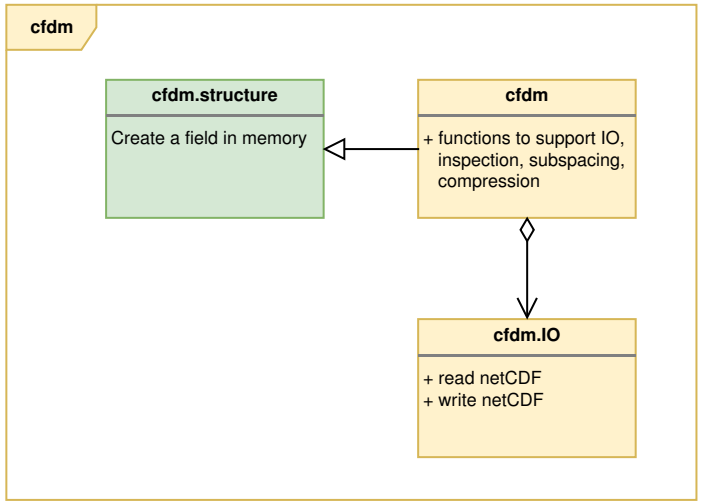
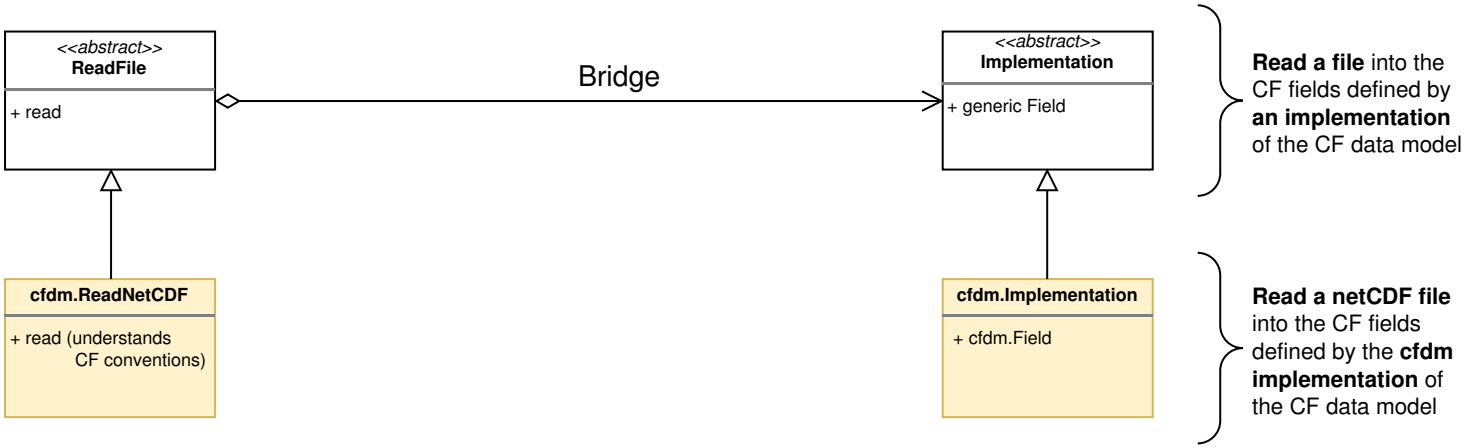
CF data model reference implementation




```
>>> import cfdm
>>> f = cfdm.structure.Field()
>>> f.set_property('standard_name', 'iron_growth_limitation_of_picophytoplankton'
>>> print f.get_property('standard_name')
'iron_growth_limitation_of_picophytoplankton'
```

```
>>> import cfdm
>>> f = cfdm.read('file.nc')
>>> print f.equals(f)
True
>>> f[:, 0] = -99
>>> cfdm.write(f, 'newfile.nc')
```

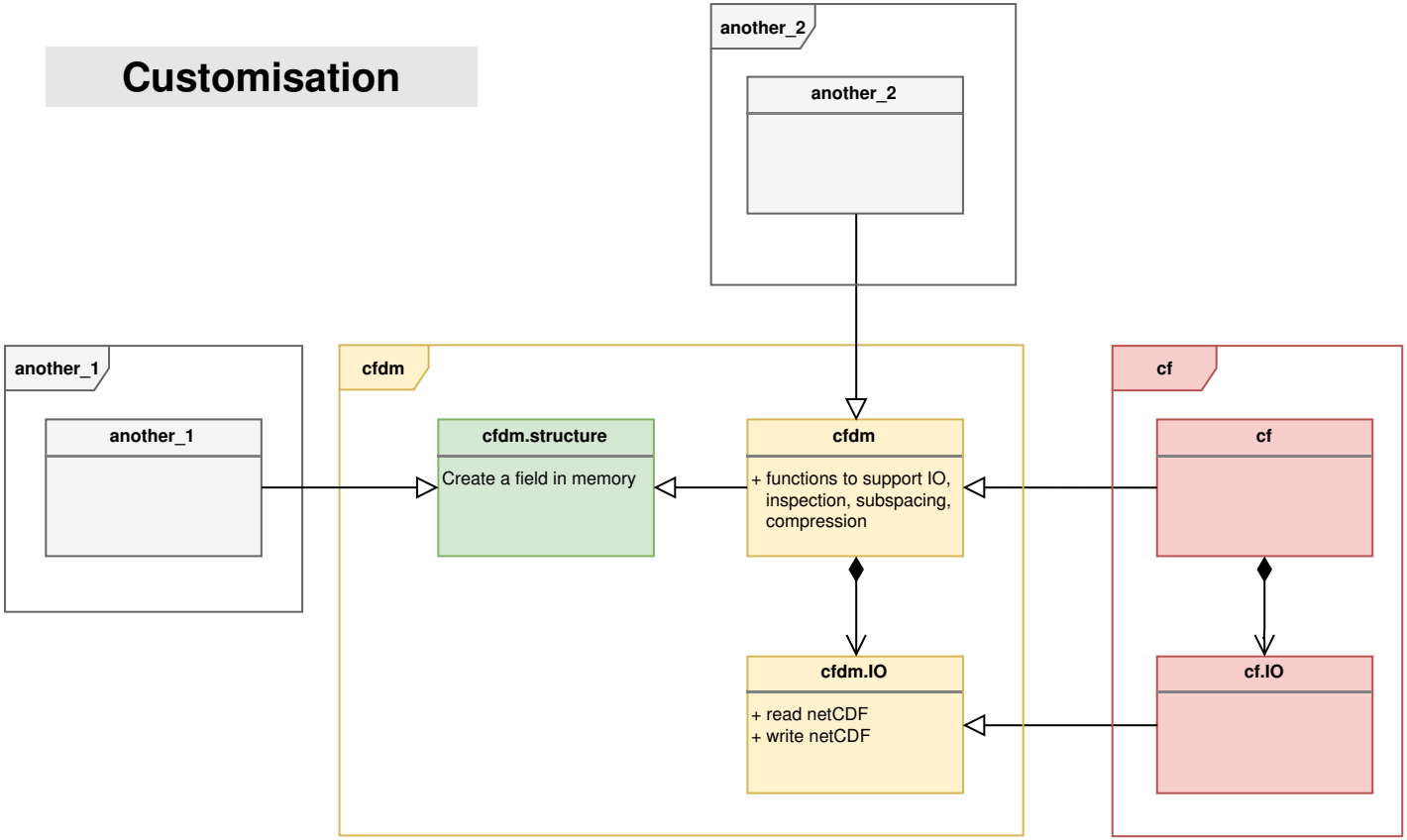


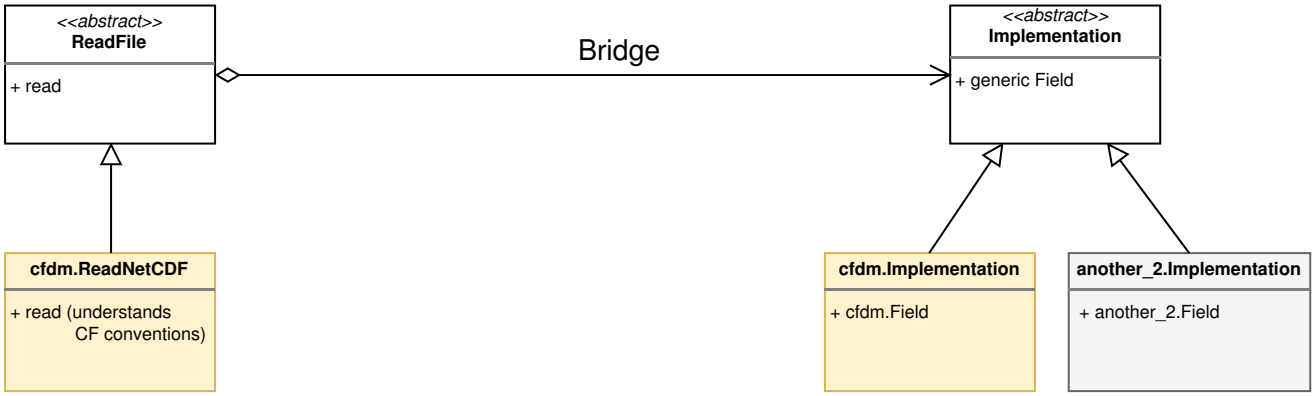

```
>>> import cfdm
>>> implementation = cfdm.Implementation(Field=cfdm.Field)
>>> netcdf = cfdm.ReadNetCDF(implementation)
>>> f = netcdf.read('myfile.nc')
>>> print f
[<Field: eastward_wind(height(1), grid_latitude(10), grid_longitude(9)) m
s-1>,
 <Field: air_temperature(grid_latitude(10), grid_longitude(9)) K>]
>>> type(f[0])
```

Bridge pattern:

Decouple an abstraction from its implementation so the two can vary independently

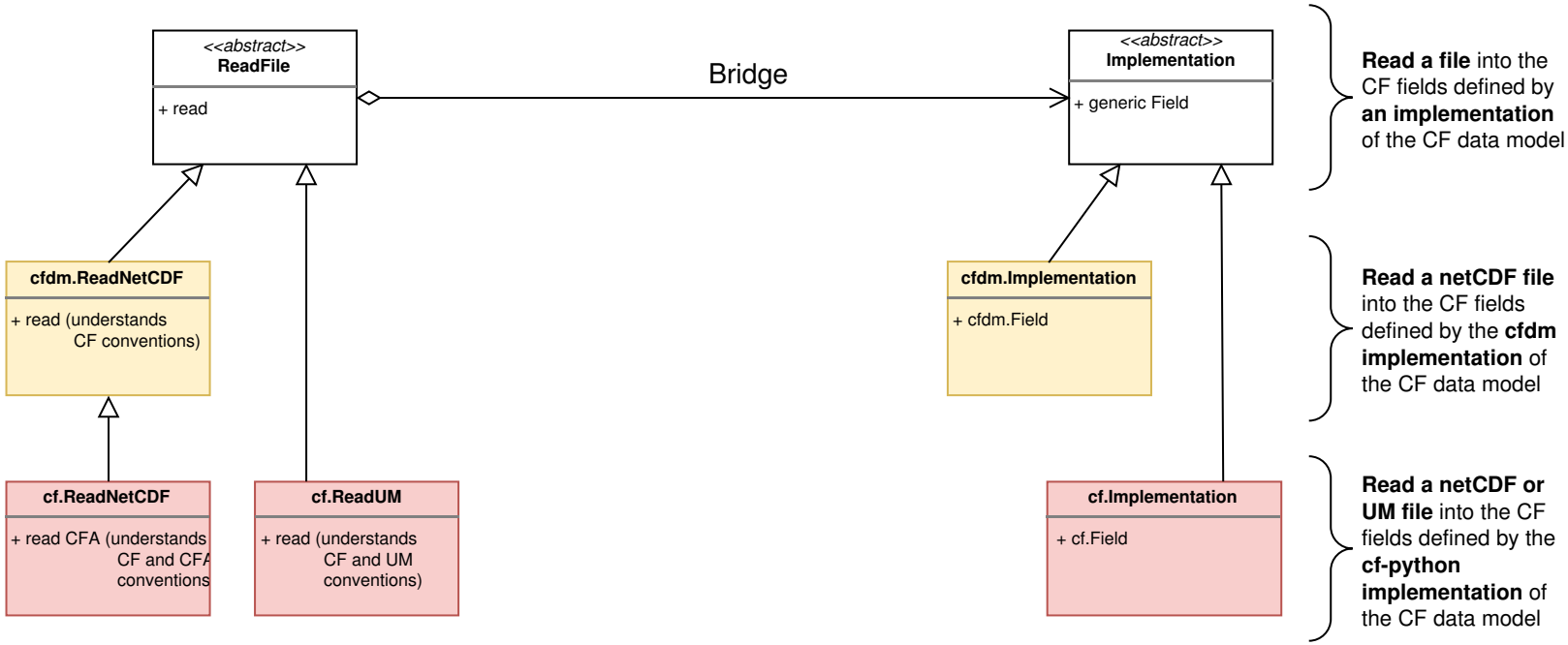
Customisation





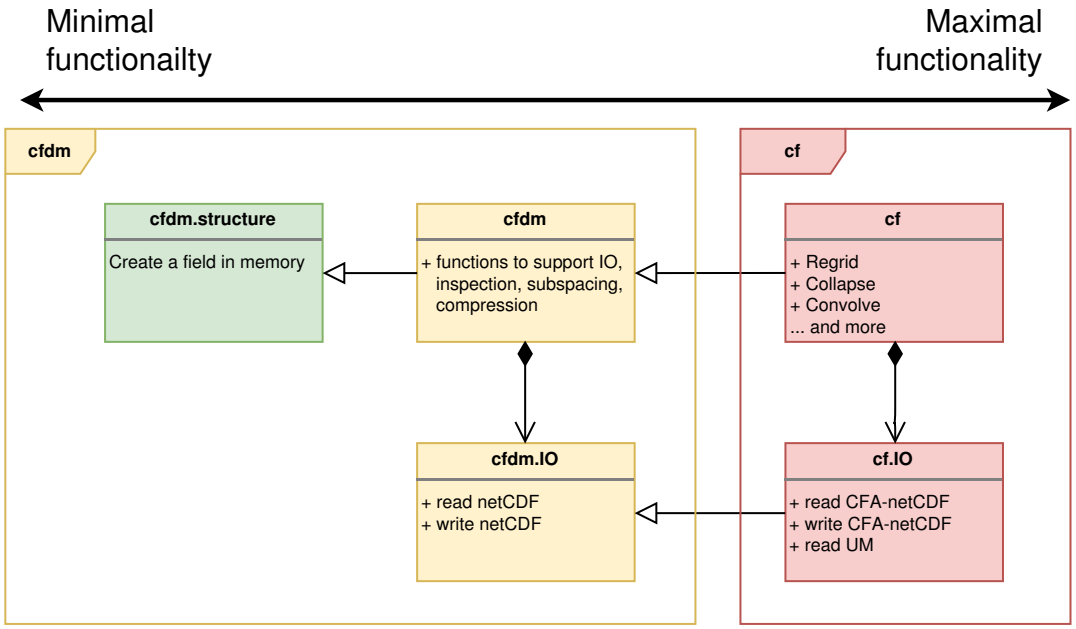

```
>>> import cfdm
>>> implementation = cfdm.Implementation(Field=cfdm.Field)
>>> netcdf = cfdm.ReadNetCDF(implementation)
>>> f = netcdf.read('myfile.nc')
>>> print f
[<Field: eastward_wind(height(1), grid_latitude(10), grid_longitude(9)) m
s-1>,
 <Field: air_temperature(grid_latitude(10), grid_longitude(9)) K>]
>>> type(f[0])
```

```
>>> import cfdm
>>> import another_2
>>> implementation = another_2.Implementation(Field=another_2.Field)
>>> netcdf = cfdm.ReadNetCDF(implementation)    # SAME
>>> f = netcdf.read('myfile.nc')                # SAME
>>> print f
[<Field: eastward_wind(height(1), grid_latitude(10), grid_longitude(9)) m
s-1>,
 <Field: air_temperature(grid_latitude(10), grid_longitude(9)) K>]
>>> type(f[0])
```




```
>>> import cfdm
>>> implementation =
cfdm.Implementation(Field=cfdm.Field)
>>> netcdf = cfdm.ReadNetCDF(implementation)
>>> f = netcdf.read('myfile.nc')
>>> type(f[0])
```

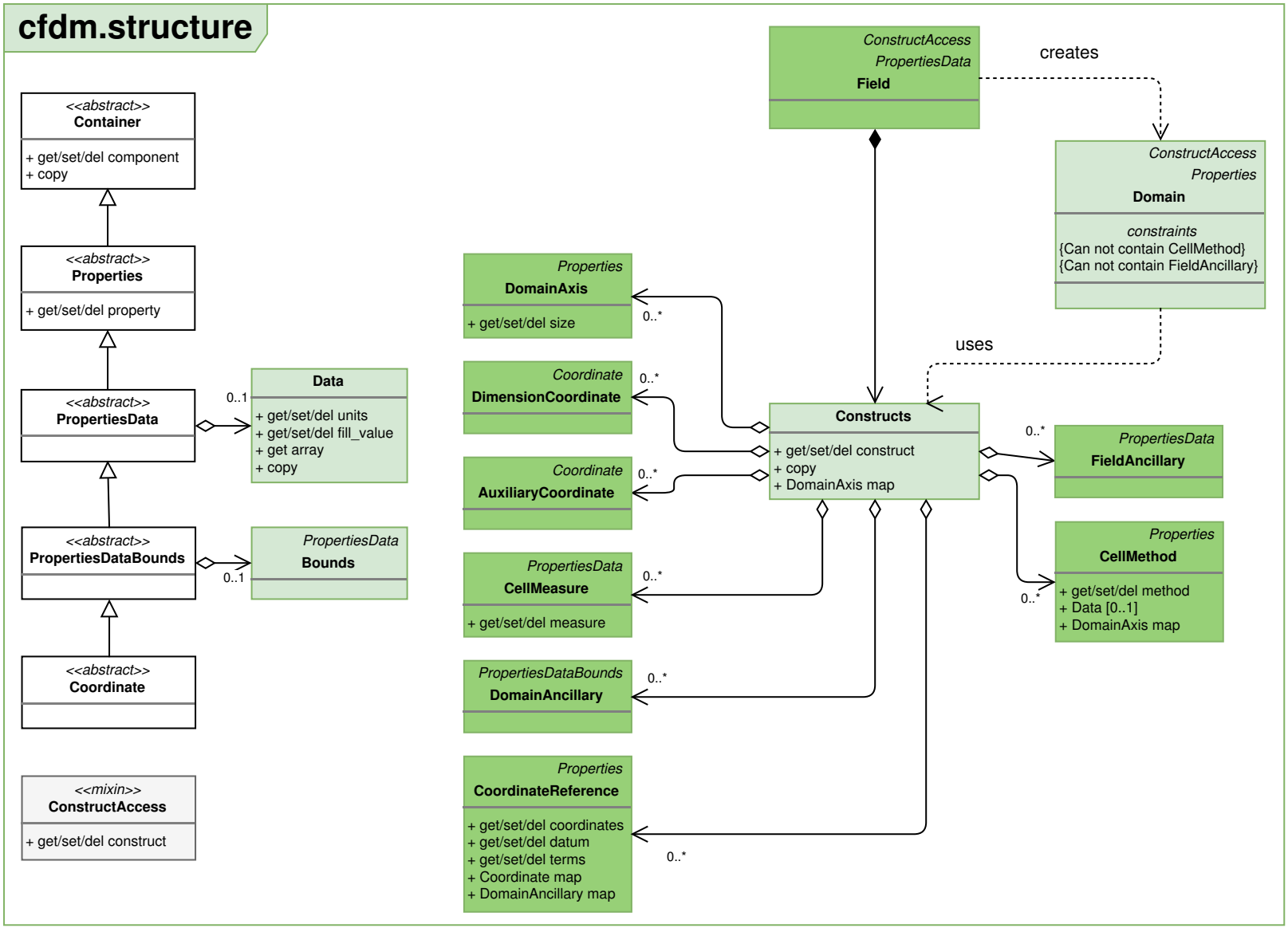
```
>>> import cf
>>> implementation = cf.Implementation(Field=cf.Field)
>>> netcdf = cf.ReadNetCDF(implementation)    #
DIFFERENT
>>> f = netcdf.read('myfile.nc')              # SAME
>>> type(f[0])
```

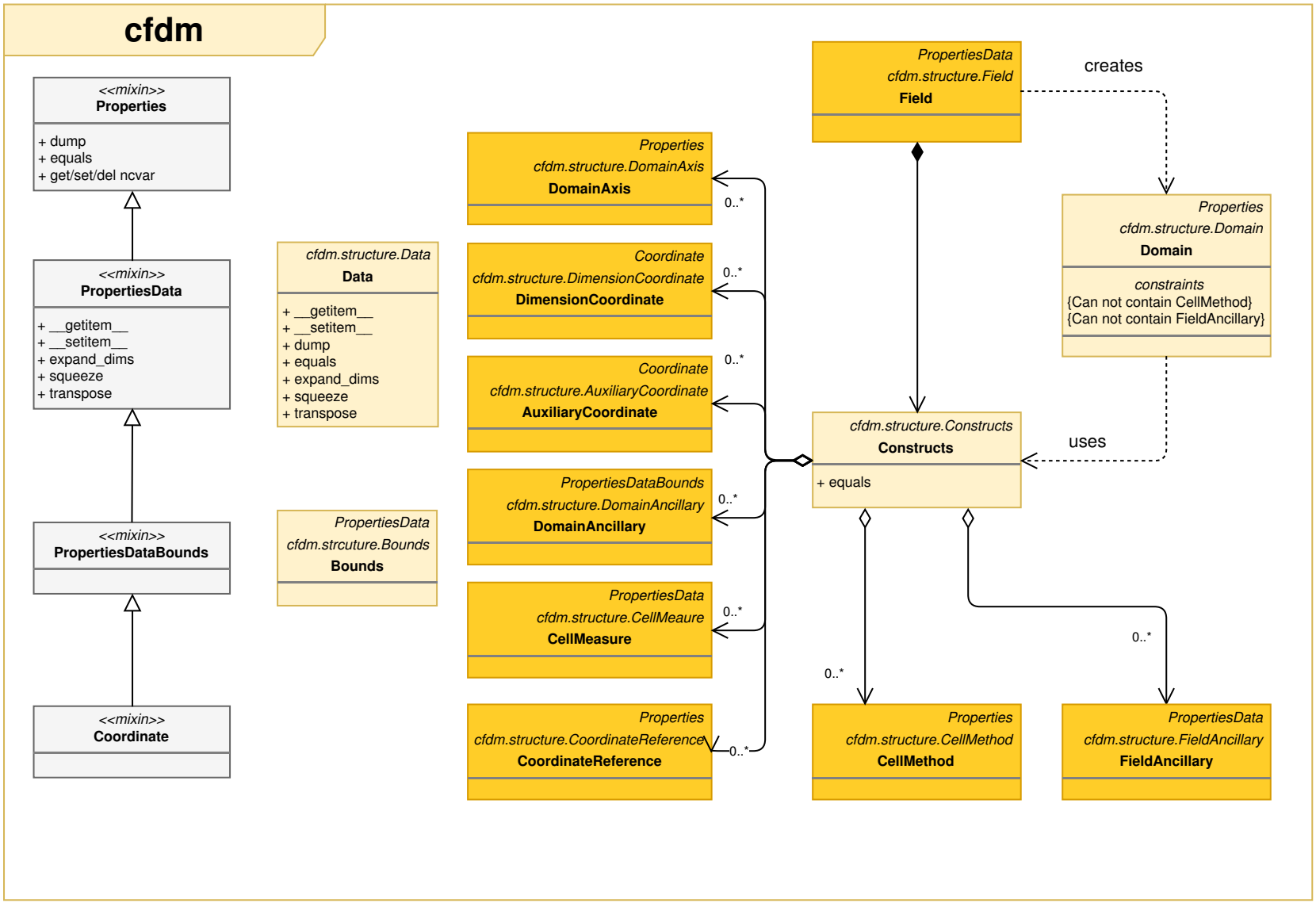


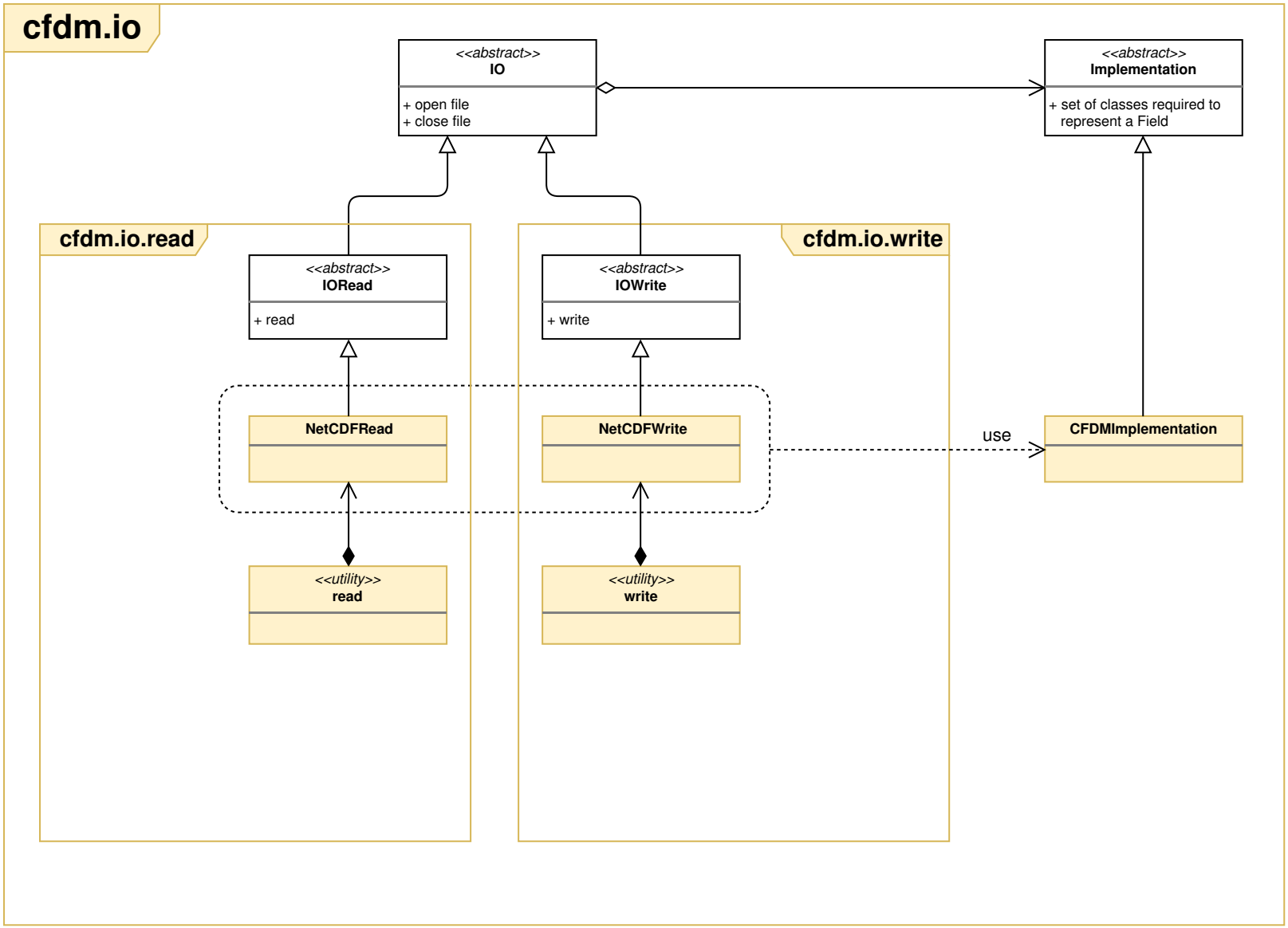

```
>>> import cfdm
>>> f = cfdm.structure.Field()
>>> f.set_property('standard_name', 'iron_growth_limitation_of_picophytoplankton'
>>> print f.get_property('standard_name')
'iron_growth_limitation_of_picophytoplankton'
```

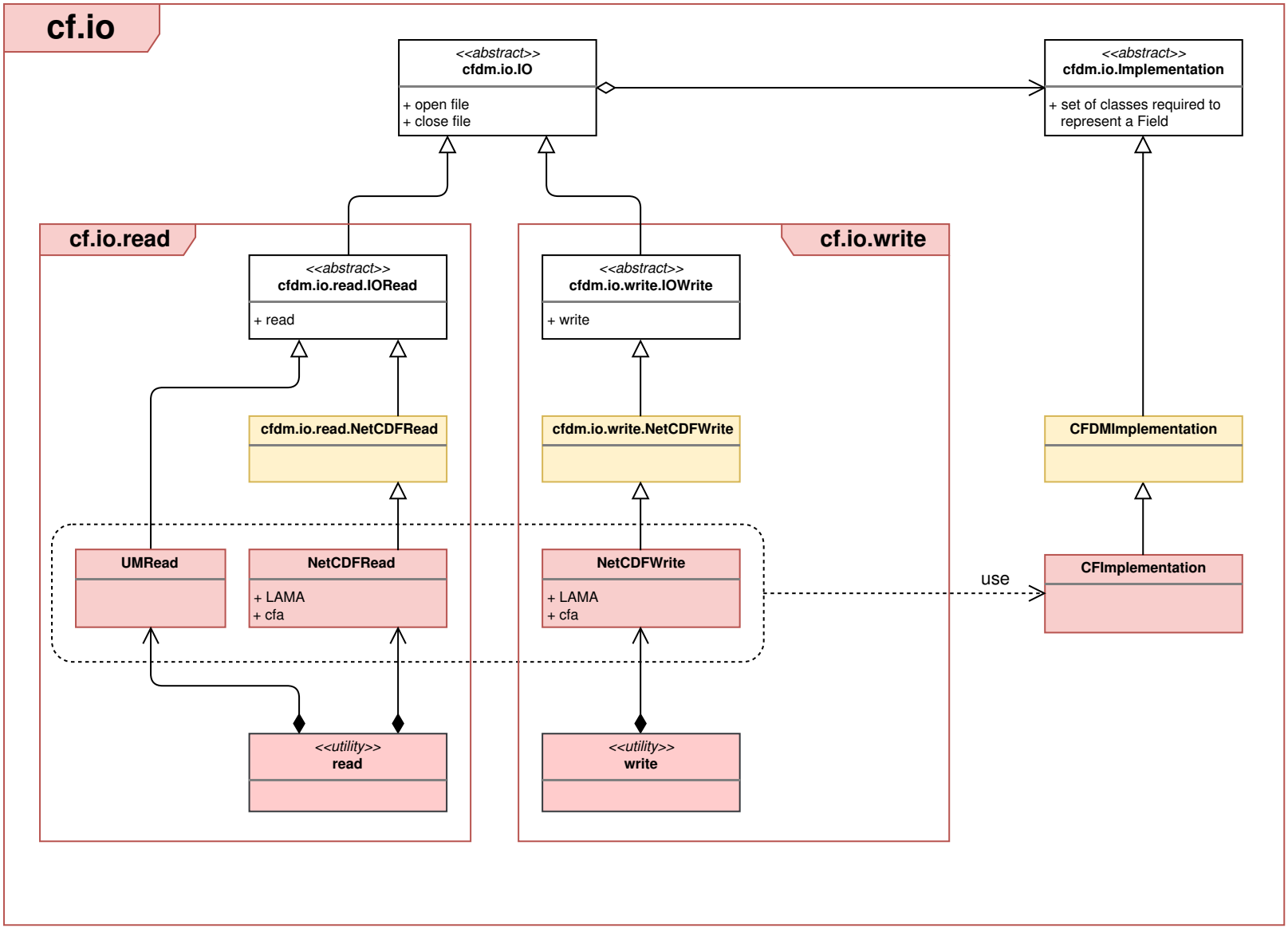
```
>>> import cfdm
>>> f = cfdm.read('file.nc')
>>> print f.equals(f)
True
>>> f[:, 0] = -99
>>> cfdm.write(f, 'newfile.nc')
```

```
>>> import cf
>>> f = cf.read('file[1-9].pp')
>>> f.standard_name = 'sea_water_practical_salinity'
>>> g = f.collapse('T: variance', weights='T')
>>> g = g.subspace(longitude=cf.wi(-10, 45))
>>> cf.write(g, 'newfile.nc', fmt='CFA4')
```


cfdm.structure
<<abstract>>
Container
+ get/set/del component
+ copy
<<abstract>>
Properties
+ get/set/del property
<<abstract>>
PropertiesData
<<abstract>>
PropertiesDataBounds
<<abstract>>
Coordinate
<<mixin>>
ConstructAccess
+ get/set/del construct
0..1
Data
+ get/set/del units
+ get/set/del fill_value
+ get array
+ copy
0..1
PropertiesData
Bounds
ConstructAccess
PropertiesData
Field
creates
ConstructAccess
Properties
Domain
constraints
{Can not contain CellMethod}
{Can not contain FieldAncillary}
uses
Properties
DomainAxis
+ get/set/del size
0..*
Coordinate
DimensionCoordinate
0..*
Coordinate
AuxiliaryCoordinate
0..*
PropertiesData
CellMeasure
+ get/set/del measure
0..*
PropertiesDataBounds
DomainAncillary
0..*
Properties
CoordinateReference
+ get/set/del coordinates
+ get/set/del datum
+ get/set/del terms
+ Coordinate map
+ DomainAncillary map
0..*
Constructs
+ get/set/del construct
+ copy
+ DomainAxis map
0..*
PropertiesData
FieldAncillary
0..*
Properties
CellMethod
+ get/set/del method
+ Data [0..1]
+ DomainAxis map


cfdm
<<mixin>>
Properties
+ dump
+ equals
+ get/set/del ncvar
<<mixin>>
PropertiesData
+ __getitem__
+ __setitem__
+ expand_dims
+ squeeze
+ transpose
<<mixin>>
PropertiesDataBounds
<<mixin>>
Coordinate
cfdm.structure.Data
Data
+ __getitem__
+ __setitem__
+ dump
+ equals
+ expand_dims
+ squeeze
+ transpose
PropertiesData
cfdm.strcuture.Bounds
Bounds
Properties
cfdm.structure.DomainAxis
DomainAxis
Coordinate
cfdm.structure.DimensionCoordinate
DimensionCoordinate
Coordinate
cfdm.structure.AuxiliaryCoordinate
AuxiliaryCoordinate
PropertiesDataBounds
cfdm.structure.DomainAncillary
DomainAncillary
PropertiesData
cfdm.structure.CellMeaure
CellMeasure
Properties
cfdm.structure.CoordinateReference
CoordinateReference
0..*
PropertiesData
cfdm.structure.Field
Field
creates
Properties
cfdm.structure.Domain
Domain
constraints
{Can not contain CellMethod}
{Can not contain FieldAncillary}
cfdm.structure.Constructs
Constructs
+ equals
uses
Properties
cfdm.structure.CellMethod
CellMethod
PropertiesData
cfdm.structure.FieldAncillary
FieldAncillary


cfdm.io
<<abstract>>
IO
+ open file
+ close file
<<abstract>>
Implementation
+ set of classes required to represent a Field
cfdm.io.read
cfdm.io.write
<<abstract>>
IORead
+ read
<<abstract>>
IOWrite
+ write
NetCDFRead
NetCDFWrite
use
CFDMImplementation
<<utility>>
read
<<utility>>
write


cf.io
<<abstract>>
cfdm.io.IO
+ open file
+ close file
<<abstract>>
cfdm.io.Implementation
+ set of classes required to represent a Field
cf.io.read
cf.io.write
<<abstract>>
cfdm.io.read.IORead
+ read
<<abstract>>
cfdm.io.write.IOWrite
+ write
cfdm.io.read.NetCDFRead
cfdm.io.write.NetCDFWrite
CFDMImplementation
UMRead
NetCDFRead
+ LAMA
+ cfa
NetCDFWrite
+ LAMA
+ cfa
use
CFImplementation
<<utility>>
read
<<utility>>
write

```
class DimensionCoordinate(MixinCoordinate, cfdm.structure.DimensionCoordinate):
```