

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

Version 1.0

1. April 2018

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

<sup>6</sup>Karlsruher Institut für Technologie, Institut für Meteorologie und Klimaforschung,  
Institut für Atmosphärische Umweltforschung

<sup>7</sup>Universität Hamburg, Meteorologisches Institut

<sup>8</sup>Freie Universität Berlin, Institut für Meteorologie

<sup>9</sup>Deutscher Wetterdienst

<sup>10</sup>Karlsruher Institut für Technologie, Institut für Meteorologie und Klimaforschung,  
Troposphärenforschung

<sup>11</sup>Fraunhofer-Institut für Bauphysik

<sup>12</sup>Forschungszentrum Jülich GmbH, Institut für Energie- und Klimaforschung, IEK-8: Troposphäre

## Arbeitsgruppe "*Datenmanagement*"

der Fördermaßnahme

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

gefördert durch das

***Bundesministerium 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 einher gehenden 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 der zuvor genannten 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 sollen im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme bereits verfügbare Daten aufbereitet und fehlende Daten über Langzeitmessungen und Intensivmesskampagnen neu erhoben werden. Dazu sind auch verbesserte Konzepte und Analysewerkzeuge erforderlich, deren Erarbeitung 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) die Zusammenarbeit innerhalb der [UC]<sup>2</sup>-Fördermaßnahme sicherzustellen.

## Inhalt

Vorwort .....	ii
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 .....	5
2.3 Globale Attribute zum räumlichen Bezug der Daten .....	6
2.4 Das globale Attribut <b>featureType</b> .....	7
3 Dimensionen, Koordinatenvariablen und Hilfskoordinatenvariablen.....	9
3.1 Zeit.....	9
3.2 Vertikale Koordinaten .....	10
3.3 Horizontale Koordinaten .....	11
3.4 <i>PALM-4U</i> -Modellgitter .....	13
3.5 Orientierung von Oberflächen .....	14
4 Datenvariablen .....	17
4.1 Obligatorische Attribute.....	18
4.2 Optionale Attribute .....	19
5 Unterstützte <b>featureType</b> -Varianten .....	22
5.1 Zeitreihen ( <b>featureType="timeSeries"</b> ).....	22
5.2 Zeitreihen von Vertikalprofilen ( <b>featureType=timeSeriesProfile</b> ).....	23
5.3 Trajektorien ( <b>featureType="trajectory"</b> ) .....	24
6 Dateibezeichnungen im DMS .....	25
Anhang .....	A-1
A1: Beispiel "Multidimensionale Daten" .....	A-1
A2: Beispiel "Zeitreihen" .....	A-3
A3: Beispiel "Zeitreihen von Vertikalprofilen" .....	A-6
A4: Beispiel "Trajektorien" .....	A-9

# 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)) wird 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.

## 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 bedeutet, 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* bereits 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* ([https://icdc.cen.uni-hamburg.de/fileadmin/user\\_upload/samd\\_docs/samd\\_product\\_standard\\_v1.0.pdf](https://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 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* Codierung 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!

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 gilt, 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 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.

## 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. 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 empfohlen, sich an *CF-1.7* oder darauf aufbauenden Konventionen zu halten. Dies gilt insbesondere auch für die Groß- und Kleinschreibung der Attribute (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 dienen der allgemeinen Charakterisierung des in einer *NetCDF*-Datei gespeicherten Datensatzes.

**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 anzeigt, deren Werte beim Upload einer Konsistenzprüfung unterliegen.

Name	Bezeichnung	Typ	Wert
<b>title</b>	Kurztitel der Daten	S	Text
<b>data_content<sup>*,**</sup></b>	Bezeichnung des Inhalts	S	Text (max. 16 Zeichen); siehe Tabellen A1 und A2
<b>source</b>	Art der Erhebung	S	Text; Trennung mehrerer Erhebungsarten durch Kommata
<b>version<sup>*,**</sup></b>	Versionsnummer	I	Ganze Zahl (zwischen 1 und 999)
<b>Conventions<sup>**</sup></b>	<i>NetCDF</i> -Konventionen	S	" <b>CF-1.7</b> "
<b>dependencies<sup>**</sup></b>	Abhängigkeiten	S	Text; Trennung mehrerer Dateinamen durch Kommata
<b>history</b>	Informationen zur Datenprozessierung	S	Text
<b>institution<sup>**</sup></b>	Institution des Urhebers bzw. Herausgebers	S	Text; siehe Tabelle A3
<b>acronym<sup>*,**</sup></b>	Abkürzung der Institution	S	Text (max. 12 Zeichen); siehe Tabelle A3
<b>author<sup>**</sup></b>	Autor/in	S	Vorname, Nachname, E-Mail-Adresse
<b>contact_person<sup>**</sup></b>	Verantwortliche Person	S	Vorname, Nachname, E-Mail-Adresse
<b>references</b>	Publikationen zu Daten oder Methoden	S	Text; Trennung mehrerer Referenzen durch Kommata
<b>comment</b>	Beliebige zusätzliche Informationen	S	Text
<b>keywords</b>	Schlüsselworte für Suche im DMS	S	Text; Trennung mehrerer Keywords durch Kommata
<b>licence<sup>**</sup></b>	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**. 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 **standard\_name** (siehe Abschnitt 4.1) 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 Kommata 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 Person Autor/in des Datensatzes ist, wird 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 Person als Ansprechpartner/in 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 Lizenzbeitrag 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 anzeigen, 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**". 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 werden Minuszeichen "-" 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.

#### **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.

<b>Name</b>	<b>Bezeichnung</b>	<b>Typ</b>	<b>Wert</b>
<b>location<sup>*,**</sup></b>	Ort, auf den sich die Daten beziehen	S	Name der Region, Stadt oder Kommune
<b>site<sup>*,**</sup></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 - 359.0)

**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, was für alle Daten, welche im Rahmen der [UC]<sup>2</sup>-Fördermaßnahme generiert wurden, verbindlich vorgeschrieben ist (siehe auch Tabelle A4).

## **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 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, muss die Angabe als UTM-Rechtswert in Meter mit Bezug auf das Referenzsystem [ETRS89](#) mit dem [GRS80](#)-Ellipsoid erfolgen!

## **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 geographische 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, wird **origin\_lon** 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- 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.

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.

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.

### 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 raumzeitliche Koordinatenvariable wird eine numerische, eindimensionale Variable bezeichnet, deren Name identisch mit ihrer Dimension ist und monoton auf- oder absteigende Werte 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 raumzeitlichen 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), welche an den entsprechenden Stellen beschrieben werden.

**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 als Arrays eingegeben werden!

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 **ni**, **nj**, **ntime**, **nz** sowie **station** bzw. **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 der Zeitangaben 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).

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, die denselben Datentyp wie die Variable **time** haben muss und keine Attribute besitzen darf. Die Anzahl der Vertices (Grenzen) eines Zeitintervalls ist in der Dimension **nv** mit dem Wert 2 spezifiziert. Die Werte der in **time\_bounds** gespeicherten Zeitintervalle werden in der Variablen **time** über das Attribut **bounds** referenziert. Dabei gilt, 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 des 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.

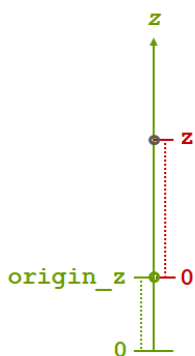
**Wichtig:** Zeitangaben von *PALM-4U*-Simulationen werden grundsätzlich als Gleitkommazahlen vom Datentyp **float** repräsentiert. Daten, die direkt in *PALM-4U* eingelesen werden sollen, müssen daher den Datentyp **float** für die Variable **time** verwenden!

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.

### 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!



**Abbildung 3.2:** Definition der vertikalen Koordinaten- bzw. Hilfskoordinatenvariable **z** (rot markiert) im [UC]<sup>2</sup>-Datenstandard. Alle absoluten 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.

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.

**Wichtig:** Im Falle von Beobachtungsdaten oder bei Verwendung einer **featureType**-Variante wird die Referenzhöhe in **origin\_z** einheitlich auf den Wert 0 gesetzt und somit **standard\_name** den Wert "height\_above\_mean\_sea\_level" haben!

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* ([www.adv-online.de/Geodaetische-Grundlagen/DHHN2016/](http://www.adv-online.de/Geodaetische-Grundlagen/DHHN2016/)) 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, die denselben Datentyp wie die Variable **z** haben muss und keine Attribute besitzen darf. Die Anzahl der Vertices (Grenzen) eines Höhenintervalls ist in der Dimension **nv** mit dem Wert 2 spezifiziert. Die Werte der in **z\_bounds** gespeicherten Höhenintervalle werden in der Variablen **z** über das Attribut **bounds** referenziert. Dabei gilt, 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** ≥ 0) bzw. die Tiefe unter Grund (**height** < 0) anzugeben.

### 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).

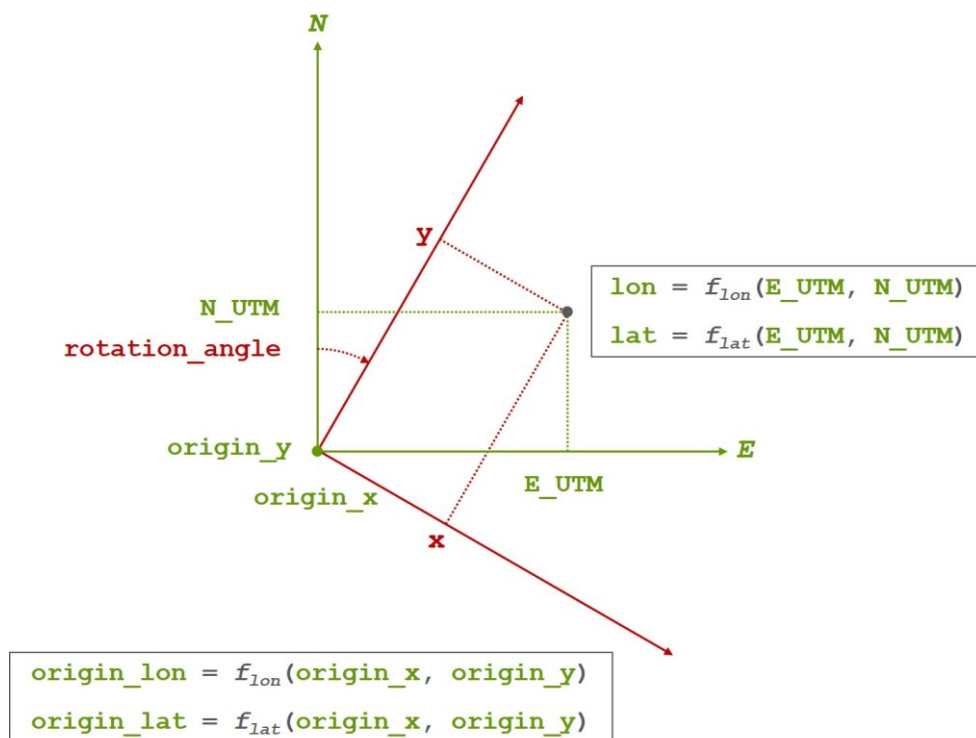
Im [UC]<sup>2</sup>-Datenstandard werden horizontale Koordinaten aber grundsätzlich über metrische Koordinaten **x** und **y** repräsentiert, um die Kompatibilität mit *PALM-4U* zu gewährleisten und um Anwendern die räumliche Zuordnung der Daten zu erleichtern. Das Attribut **long\_name** muss den Wert "distance to origin in x-direction" bzw. "distance to origin in y-direction" haben.

**Wichtig:** Das Attribut **standard\_name** darf bei **x** und **y** 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 gemäß 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. 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.

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



**Abbildung 3.3:** Definition eines rotierten, horizontalen, metrischen Koordinatensystems mit den Koordinaten **x** und **y** (rot markiert) im [UC]<sup>2</sup>-Datenstandard. Alle *UTM*-Koordinaten (hier: **E\_UTM** und **N\_UTM**, **origin\_x** und **origin\_y**; grün markiert) sowie die entsprechenden geographischen Längen und Breiten (hier: **lon** und **lat**, **origin\_lon** und **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  $f_{\text{lon}}$  und  $f_{\text{lat}}$  dienen der Umrechnung der *UTM*-Koordinaten in geographische Längen und Breiten und sind in frei verfügbaren Programmibliotheken wie PROJ.4 enthalten.

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

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

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

Der [UC]<sup>2</sup>-Datenstandard verlangt, die *UTM*-Koordinaten **E\_UTM** und **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 zugrunde liegende *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 **E\_UTM(x)** bzw. **N\_UTM(y)**, während diese in rotierten, horizontalen Koordinatensystemen als zweidimensionale Variablen **E\_UTM(y, x)** bzw. **N\_UTM(y, x)** definiert werden müssen.

Gemäß *CF-1.7* müssen alle horizontalen Koordinaten als geographische Längen und Breiten angegeben werden. Die Umrechnung kann u.a. mit frei verfügbaren Tools erfolgen (siehe z.B. [proj4.org/apps/cs2cs.html](http://proj4.org/apps/cs2cs.html), [trac.osgeo.org/osgeo4w/](http://trac.osgeo.org/osgeo4w/)). Die entsprechenden, obligatorischen Hilfskoordinatenvariablen müssen zweidimensional, d.h. als **lon(y, x)** bzw. **lat(y, x)** 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.

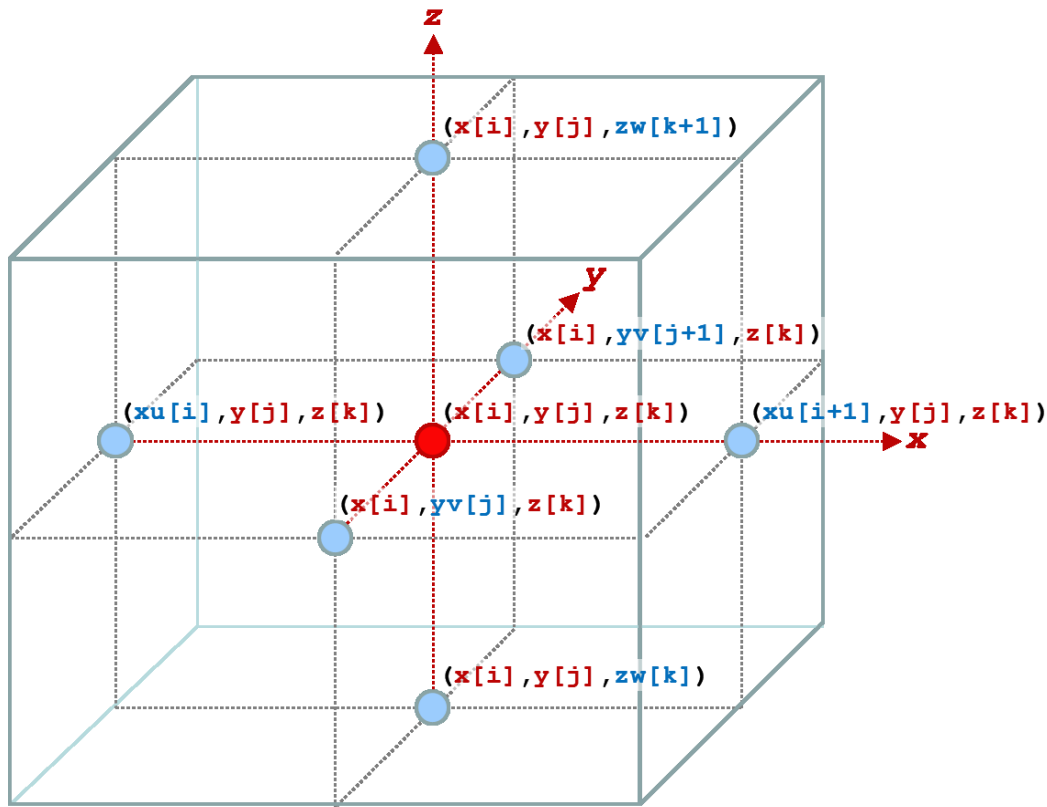
### 3.4 PALM-4U-Modellgitter

*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  $\langle Nx \rangle \cdot \langle Ny \rangle \cdot \langle Nz \rangle$  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 Luftinhaltsstoffe (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 Orientierung von Oberflächen

Um Daten für orientierte 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.

#### 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  $z_t(\mathbf{y}, \mathbf{x})$  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 zweidimensionale Felder `azimuth(y, x)` und `zenith(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 Angaben von Windrichtungen die folgende

Konvention gilt: Nord = 0; Ost = 90; Süd = 180; West = 270. 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‘). Mit dieser Festlegung entspricht bei DGM somit der Winkel **azimuth** der Exposition und der Winkel **zenith** der Hangneigung.

### 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 **zenith(s)** und **azimuth(s)** benötigt. Dazu kommen noch weitere Datenvariablen, welche die Gebäudeflächen charakterisieren.

### Pixelbasierte Oberflächendaten

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. Sofern die Koordinaten (**x[row,col],y[row,col],z[row,col]**) der Mittelpunkte der Pixel mit den Indizes (**row,col**) bekannt sind und somit die Georeferenz der Daten bekannt ist, können die Daten zusammen mit den Koordinaten in einer *NetCDF*-Datei gespeichert werden. Dazu müssen zwei zusätzliche Dimensionen **nrow** und **ncol** eingeführt werden, welche die Bildgröße spezifizieren. Somit werden die Hilfskoordinatenvariablen **x(nrow,ncol)**, **y(nrow,ncol)**, **z(nrow,ncol)**, **lon(nrow,ncol)**, **lat(nrow,ncol)**, **E\_UTM(nrow,ncol)** und **N\_UTM(nrow,ncol)** sowie die Datenvariablen, z.B. **ts(nrow,ncol)**, zweidimensionale Felder dieser beiden Dimensionen.

**Wichtig:** Bei der Definition der Dimensionen **nrow** und **ncol** muss beachtet werden, dass **ncol** am schnellsten variiert und somit i.d.R. die Anzahl der Bildspalten (columns) spezifiziert!

Es wird empfohlen, zusätzlich für jedes Pixel auch die Orientierung mittels der Datenvariablen **azimuth(nrow,ncol)** und **zenith(nrow,ncol)** anzugeben.

Sollten Bildzeitreihen zeitlich stationärer Oberflächen vorliegen, können diese in analoger Weise unter zusätzlicher Verwendung der Dimension **time** abgespeichert werden. Die Zeitdimension muss dabei als erste, d.h. am langsamsten variiere Dimension verwendet werden, z.B. **ts(time,nrow,ncol)**. Falls sich auch die Lage bzw. Orientierung der Oberflächen zeitlich ändert, werden alle räumlichen Hilfskoordinatenvariablen ebenfalls dreidimensionale Variablen (**x(time,nrow,ncol)** etc.).

Alternativ zur vorgeschlagenen Weise könnten pixelbasierte Oberflächendaten zeitlich 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

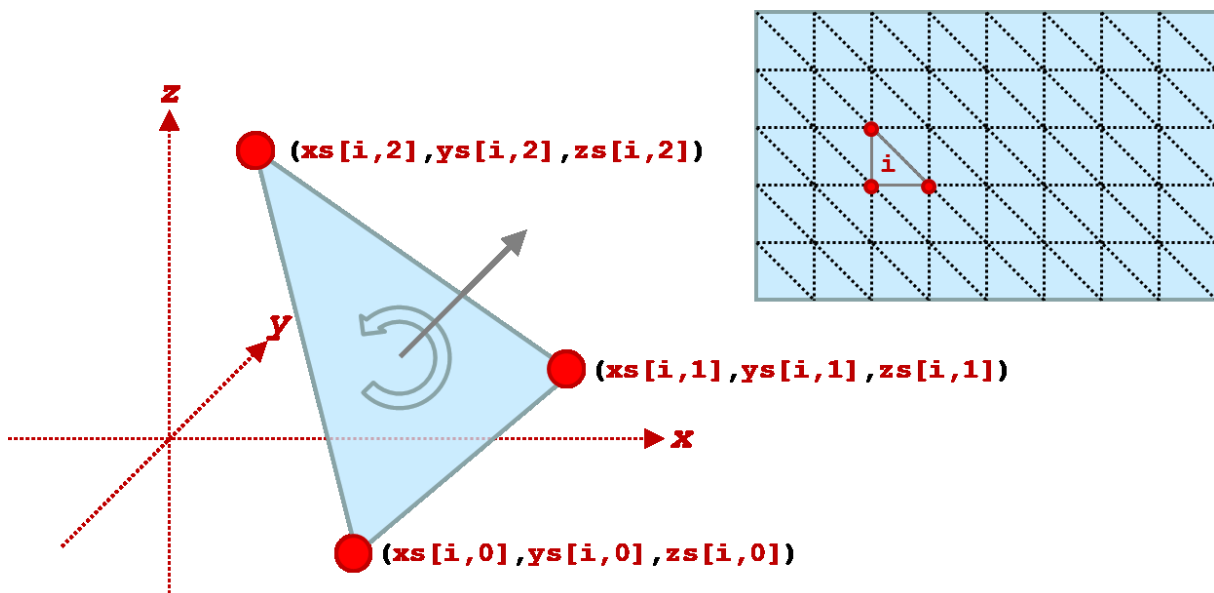
Bei den beiden o.a. Fällen sind die Eckpunkte der orientierten Flächen durch zwei- bzw. dreidimensionale Gitter definiert. Im Falle von Beobachtungsdaten, z.B. Thermalbildzeitreihen von beliebig orientierten Gebäudeflächen und Vegetationsbeständen, oder bei realitätsnahen 3D-Gebäudemodellen aus GIS-Datenbeständen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

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

Komplexe Objekte (z.B. reale Gebäude mit Satteldächern, die nicht in einem *PALM-4U*-Modellgitter dargestellt werden können), werden in elementare, miteinander verbundene Dreiecke zerlegt (Abbildung 3.5). Dabei werden die darzustellenden 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.

Eine Fläche 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 fest (siehe Abbildung 3.5). Die Flächennormale ist dabei immer in die Atmosphäre gerichtet, weshalb die Angabe von **azimuth** und **zenith** der Flächennormalen nicht erforderlich ist (zur Vereinfachung der Datennutzung können diese aber dennoch zusätzlich angegeben werden). Für jede der  $s$  Oberflächen werden somit die Koordinaten  $xs(s, npt)$ ,  $ys(s, npt)$ , und  $zs(s, npt)$ , sowie die Hilfskoordinatenvariablen  $Es\_UTM(s, npt)$ ,  $Ns\_UTM(s, npt)$ ,  $lons(s, npt)$  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: Außenraumatmosphäre; Innenwände: Innenraumatmosphäre). Komplexe bzw. ausgedehnte Oberflächen können durch Zerlegung in eine Vielzahl von Dreiecken mit hinreichender räumlicher Differenzierung dargestellt werden.

## 4 Datenvariablen

Als Datenvariable werden im [UC]<sup>2</sup>-Datenstandard Variablen bezeichnet, die keine Koordinaten- oder Hilfskoordinatenvariablen sind und keine Referenzsysteme (**vrs**, **crs**) beschreiben. Eine Datenvariable enthält somit Daten und variablenspezifische Metadaten von physikalischen bzw. chemischen 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 *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 ohne großen Aufwand möglich ist. Darüber hinaus wird festgelegt, Datenvariablen für eine Größe 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. 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>long_name</b>	Langname der Größe	S	Gemäß Tabelle A1
<b>standard_name*</b>	Standardname der Größe	S	Gemäß Tabelle A1; das Attribut darf nicht definiert werden, wenn es keinen <i>CF-1.7</i> -konformen Standardnamen gibt!
<b>units</b>	Einheit der Größe	S	Text; <i>UDUNITS</i> -konforme Einheit
<b>_FillValue</b>	Füllwert bei fehlenden oder ungültigen Daten	N	<b>-9999 bzw. -9999.0</b>
<b>coordinates</b>	Alle Koordinaten- und Hilfskoordinatenvariablen	S	Z.B. "lon lat E_UTM N_UTM x y z time"
<b>grid_mapping</b>	Koordinatenreferenzsystemvariable	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.

### **standard\_name**

Dieses Attribut ist in *CF-1.7* für eine Vielzahl von Größen festgelegt (siehe [cfconventions.org/Data/cf-standard-names/27/build/cf-standard-name-table.html](http://cfconventions.org/Data/cf-standard-names/27/build/cf-standard-name-table.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**

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/Data/cf-standard-names/27/build/cf-standard-name-table.html](http://cfconventions.org/Data/cf-standard-names/27/build/cf-standard-name-table.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 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".
```

## grid\_mapping

Dieses Attribut enthält die Zuordnung der horizontalen Hilfskoordinatenvariablen zur Koordinatenreferenzsystemvariablen **crs**.

## 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.

Name	Erläuterung	Typ	Wert
<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



### **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 im Attribut **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.

**Wichtig:** 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 Bereich 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.

## 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 müssen.

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 in A2 definiert werden. Die Dimension **max\_name\_len** ist dabei auf 32 Zeichen festgelegt und bestimmt somit die Anzahl der Zeichen des längsten gültigen Namens. Die Stationsnamen in können frei gewählt werden, wobei nur ASCII-Zeichen verwendet werden dürfen. Da **station\_name** eine Hilfskoordinatenvariable ist, müssen alle Stationsnamen 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 Beispiel in A2). 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 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 kürzeren Zeitreihen mit Fehlwerten aufgefüllt werden.

**Wichtig:** Im Falle zeitlich aggregierter 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 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!

## 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 müssen. 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 in A3 definiert werden. Die Dimension **max\_name\_len** ist dabei auf 32 Zeichen festgelegt und bestimmt somit die Anzahl der Zeichen des längsten gültigen Namens. Die Stationsnamen in können frei gewählt werden, wobei nur ASCII-Zeichen verwendet werden dürfen. Da **station\_name** eine Hilfskoordinatenvariable ist, müssen alle Stationsnamen 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 Beispiel 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:** Im Falle zeitlich aggregierter 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 müssen. 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 gleichen 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 Trajektorienamen enthält, muss gemäß dem Beispiel in A4 definiert werden. Die Dimension **max\_name\_len** ist dabei auf 32 Zeichen festgelegt und bestimmt somit die Anzahl der Zeichen des längsten gültigen Namens. Die Trajektorienamen in können frei gewählt werden, wobei nur ASCII-Zeichen verwendet werden dürfen. Da **traj\_name** eine Hilfskoordinatenvariable ist, müssen alle Trajektorienamen 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 Variable **height** gemäß dem Beispiel in 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 nach folgendem Schlüssel auf Basis der in der Datei gespeicherten Werte der globalen Attribute umbenannt:

```
<campaign>-<location>-<site>-<acronym>-<data_content>-<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 Dateinamenbestandteile **<campaign>**, **<location>**, **<site>**, **<acronym>** und **<data\_content>** werden direkt aus den Werten der entsprechenden globalen Attribute gebildet. Dabei ist zu beachten, dass Minuszeichen "-" bei der Generierung der Dateinamen in Unterstriche "\_" konvertiert 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 des Dateinamens in einfacher, leicht lesbarer Weise Rückschlüsse auf den darin enthaltenen Datensatz 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. Dabei sind 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 Dateinamenkonflikt 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        = 1;
:Conventions    = "CF-1.7";
:institution    = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBklima";
:author         = "Dieter Scherer dieter.scherer@tu-berlin.de";
:contact_person = "Dieter Scherer dieter.scherer@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  = "2018-04-01 16:23:37 +00";

:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.0;
:origin_y       = 5813054.0;
:origin_lon     = 13.313614;
:origin_lat     = 52.455634;
:origin_z       = 42.0;
:rotation_angle = 0;

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";

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

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

    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";

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

float 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";

int crs;
  crs:long_name      = "coordinate reference system";
  crs:grid_mapping_name = "transverse_mercator";
  crs:semi_major_axis = 6378137.0;
  crs:inverse_flattening = 298.257222101;
  crs:longitude_of_prime_meridian = 0.0;
  crs:longitude_of_central_meridian = 15.0;
  crs:scale_factor_at_central_meridian = 0.9996;
  crs:latitude_of_projection_origin = 0.0;
  crs:false_easting = 500000.0;
  crs:false_northing = 0.0;
  crs:units         = "m";
  crs:epsg_code     = "EPSG:25833";

float ta(time, z);
  ta:long_name      = "air temperature";
  ta:standard_name = "air_temperature";
  ta:units         = "K";
  ta:_FillValue     = -9999.9;
  ta:coordinates    = "lon lat E_UTM N_UTM x y z time";
  ta:grid_mapping   = "crs";

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

```

## A2 Beispiel "Zeitreihen"

```
// Global attributes:

:title           = "featureType example for timeSeries data";
:data_content    = "ta";
:source         = "AWS";
:version        = 1;
:Conventions    = "CF-1.7";
:institution    = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBklima";
:author         = "Dieter Scherer dieter.scherer@tu-berlin.de";
:contact_person = "Dieter Scherer dieter.scherer@tu-berlin.de";
:dependencies   = "";
:history        = "";
:references     = "JOHANNSON, 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  = "2018-04-01 16:23:37 +00";

:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.0;
:origin_y       = 5813054.0;
:origin_lon     = 13.313614;
:origin_lat     = 52.455634;
:origin_z       = 0;
:rotation_angle = 0;

:featureType    = "timeSeries";

dimensions:

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

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);

float z(station);
    z:long_name      = "height above origin";
    z:standard_name = "height_above_mean_sea_level"; // origin_z = 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";

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

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";

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

float 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";

int crs;
    crs:long_name           = "coordinate reference system";
    crs:grid_mapping_name   = "transverse_mercator";
    crs:semi_major_axis     = 6378137.0;
    crs:inverse_flattening  = 298.257222101;
    crs:longitude_of_prime_meridian = 0.0;
    crs:longitude_of_central_meridian = 15.0;
    crs:scale_factor_at_central_meridian = 0.9996;
    crs:latitude_of_projection_origin = 0.0;
    crs:false_easting       = 500000.0;
    crs:false_northing      = 0.0;
    crs:units               = "m";
    crs:epsg_code           = "EPSG:25833";

float ta(station, ntime);
    ta:long_name      = "air temperature";
    ta:standard_name  = "air_temperature";
    ta:units          = "K";
    ta:_FillValue     = -9999.9;
    ta:coordinates    = "lon lat E_UTM N_UTM x y z time station_name";
    ta:grid_mapping   = "crs";
    ta:cell_methods   = "time: mean";

data:

    station_name = "AWS 1";
    time         = 3600, 7200, ... , 82800, 86400;           // last second of hours
    time_bounds  = 0, 3600, 3600, 7200, ... , 82800, 86400; // boundaries of hours
    z            = 44.0;                                     // 2 m above ground
    station_h    = 42.0;
    y            = 0.0;
    x            = 0.0;

```

```
N_UTM      = 5813054.0;  
E_UTM      = 385412.0;  
lat        = 52.455634;  
lon        = 13.313614;  
...
```

### A3 Beispiel "Zeitreihen von Vertikalprofilen"

```
// Global attributes:

:title           = "featureType example for timeSeriesProfile data";
:data_content    = "ta";
:source         = "AWS";
:version        = 1;
:Conventions    = "CF-1.7";
:institution    = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBklima";
:author         = "Dieter Scherer dieter.scherer@tu-berlin.de";
:contact_person = "Dieter Scherer dieter.scherer@tu-berlin.de";
:dependencies    = "";
:history        = "";
:references     = "";
:comment        = "demo data for daily mean profile 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  = "2018-04-01 16:23:37 +00";

:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.0;
:origin_y       = 5813054.0;
:origin_lon     = 13.313614;
:origin_lat     = 52.455634;
:origin_z       = 0;
:rotation_angle = 0;

:featureType    = "timeSeriesProfile";

dimensions:

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

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);

    float z(station, ntime, nz);
        z:long_name                 = "height above origin";
        z:standard_name             = "height_above_mean_sea_level"; // origin_z = 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";

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

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";

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

float 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";

int crs;
  crs:long_name = "coordinate reference system";
  crs:grid_mapping_name = "transverse_mercator";
  crs:semi_major_axis = 6378137.0;
  crs:inverse_flattening = 298.257222101;
  crs:longitude_of_prime_meridian = 0.0;
  crs:longitude_of_central_meridian = 15.0;
  crs:scale_factor_at_central_meridian = 0.9996;
  crs:latitude_of_projection_origin = 0.0;
  crs:false_easting = 500000.0;
  crs:false_northing = 0.0;
  crs:units = "m";
  crs:epsg_code = "EPSG:25833";

float ta(station, ntime, nz);
  ta:long_name = "air temperature";
  ta:standard_name = "air_temperature";
  ta:units = "K";
  ta:_FillValue = -9999.9;
  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";
  station_h = 42.0;
  time = 86400; // last second of 17.01.2017
  time_bounds = 0, 86400; // time boundaries for 17.01.2017
  z = 44.0, 46.0, 48.0; // mid heights of vertical intervals

```

```
z_bounds = 43.9, 44.1, ... , 47.9, 48.1; // vertical intervals (0.2 m extent)
...
```



## A4 Beispiel "Trajektorien"

```
// Global attributes:

:title           = "featureType example for trajectory data";
:data_content    = "ta";
:source         = "mobile";
:version        = 1;
:Conventions    = "CF-1.7";
:institution     = "Technische Universität Berlin, Fachgebiet Klimatologie";
:acronym        = "TUBklima";
:author         = "Dieter Scherer dieter.scherer@tu-berlin.de";
:contact_person = "Dieter Scherer dieter.scherer@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  = "2018-04-01 16:23:37 +00";

:location       = "B";
:site           = "rothabllawn";
:origin_x       = 385412.0;
:origin_y       = 5813054.0;
:origin_lon     = 13.313614;
:origin_lat     = 52.455634;
:origin_z       = 0;
:rotation_angle = 0;

:featureType    = "trajectory";

dimensions:

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

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";

float z(traj, ntime);
    z:long_name            = "height above origin";
    z:standard_name       = "height_above_mean_sea_level"; // origin_z = 0!
    z:units               = "m";
    z:axis                = "Z";
    z:positive            = "up";

float height(traj, ntime);
    height:long_name       = "height above surface";
    height:standard_name  = "height";
    height:units          = "m";
```

```

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

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 = "Y";

float N_UTM(traj);
  N_UTM:long_name = "northing";
  N_UTM:standard_name = "projection_y_coordinate";
  N_UTM:units = "m";

float E_UTM(traj);
  E_UTM:long_name = "easting";
  E_UTM:standard_name = "projection_x_coordinate";
  E_UTM:units = "m";

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

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

int crs;
  crs:long_name = "coordinate reference system";
  crs:grid_mapping_name = "transverse_mercator";
  crs:semi_major_axis = 6378137.0;
  crs:inverse_flattening = 298.257222101;
  crs:longitude_of_prime_meridian = 0.0;
  crs:longitude_of_central_meridian = 15.0;
  crs:scale_factor_at_central_meridian = 0.9996;
  crs:latitude_of_projection_origin = 0.0;
  crs:false_easting = 500000.0;
  crs:false_northing = 0.0;
  crs:units = "m";
  crs:epsg_code = "EPSG:25833";

float ta(traj, ntime);
  ta:long_name = "air temperature";
  ta:standard_name = "air_temperature";
  ta:units = "K";
  ta:_FillValue = -9999.9;
  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, 45800; // 12 UTC, 10 min time steps
  z = 43.1, 43.2, 43.9, 43.5, 43.4; // altitudes of measurements
  height = 1.1; // constant height above ground
  ...

```