
      13.313597, 13.313612, ..., 13.314156, 13.314171;  
  
ta = 0.0226138, -0.0270021, ..., 0.547265, 0.647889;
```

## A2 Beispiel "Zeitreihen"

```
// Global attributes:

:title           = "featureType example for timeSeries data";
:data_content    = "ta";
:source         = "AWS";
:version        = 1s;
:Conventions    = "CF-1.7";
:institution     = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBKlima";
:author         = "Scherer, Dieter, dieter.scherer@tu-berlin.de;
                  Holtmann, Achim, achim.holtmann@tu-berlin.de";
:contact_person = "Meier, Fred, fred.meier@tu-berlin.de; Fehrenbach,
                  Ute, ute.fehrenbach@tu-berlin.de";

:dependencies   = "";
:history        = "";
:references     = "JOHANSSON, E., S. THORSSON, R. EMMANUEL, E. KRÜGER 2014:
                  Instruments and methods in outdoor thermal comfort studies -
                  The need for standardization - Urban Climate 10, 346-366.
                  DOI: 10.1016/j.uclim.2013.12.002.
                  http://link.springer.com/article/10.1007/s12665-014-3068-1";

:comment       = "demo data for hourly mean values";
:keywords      = "air temperature; Berlin; time series data";
:licence       = "[UC]2 Open Licence; see [UC]2 data policy available at
                  www.uc2-program.org/uc2_data_policy.pdf";

:campaign      = "IOP01";
:origin_time   = "2017-01-17 00:00:00 +00";
:creation_time = "2020-05-07 15:15:05 +00";
:location      = "B";
:site         = "rothabllawn";
:origin_x     = 385412.;
:origin_y     = 5813054.;
:origin_lon   = 13.3136143016031;
:origin_lat   = 52.4556312383161;
:origin_z     = 0.;
:rotation_angle = 0.f;
:featureType  = "timeSeries";

dimensions:

    station      = 2;
    ntime        = 24;
    nv           = 2;
    max_name_len = 32;

variables:

char station_name(station, max_name_len);
    station_name:long_name      = "station name";
    station_name:standard_name = "platform_name";
    station_name:cf_role       = "timeseries_id";

int time(station, ntime);
    time:long_name      = "time";
    time:standard_name = "time";
    time:units         = "seconds since 2017-01-17 00:00:00 +00";
    time:calendar      = "proleptic_gregorian";
    time:axis          = "T";
    time:_FillValue    = -9999;
    time:bounds        = "time_bounds";

int time_bounds(station, ntime, nv);

int vrs;
    vrs:long_name      = "vertical reference system";
    vrs:system_name    = "DHHN2016";

float z(station);
```

```

z:long_name      = "height above origin";
z:standard_name = "height_above_mean_sea_level"; // origin_z = 0.0!
z:units         = "m";
z:axis         = "Z";
z:positive     = "up";

float station_h(station);
station_h:long_name      = "surface altitude";
station_h:standard_name = "surface_altitude";
station_h:units         = "m";

int crs;
crs:long_name           = "coordinate reference system";
crs:grid_mapping_name  = "transverse_mercator";
crs:semi_major_axis    = 6378137.f;
crs:inverse_flattening = 298.257222101;
crs:longitude_of_prime_meridian = 0.f;
crs:longitude_of_central_meridian = 15.f;
crs:scale_factor_at_central_meridian = 0.9996f;
crs:latitude_of_projection_origin = 0.f;
crs:false_easting     = 500000.f;
crs:false_northing    = 0.f;
crs:units             = "m";
crs:epsg_code         = "EPSG:25833";

float y(station);
y:long_name = "distance to origin in y-direction";
y:units     = "m";
y:axis     = "Y";

float x(station);
x:long_name = "distance to origin in x-direction";
x:units     = "m";
x:axis     = "X";

double N_UTM(station);
N_UTM:long_name      = "northing";
N_UTM:standard_name = "projection_y_coordinate";
N_UTM:units         = "m";

double E_UTM(station);
E_UTM:long_name      = "easting";
E_UTM:standard_name = "projection_x_coordinate";
E_UTM:units         = "m";

double lat(station);
lat:long_name      = "latitude";
lat:standard_name = "latitude";
lat:units         = "degrees_north";

double lon(station);
lon:long_name      = "longitude";
lon:standard_name = "longitude";
lon:units         = "degrees_east";

float ta(station, ntime);
ta:long_name      = "air temperature";
ta:standard_name = "air_temperature";
ta:units         = "K";
ta:_FillValue    = -9999.f;
ta:coordinates   = "lon lat E_UTM N_UTM x y z time station_name";
ta:grid_mapping  = "crs";
ta:cell_methods  = "time: mean";
ta:uncertainty_abs = 0.2f;
ta:processing_level = 1;
ta:processing_info = "Calibrated data";

data:
station_name = "AWS 1", "AWS 2";

```

```

time = 3600, 7200, ..., 82800, 86400, // last second of hours
      3600, 7200, ..., -9999, -9999; // shorter time series at 2nd station

time_bounds =    0, 3600, // boundaries of hours
                3600, 7200,
                ..., ...,
                79200, 82800,
                82800, 86400,

                0, 3600,
                3600, 7200,
                ..., ...,
                -9999, -9999,
                -9999, -9999; // shorter time series at 2nd station

vrs = _; // no value assigned

z = 44.0000, 44.5000; // 2 m above ground

station_h = 42.0000, 42.5000;

crs = _; // no value assigned

y = 0.00000, -1.00000;

x = 0.00000, 2.00000;

N_UTM = 5813054.0, 5813053.0;

E_UTM = 385412.00, 385414.00;

lat = 52.455631, 52.455623;

lon = 13.313614, 13.313644;

ta = 0.152663, -0.630299, ..., 0.674597, 0.970390,
     -0.0636457, -0.211619, ..., -9999.00, -9999.00;

```

### A3 Beispiel "Zeitreihen von Vertikalprofilen"

```
// Global attributes:
```

```
:title           = "featureType example for timeSeriesProfile data";
:data_content    = "ta";
:source         = "AWS";
:version        = 1s;
:Conventions    = "CF-1.7";
:institution     = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBKlima";
:author         = "Scherer, Dieter, dieter.scherer@tu-berlin.de;
                  Holtmann, Achim, achim.holtmann@tu-berlin.de";
:contact_person = "Meier, Fred, fred.meier@tu-berlin.de; Fehrenbach,
                  Ute, ute.fehrenbach@tu-berlin.de";
:dependencies   = "";
:history        = "";
:references     = "";
:comment        = "demo data for daily mean vertical profile values";
:keywords       = "air temperature; Berlin; vertical profile";
:licence        = "[UC]2 Open Licence; see [UC]2 data policy available at
                  www.uc2-program.org/uc2_data_policy.pdf";
:campaign       = "IOP01";
:origin_time    = "2017-01-17 00:00:00 +00";
:creation_time  = "2020-05-07 15:15:06 +00";
:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.;
:origin_y       = 5813054.;
:origin_lon     = 13.3136143016031;
:origin_lat     = 52.4556312383161;
:origin_z       = 0.;
:rotation_angle = 0.f;
:featureType    = "timeSeriesProfile";
```

```
dimensions:
```

```
station      = 1;
ntime        = 1;
nv           = 2;
max_name_len = 32;
nz           = 3;
```

```
variables:
```

```
char station_name(station, max_name_len);
  station_name:long_name      = "station name";
  station_name:standard_name = "platform_name";
  station_name:cf_role        = "timeseries_id";

int time(station, ntime);
  time:long_name      = "time";
  time:standard_name = "time";
  time:units          = "seconds since 2017-01-17 00:00:00 +00";
  time:calendar        = "proleptic_gregorian";
  time:axis            = "T";
  time:bounds          = "time_bounds";

int time_bounds(station, ntime, nv);

int vrs;
  vrs:long_name      = "vertical reference system";
  vrs:system_name    = "DHHN2016";

float z(station, ntime, nz);
  z:long_name      = "height above origin";
  z:standard_name  = "height_above_mean_sea_level"; // origin_z = 0.0!
  z:units          = "m";
  z:axis            = "Z";
```



```

z:positive      = "up";
z:bounds        = "z_bounds";

float z_bounds(station, ntime, nz, nv);

float station_h(station);
station_h:long_name      = "surface altitude";
station_h:standard_name = "surface_altitude";
station_h:units         = "m";

int crs;
crs:long_name           = "coordinate reference system";
crs:grid_mapping_name   = "transverse_mercator";
crs:semi_major_axis     = 6378137.f;
crs:inverse_flattening  = 298.257222101;
crs:longitude_of_prime_meridian = 0.f;
crs:longitude_of_central_meridian = 15.f;
crs:scale_factor_at_central_meridian = 0.9996f;
crs:latitude_of_projection_origin = 0.f;
crs:false_easting       = 500000.f;
crs:false_northing      = 0.f;
crs:units                = "m";
crs:epsg_code           = "EPSG:25833";

float y(station);
y:long_name = "distance to origin in y-direction";
y:units     = "m";
y:axis      = "Y";

float x(station);
x:long_name = "distance to origin in x-direction";
x:units     = "m";
x:axis      = "X";

double N_UTM(station);
N_UTM:long_name      = "northing";
N_UTM:standard_name = "projection_y_coordinate";
N_UTM:units         = "m";

double E_UTM(station);
E_UTM:long_name      = "easting";
E_UTM:standard_name = "projection_x_coordinate";
E_UTM:units         = "m";

double lat(station);
lat:long_name      = "latitude";
lat:standard_name = "latitude";
lat:units         = "degrees_north";

double lon(station);
lon:long_name      = "longitude";
lon:standard_name = "longitude";
lon:units         = "degrees_east";

float ta(station, ntime, nz);
ta:long_name      = "air temperature";
ta:standard_name = "air_temperature";
ta:units         = "K";
ta:_FillValue    = -9999.f;
ta:coordinates   = "lon lat E_UTM N_UTM x y z time station_name";
ta:grid_mapping  = "crs";
ta:cell_methods  = "time: mean nz: mean";

data:
station_name = "AWS 1";

time = 86400; // last second of 17.01.2017

time_bounds = 0, 86400; // time boundaries for 17.01.2017

```

```
vrs = _; // no value assigned
z = 44.0000, 46.0000, 48.0000; // mid heights of vertical intervals
z_bounds = 43.9000, 44.1000, // vertical intervals (0.2 m extent)
           45.9000, 46.1000,
           47.9000, 48.1000;
station_h = 42.0000;
crs = _; // no value assigned
y = 0.00000;
x = 0.00000;
N_UTM = 5813054.0;
E_UTM = 385412.00;
lat = 52.455631;
lon = 13.313614;
ta = -0.254583, 1.74172, 0.660524;
```

## A4 Beispiel "Trajektorien"

```
// Global attributes:

:title           = "featureType example for trajectory data";
:data_content    = "ta";
:source         = "mobile";
:version        = 1s;
:Conventions    = "CF-1.7";
:institution    = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBklima";
:author         = "Scherer, Dieter, dieter.scherer@tu-berlin.de;
                  Holtmann, Achim, achim.holtmann@tu-berlin.de";
:contact_person = "Meier, Fred, fred.meier@tu-berlin.de;
                  Fehrenbach, Ute, ute.fehrenbach@tu-berlin.de";

:dependencies   = "";
:history        = "";
:references     = "";
:comment        = "demo data for instantaneous trajectory values";
:keywords       = "air temperature; Berlin; trajectory data";
:licence        = "[UC]2 Open Licence; see [UC]2 data policy available at
                  www.uc2-program.org/uc2_data_policy.pdf";
:campaign       = "IOP01";
:origin_time    = "2017-01-17 00:00:00 +00";
:creation_time  = "2020-05-07 15:15:06 +00";
:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.;
:origin_y       = 5813054.;
:origin_lon     = 13.3136143016031;
:origin_lat     = 52.4556312383161;
:origin_z       = 0.;
:rotation_angle = 0.f;
:featureType    = "trajectory";

dimensions:

    traj          = 1;
    ntime         = 5;
    max_name_len  = 32;

variables:

char traj_name(traj, max_name_len);
    traj_name:long_name      = "trajectory name";
    traj_name:standard_name = "platform_name";
    traj_name:cf_role        = "trajectory_id";

int time(traj, ntime);
    time:long_name          = "time";
    time:standard_name     = "time";
    time:units              = "seconds since 2017-01-17 00:00:00 +00";
    time:calendar           = "proleptic_gregorian";
    time:axis               = "T";

int vrs;
    vrs:long_name          = "vertical reference system";
    vrs:system_name        = "DHHN2016";

float z(traj, ntime);
    z:long_name            = "height above origin";
    z:standard_name        = "height_above_mean_sea_level"; // origin_z = 0.0!
    z:units                = "m";
    z:axis                 = "Z";
    z:positive             = "up";

float height;
    height:long_name       = "height above surface";
    height:standard_name   = "height";
```

```

height:units          = "m";

int crs;
  crs:long_name        = "coordinate reference system";
  crs:grid_mapping_name = "transverse_mercator";
  crs:semi_major_axis  = 6378137.f;
  crs:inverse_flattening = 298.257222101;
  crs:longitude_of_prime_meridian = 0.f;
  crs:longitude_of_central_meridian = 15.f;
  crs:scale_factor_at_central_meridian = 0.9996f;
  crs:latitude_of_projection_origin = 0.f;
  crs:false_easting    = 500000.f;
  crs:false_northing   = 0.f;
  crs:units            = "m";
  crs:epsg_code        = "EPSG:25833";

float y(traj, ntime);
  y:long_name = "distance to origin in y-direction";
  y:units     = "m";
  y:axis      = "Y";

float x(traj, ntime);
  x:long_name = "distance to origin in x-direction";
  x:units     = "m";
  x:axis      = "X";

double N_UTM(traj, ntime);
  N_UTM:long_name      = "northing";
  N_UTM:standard_name = "projection_y_coordinate";
  N_UTM:units          = "m";

double E_UTM(traj, ntime);
  E_UTM:long_name      = "easting";
  E_UTM:standard_name = "projection_x_coordinate";
  E_UTM:units          = "m";

double lat(traj, ntime);
  lat:long_name      = "latitude";
  lat:standard_name = "latitude";
  lat:units          = "degrees_north";

double lon(traj, ntime);
  lon:long_name      = "longitude";
  lon:standard_name = "longitude";
  lon:units          = "degrees_east";

float ta(traj, ntime);
  ta:long_name      = "air temperature";
  ta:standard_name = "air_temperature";
  ta:units          = "K";
  ta:_FillValue    = -9999.f;
  ta:coordinates   = "lon lat E_UTM N_UTM x y z time traj_name";
  ta:grid_mapping  = "crs";

data:
  traj_name = "Trajectory 1";

  time = 43200, 43800, 44400, 45000, 45600; // 12 UTC, 10 min time steps

  vrs = _; // no value assigned

  z = 43.1000, 43.2000, 43.9000, 43.5000, 43.4000; // altitudes of measurements

  height = 1.10000; // constant height above ground

  crs = _; // no value assigned

  y = 0.00000, 1.00000, 0.200000, 2.20000, 3.00000;

  x = 0.00000, 1.30000, 2.50000, 4.70000, 5.80000;

```

```
N_UTM = 5813054.0, 5813055.0, 5813054.2, 5813056.2, 5813057.0;  
E_UTM = 385412.00, 385413.30, 385414.50, 385416.70, 385417.80;  
lat = 52.455631, 52.455640, 52.455634, 52.455652, 52.455659;  
lon = 13.313614, 13.313633, 13.313651, 13.313683, 13.313699;  
ta = 0.861230, -1.53924, 0.710852, 1.28526, -0.224212;
```

## A5 Beispiel "Hilfsvariablen und Flags"

```
// Global attributes:
```

```
:title           = "example for ancillary variables and flags";
:data_content    = "meteo";
:source         = "AWS";
:version        = 1s;
:Conventions    = "CF-1.7";
:institution     = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBKlima";
:author         = "Scherer, Dieter, dieter.scherer@tu-berlin.de;
                  Holtmann, Achim, achim.holtmann@tu-berlin.de";
:contact_person = "Meier, Fred, fred.meier@tu-berlin.de;
                  Fehrenbach, Ute, ute.fehrenbach@tu-berlin.de";
:dependencies   = "";
:history        = "";
:references     = "";
:comment        = "demo data for ancillary variables and flags";
:keywords       = "air temperature; Berlin; ancillary variables; flags";
:licence        = "[UC]2 Open Licence; see [UC]2 data policy available at
                  www.uc2-program.org/uc2_data_policy.pdf";
:campaign       = "LTO";
:origin_time    = "2017-01-01 00:00:00 +00";
:creation_time  = "2020-05-07 15:15:06 +00";
:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.;
:origin_y       = 5813054.;
:origin_lon     = 13.3136143016031;
:origin_lat     = 52.4556312383161;
:origin_z       = 0.;
:rotation_angle = 0.f;
:featureType    = "timeSeries";
```

```
dimensions:
```

```
station      = 1;
ntime        = 24;
nv           = 2;
max_name_len = 32;
```

```
variables:
```

```
char station_name(station, max_name_len);
  station_name:long_name   = "station name";
  station_name:standard_name = "platform_name";
  station_name:cf_role     = "timeseries_id";

int time(station, ntime);
  time:long_name       = "time";
  time:standard_name  = "time";
  time:units           = "seconds since 2017-01-01 00:00:00 +00";
  time:calendar        = "proleptic_gregorian";
  time:axis            = "T";
  time:bounds          = "time_bounds";

int time_bounds(station, ntime, nv);

int vrs;
  vrs:long_name   = "vertical reference system";
  vrs:system_name = "DHHN2016";

float z(station);
  z:long_name       = "height above origin";
  z:standard_name  = "height_above_mean_sea_level";
  z:units           = "m";
  z:axis            = "Z";
  z:positive        = "up";
```

```

float station_h(station);
    station_h:long_name = "surface altitude";
    station_h:standard_name = "surface_altitude";
    station_h:units = "m";

int crs;
    crs:long_name = "coordinate reference system";
    crs:grid_mapping_name = "transverse_mercator";
    crs:semi_major_axis = 6378137.f;
    crs:inverse_flattening = 298.257222101;
    crs:longitude_of_prime_meridian = 0.f;
    crs:longitude_of_central_meridian = 15.f;
    crs:scale_factor_at_central_meridian = 0.9996f;
    crs:latitude_of_projection_origin = 0.f;
    crs:false_easting = 500000.f;
    crs:false_northing = 0.f;
    crs:units = "m";
    crs:epsg_code = "EPSG:25833";

float y(station);
    y:long_name = "distance to origin in y-direction";
    y:units = "m";
    y:axis = "Y";

float x(station);
    x:long_name = "distance to origin in x-direction";
    x:units = "m";
    x:axis = "X";

double N_UTM(station);
    N_UTM:long_name = "northing";
    N_UTM:standard_name = "projection_y_coordinate";
    N_UTM:units = "m";

double E_UTM(station);
    E_UTM:long_name = "easting";
    E_UTM:standard_name = "projection_x_coordinate";
    E_UTM:units = "m";

double lat(station);
    lat:long_name = "latitude";
    lat:standard_name = "latitude";
    lat:units = "degrees_north";

double lon(station);
    lon:long_name = "longitude";
    lon:standard_name = "longitude";
    lon:units = "degrees_east";

float ta(station, ntime);
    ta:long_name = "air temperature";
    ta:standard_name = "air_temperature";
    ta:units = "degree_C";
    ta:_FillValue = -9999.f;
    ta:coordinates = "lon lat E_UTM N_UTM x y z time station_name";
    ta:grid_mapping = "crs";
    ta:cell_methods = "time: mean";
    ta:ancillary_variables = "ancillary_ta_flags ancillary_instrument_status";

float hur(station, ntime);
    hur:long_name = "relative humidity";
    hur:standard_name = "relative_humidity";
    hur:units = "1";
    hur:_FillValue = -9999.f;
    hur:coordinates = "lon lat E_UTM N_UTM x y z time station_name";
    hur:grid_mapping = "crs";
    hur:cell_methods = "time: mean";
    hur:ancillary_variables = "ancillary_instrument_status";

```

```

byte ancillary_instrument_status(station, ntime);
  ancillary_instrument_status:long_name      = "instrument status";
  ancillary_instrument_status:flag_meanings = "low_battery ventilation_on";
  ancillary_instrument_status:flag_masks    = 1b, 2b;

byte ancillary_ta_flags(station, ntime);
  ancillary_ta_flags:long_name              = "air temperature flags";
  ancillary_ta_flags:flag_meanings         = "okay out_of_range NA";
  ancillary_ta_flags:flag_values           = 0b, 1b, 2b;

data:
  station_name = "AWS 1";

  time = 1386000, 1389600, ..., 1465200, 1468800; // sec after beginning of month

  time_bounds = 1382400, 1386000, // boundaries of hours
                1386000, 1389600,
                ...,
                1461600, 1465200,
                1465200, 1468800;

  vrs = _; // no value assigned

  z = 44.0000; // 2 m above ground

  station_h = 42.0000;

  crs = _; // no value assigned

  y = 0.00000;

  x = 0.00000;

  N_UTM = 5813054.0;

  E_UTM = 385412.00;

  lat = 52.455631;

  lon = 13.313614;

  ta = -9999.00, -0.628962, ..., 30.0000, 0.774297; // first is fill, 30 too large

  hur = 0.610804, 0.653351, ..., 0.503260, 0.517553;

  ancillary_instrument_status = 0b, 1b, ..., 3b, 2b; // 1=low bat, 2=vent, 3=both

  ancillary_ta_flags = 2b, 0b, ..., 1b, 0b; // first is NA, 30 out of range

```



## A6 Beispiel "Spektrale Daten"

```
// Global attributes:

:title           = "example for spectral data";
:data_content    = "ncaa";
:source         = "APSS";
:version        = 1s;
:Conventions    = "CF-1.7";
:institution    = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBklima";
:author         = "Scherer, Dieter, dieter.scherer@tu-berlin.de;
                  Holtmann, Achim, achim.holtmann@tu-berlin.de";
:contact_person = "Meier, Fred, fred.meier@tu-berlin.de;
                  Fehrenbach, Ute, ute.fehrenbach@tu-berlin.de";

:dependencies   = "";
:history        = "";
:references     = "";
:comment        = "demo data for spectral data";
:keywords       = "particles; Berlin; spectral data";
:licence        = "[UC]2 Open Licence; see [UC]2 data policy available at www.uc2-
program.org/uc2_data_policy.pdf";
:campaign       = "LTO";
:origin_time    = "2017-01-01 00:00:00 +00";
:creation_time  = "2020-05-07 15:15:06 +00";
:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.;
:origin_y       = 5813054.;
:origin_lon     = 13.3136143016031;
:origin_lat     = 52.4556312383161;
:origin_z       = 0.;
:rotation_angle = 0.f;
:featureType    = "timeSeries";

dimensions:

    station      = 1;
    ntime        = 24;
    nv           = 2;
    max_name_len = 32;
    bands_pm     = 3;

variables:

    char station_name(station, max_name_len);
        station_name:long_name      = "station name";
        station_name:standard_name = "platform_name";
        station_name:cf_role        = "timeseries_id";

    int time(station, ntime);
        time:long_name              = "time";
        time:standard_name          = "time";
        time:units                  = "seconds since 2017-01-01 00:00:00 +00";
        time:calendar               = "proleptic_gregorian";
        time:axis                   = "T";
        time:bounds                 = "time_bounds";

    int time_bounds(station, ntime, nv);

    int vrs;
        vrs:long_name              = "vertical reference system";
        vrs:system_name            = "DHHN2016";

    float z(station);
        z:long_name                = "height above origin";
        z:standard_name            = "height_above_mean_sea_level";
        z:units                    = "m";
        z:axis                     = "Z";
```

```

z:positive      = "up";

float station_h(station);
  station_h:long_name      = "surface altitude";
  station_h:standard_name = "surface_altitude";
  station_h:units         = "m";

int crs;
  crs:long_name      = "coordinate reference system";
  crs:grid_mapping_name      = "transverse_mercator";
  crs:semi_major_axis      = 6378137.f;
  crs:inverse_flattening   = 298.257222101;
  crs:longitude_of_prime_meridian = 0.f;
  crs:longitude_of_central_meridian = 15.f;
  crs:scale_factor_at_central_meridian = 0.9996f;
  crs:latitude_of_projection_origin = 0.f;
  crs:false_easting       = 500000.f;
  crs:false_northing      = 0.f;
  crs:units               = "m";
  crs:epsg_code           = "EPSG:25833";

float y(station);
  y:long_name = "distance to origin in y-direction";
  y:units     = "m";
  y:axis      = "Y";

float x(station);
  x:long_name = "distance to origin in x-direction";
  x:units     = "m";
  x:axis      = "X";

double N_UTM(station);
  N_UTM:long_name      = "northing";
  N_UTM:standard_name = "projection_y_coordinate";
  N_UTM:units         = "m";

double E_UTM(station);
  E_UTM:long_name      = "easting";
  E_UTM:standard_name = "projection_x_coordinate";
  E_UTM:units         = "m";

double lat(station);
  lat:long_name      = "latitude";
  lat:standard_name = "latitude";
  lat:units         = "degrees_north";

double lon(station);
  lon:long_name      = "longitude";
  lon:standard_name = "longitude";
  lon:units         = "degrees_east";

float ncaa(bands_pm, station, ntime);
  ncaa:long_name      = "number concentration of ambient aerosol
                        particles in air";
  ncaa:standard_name = "number_concentration_of_ambient_aerosol_
                        particles_in_air";
  ncaa:units          = "m-3"; // particles per cubic metre
  ncaa:_FillValue     = -9999.f;
  ncaa:coordinates   = "lon lat E_UTM N_UTM x y z time station_name
                        bands_pm bands_pm_size";
  ncaa:grid_mapping  = "crs";
  ncaa:cell_methods  = "time: mean";

int bands_pm(bands_pm);
  bands_pm:long_name = "particle size class number";
  bands_pm:units     = "1";

float bands_pm_size(bands_pm);
  bands_pm_size:long_name = "particle diameter";
  bands_pm_size:units     = "m-6";

```

```

bands_pm_size:bounds      = "bands_pm_size_bounds";
float bands_pm_size_bounds(bands_pm, nv);
data:
station_name = "APSS 1";

time = 1386000, 1389600, ..., 1465200, 1468800; // sec since beginning of month

time_bounds = 1382400, 1386000, // boundaries of hours
              1386000, 1389600,
              ...,
              1461600, 1465200,
              1465200, 1468800;

vrs = _; // no value assigned

z = 44.0000; // 2 m above ground

station_h = 42.0000;

crs = _; // no value assigned

y = 0.00000;

x = 0.00000;

N_UTM = 5813054.0;

E_UTM = 385412.00;

lat = 52.455631;

lon = 13.313614;

ncaa = 1.08040, 1.10315, ..., 0.707215, 0.651837,
       0.158736, 0.604397, ..., 0.279908, 0.421663,
       1.30170, 0.739476, ..., 1.29708, 0.628731;

bands_pm = 1, 2, 3; // size class number

bands_pm_size = 0.600000, 1.75000, 6.25000; // central diameter of size class

bands_pm_size_bounds = 0.200000, 1.00000, // bounds of size classes
                      1.00000, 2.50000,
                      2.50000, 10.0000;

```