DESY IT Seminar

HDF5, Nexus, and what it is all about

Eugen Wintersberger
HDF5 and Nexus
DESY IT, 27.05.2013

HELMHOLTZ ASSOCIATION

DESY

Why should we care about Nexus and HDF5?

Current state:

Data is stored either as

- ➔ ASCII file
- ➔ Binary image formats

```
!
! Comments
%c
  filename  = va696a_00001
  date      =  4-Nov-2009 time     = 00:45:42
  counter   = Mythen Integral
  scan mot  = diff17;  XS
!
! Parameter
%p
 SAMPLE_TIME = 1
 .... data omitted ...
!
! Data
%d
 Col 1 diff17[°/mm] FLOAT
 Col 2 NOMOT[°/mm] FLOAT
 Col 3 NOMOT[°/mm] FLOAT
 ... data omitted ...
 -1.38990000  0.00000000 0.00000000 1550673000 1692857 9.16010e+02 4255
 -1.33990000  0.00000000 0.00000000 1658145000 1810183 9.16010e+02 4403
 -1.28990000  0.00000000 0.00000000 1636303000 1786338 9.16010e+02 4421
... data omitted ...
```

There is a solution to this problem:

The synchrotron community agreed to store data as Nexus files with HDF5 as a physical storage format.

DESY

HDF5
an introduction

DESY

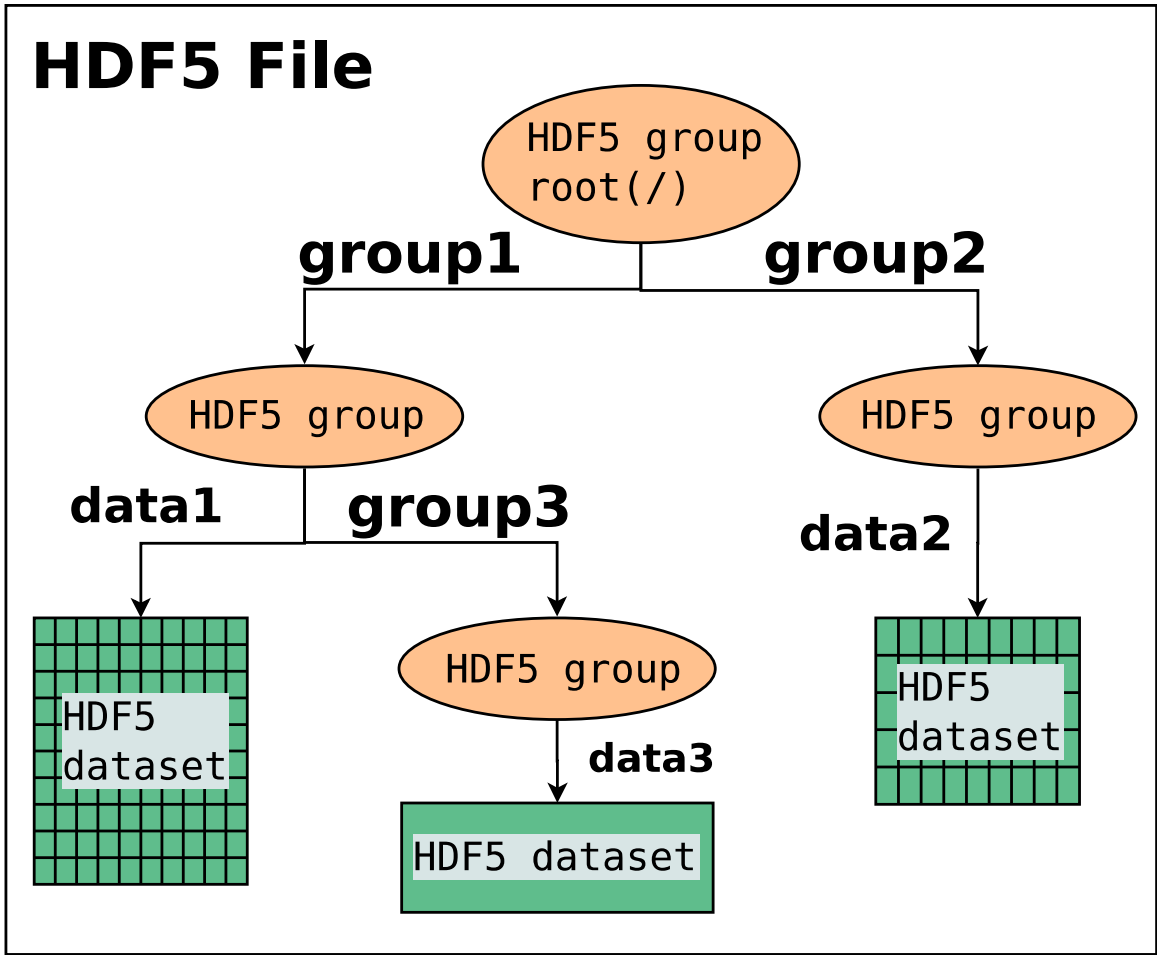
HDF5 – a kind of binary XML

HDF5 – a binary file format accessed via a C library!

- Groups to organize data
- Datasets for storing all kind of data
- Attributes to attach metadata to objects
- Platform independent data types
- Links to avoid data duplication
- Transparent big-/little-endian and data type conversion
- Partial I/O (read only parts of a data set)
- Apply filters to data for compression

The basic concept of HDF5 ...

Data is organized in a tree like structure:

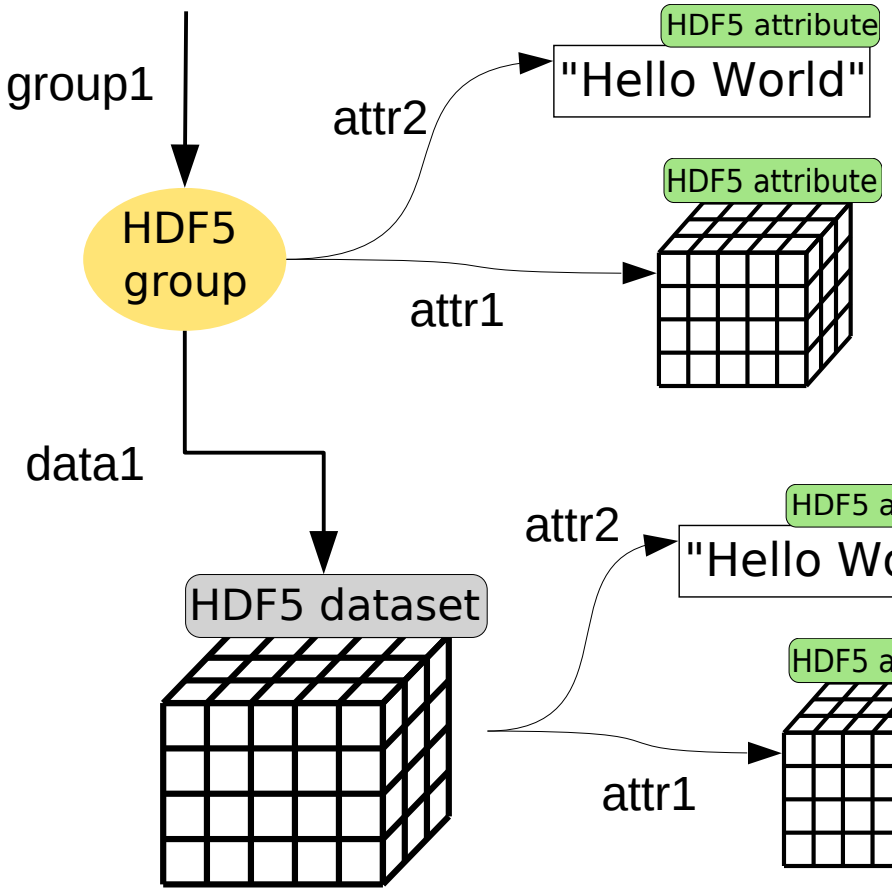


Objects are addressed via paths: `/group1/group3/data3`

DESY

Attributes for metadata ...

Attributes can be attached to groups and datasets for storing additional metadata

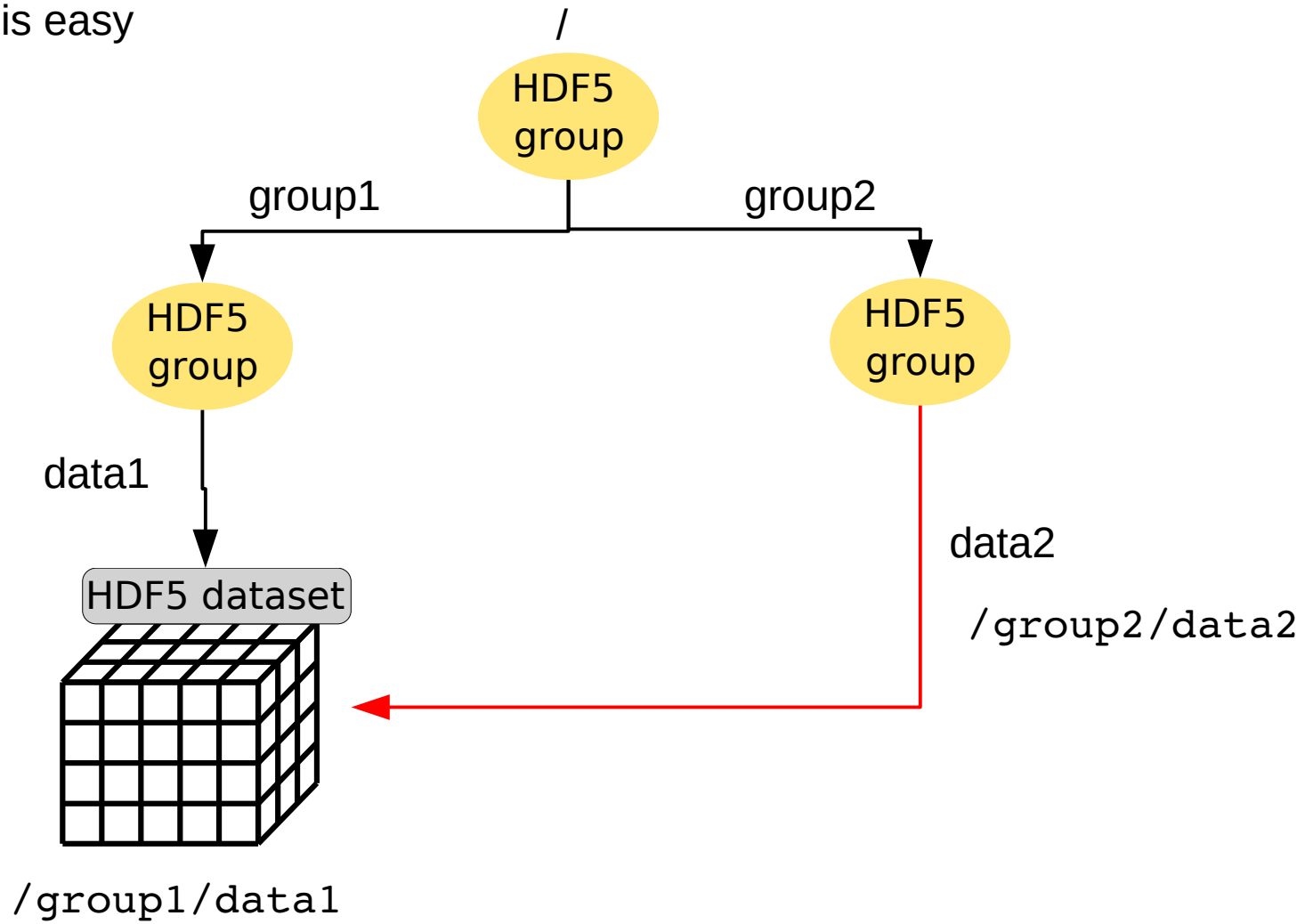


Attributes behave pretty much like datasets, but

- ➔ No filters
- ➔ No partial IO

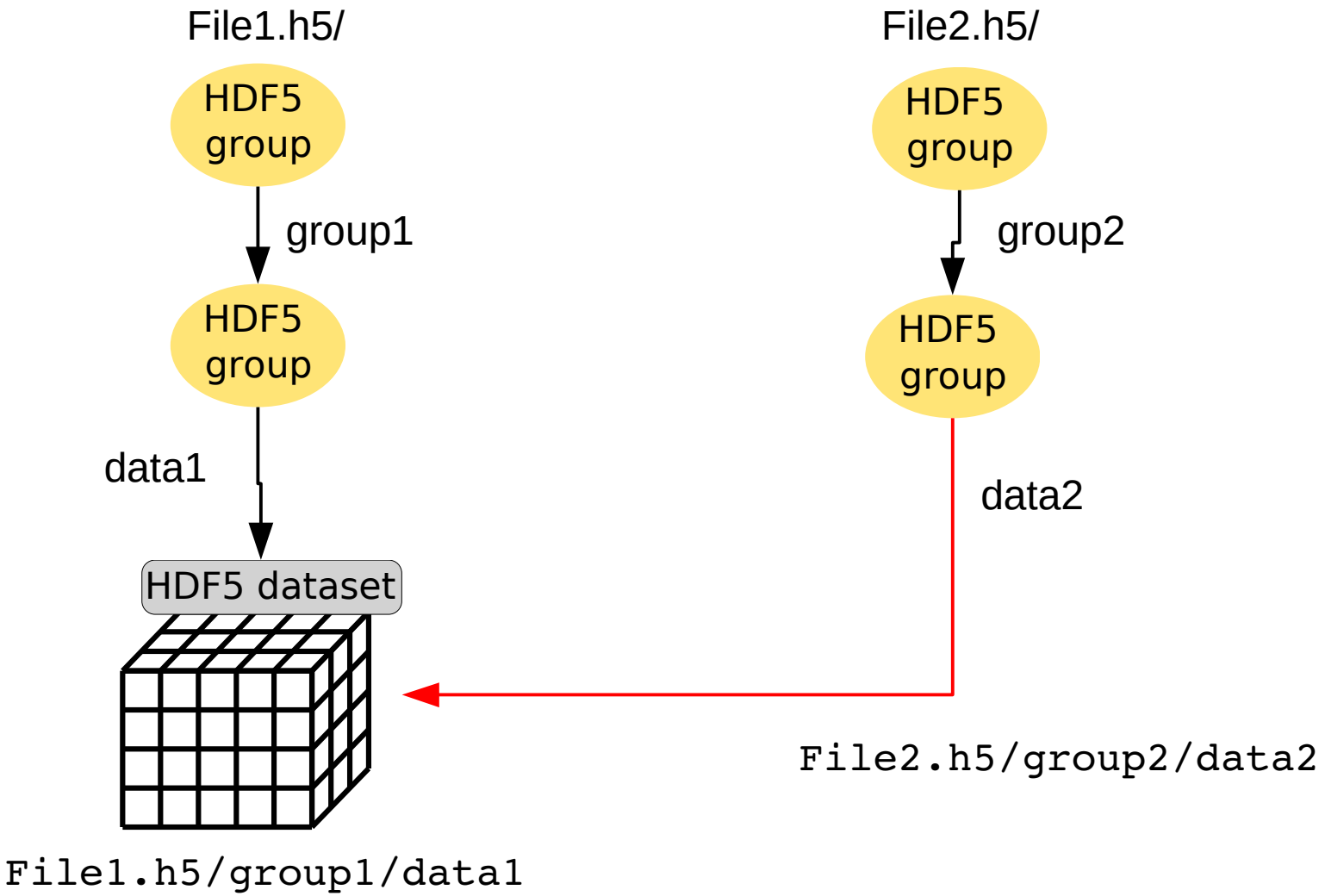
Internal links in HDF5 ...

As an HDF5 file can be considered as a bunch of nodes connected by graphs linking is easy



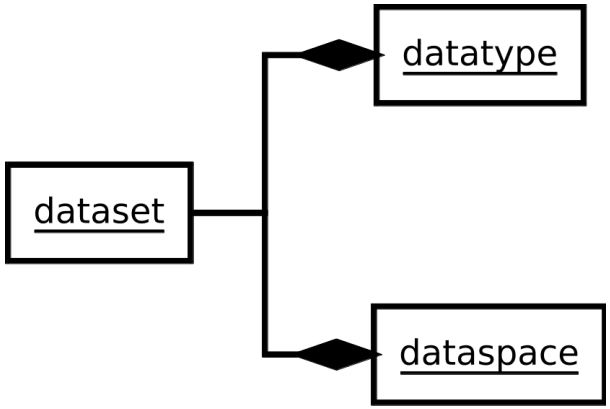
HDF5 and external links ...

Linking works also across file boundaries



HDF5 datasets
The core of HDF5 I/O

Anatomy of a dataset ...

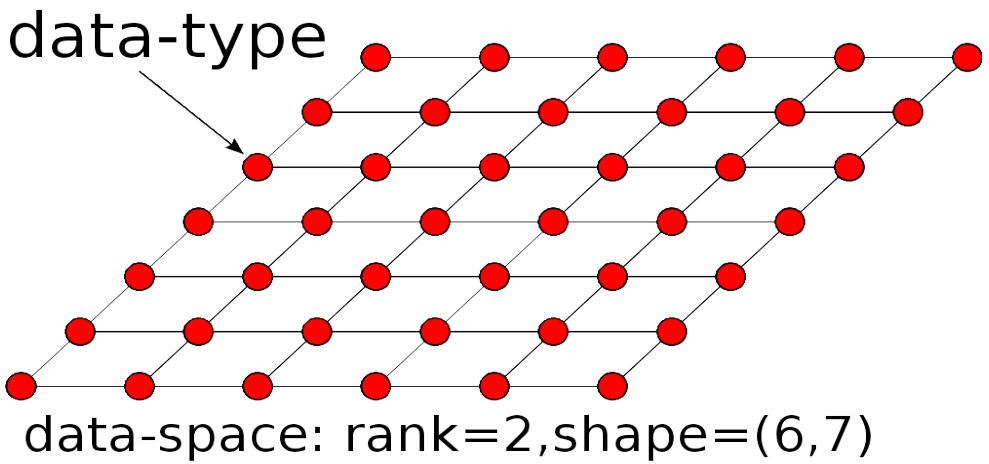


Datatype: what shall be stored.

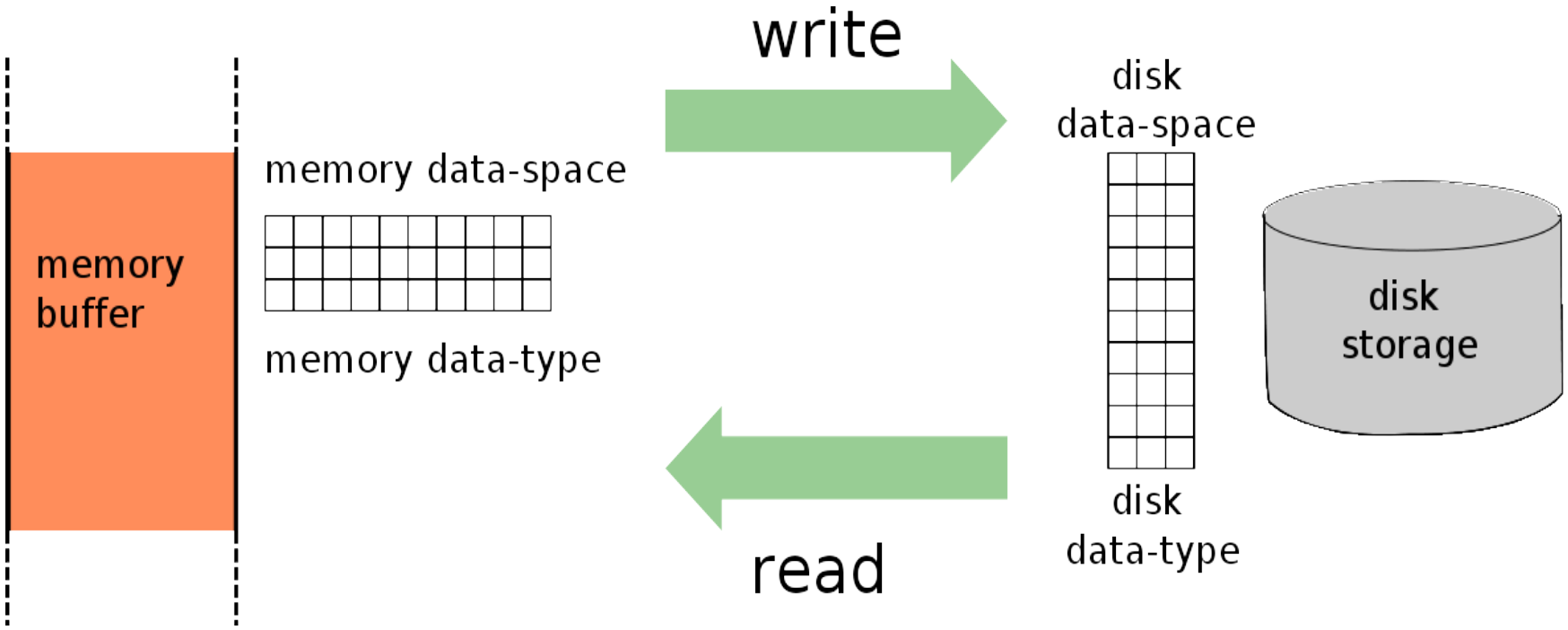
Dataspace: logical organization of the data.

Terminology:

- **rank**: number of dimensions
- **shape**: number of elements along each dimension.



Basic I/O with HDF5 ...



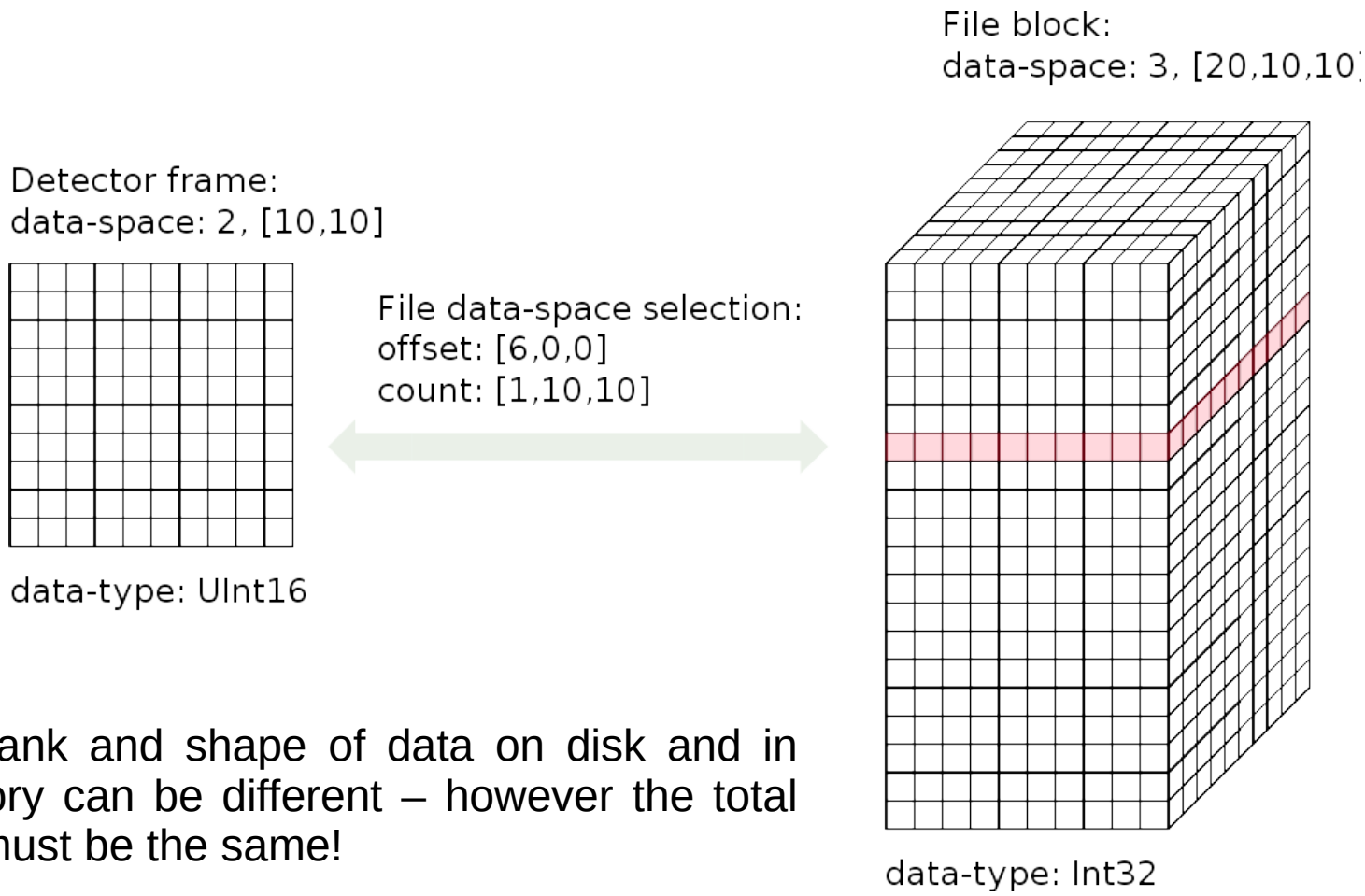
The HDF5 library takes care about:

- ➔Convert data type between memory and disk
- ➔Convert from little endian and big endian if required
- ➔Apply filters

DESY

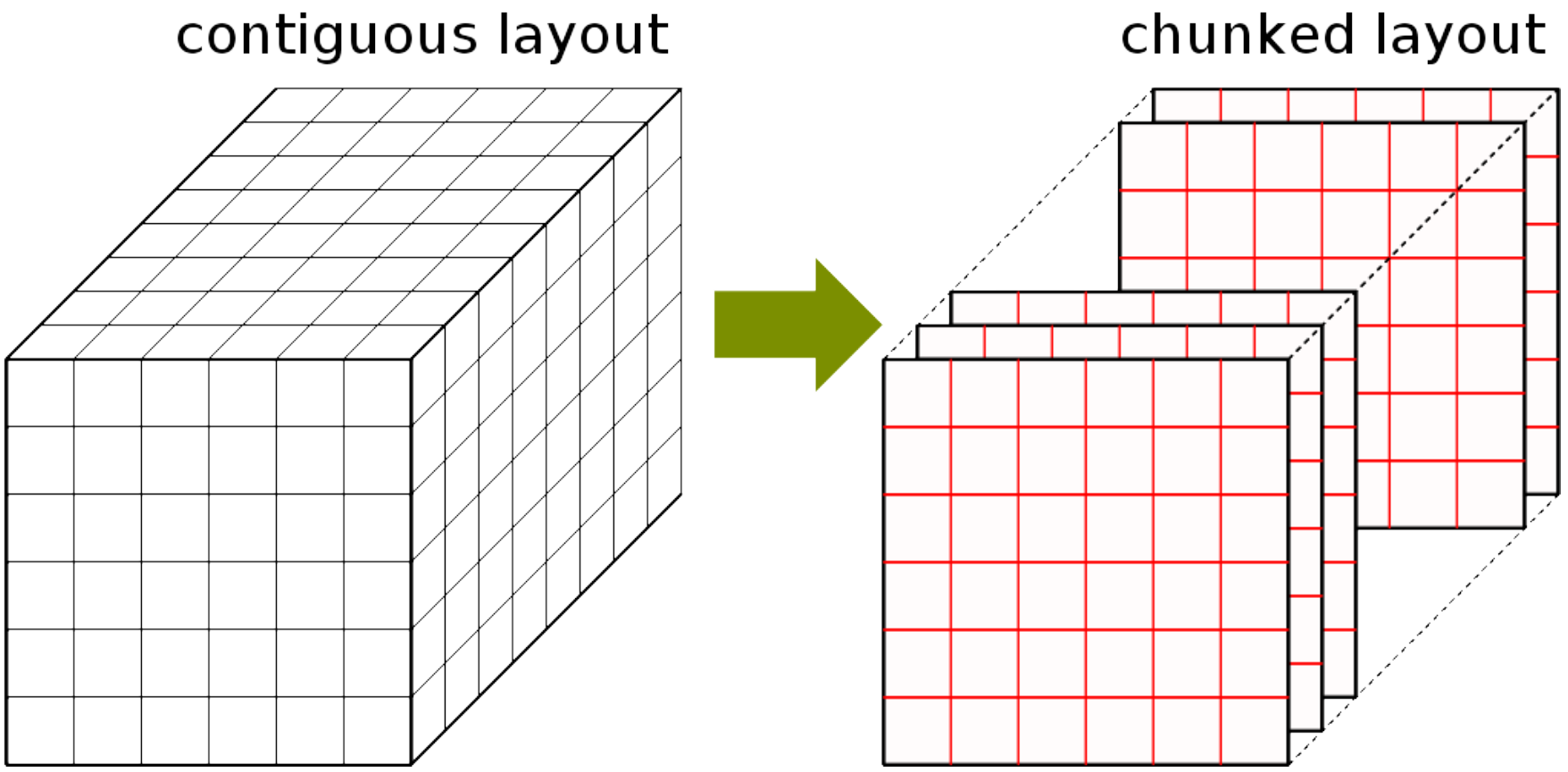
Partial I/O ...

Sometime one does not read/write all the data



The rank and shape of data on disk and in memory can be different – however the total size must be the same!

Chunks – reading and writing by parts ...



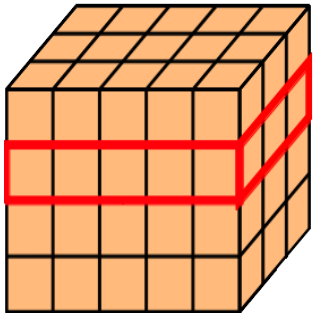
- All data is read or written
- Filters are applied to the entire data

- Data is read/written in portions of chunks
- Filters are applied to individual chunks

DESY

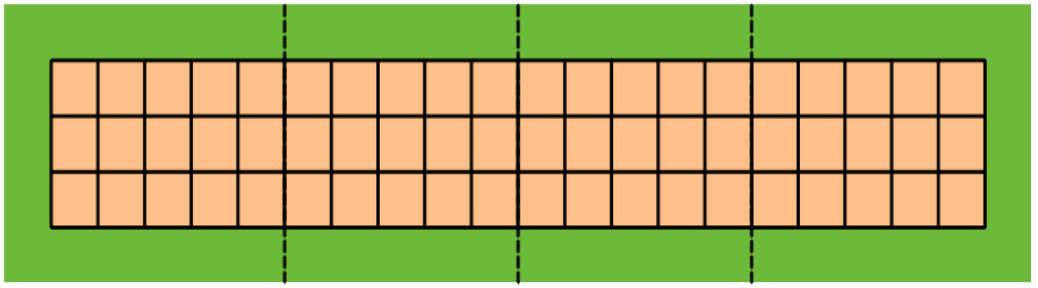
Chunked vs. contiguous layout on disk ...

data in memory



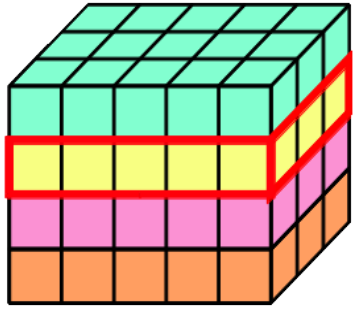
Contiguous layout

data on disk

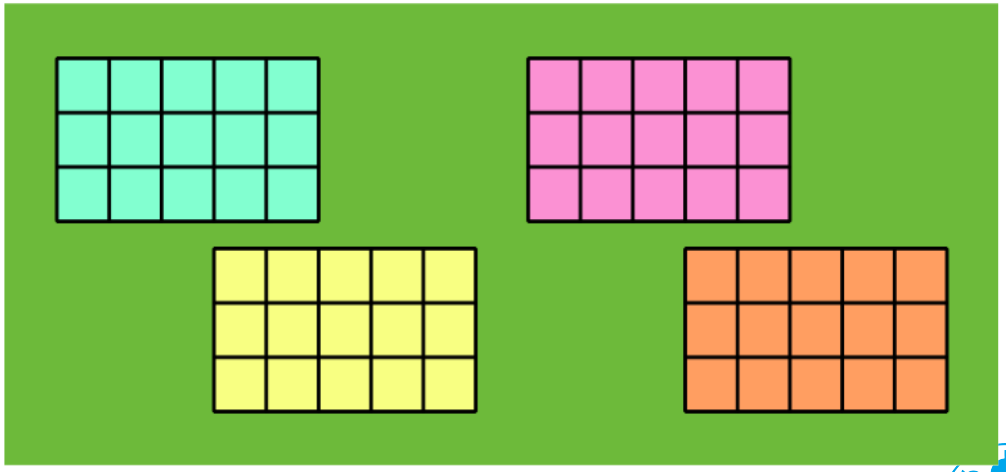


Chunked layout

data in memory



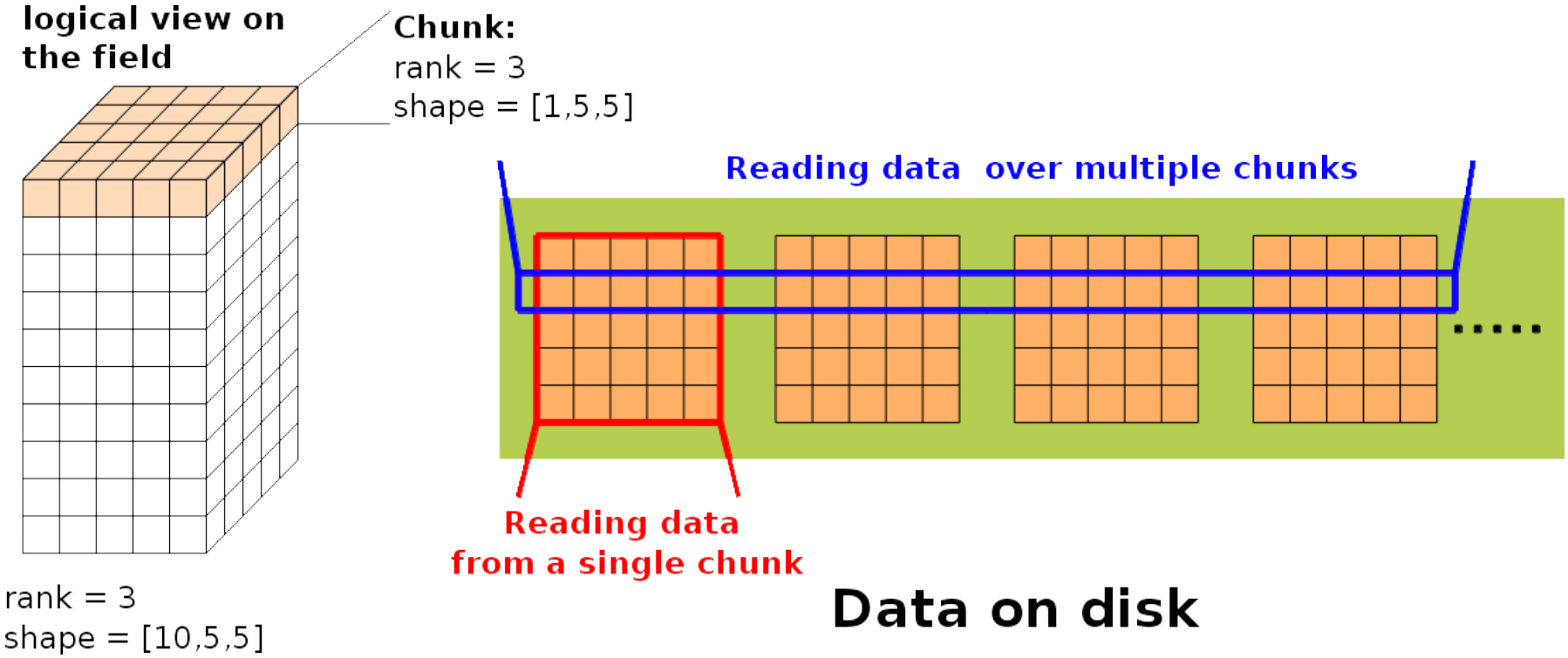
data on disk



DESY

Why should I care about chunking?

Chunking has serious effects on I/O performance



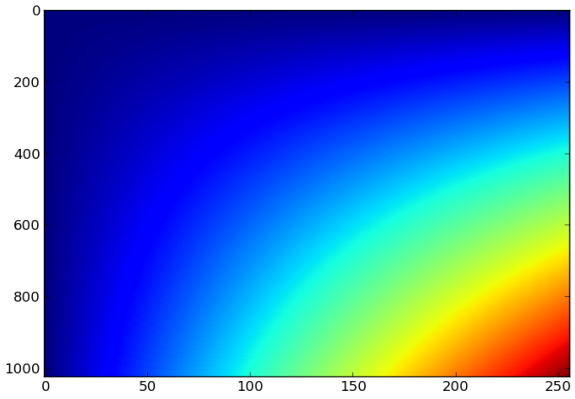
Access to elements within a single chunk is fast!

Filters are applied to individual chunks.

HDF5 – an example in Python

A simple example

```
 1 import numpy
 2 import h5py
 3 from matplotlib import pyplot
 4
 5 f = h5py.File("test.h5",'w')
 6
 7 #create objects
 8 data_g = f.create_group("data")
 9 log_g  = f.create_group("log")
10
11 image  = data_g.create_dataset('frame',(1024,256),dtype="float64")
12 log = log_g.create_dataset('log',(1024,),
13                            dtype=h5py.special_dtype(vlen=str))
14
15 #write some data
16 log[0] = "hello"
17 log[1] = "world. This"
18 log[2] = "is a test"
19
20 x = numpy.arange(0,1024)
21 y = numpy.arange(0,256)
22 image[...] = x[:,numpy.newaxis]*y[numpy.newaxis,:]
23
24
25 #reading data back
26 pyplot.imshow(f['/data/frame'][...],aspect='auto')
27 pyplot.savefig('test.png')
28
29 f.close()
```



DESY

Managing attributes and links ...

```python
import numpy
import h5py


f = h5py.File("test.h5",'w')

#create objects
data_g = f.create_group("data")

#creating attributes
data_g.attrs["date"] = 'May 26,2013'
data_g.attrs["temperature"] = 1.234
data_g.attrs["velocity"] = numpy.array([1.2,0.3,4.])

for name,value in data_g.attrs.iteritems():
    print "{0} = {1}".format(name,value)

f.close()
```

```
date = May 26,2013
temperature = 1.234
velocity = [ 1.2  0.3  4. ]
```

```python
import numpy
import h5py

f = h5py.File("test.h5",'w')

data_g = f.create_group("data")
data   = data_g.create_dataset("velocity",(3,),dtype="float64")
data[...] = numpy.array([1,2,3])
f.close()

f = h5py.File("test1.h5",'w')
f["velocity"] = h5py.ExternalLink("test.h5","/data/velocity")
print f["velocity"][...]

f.close()
```

```
[ 1.  2.  3.]
```


DESY

Language support ...

Bindings are provided for the following programming languages

- C (native language for the HDF 5 library)
- C++
- Fortran (90, 95, 2003)
- Java
- Python (h5py, pytables)
- .NET
- Perl
- R

Software support

- Matlab
- Mathematica
- IDL

Current limitations and new projects

Current limitations of HDF5 ...

- ➔ Writing and reading concurrently does not work
- ➔ Thread safe implementation causes total IO serialization
- ➔ High performance MPI-IO works only with certain restrictions
- ➔ …

Some of these issues are attacked by current projects ...

DESY

Current projects to improve HDF5 ...

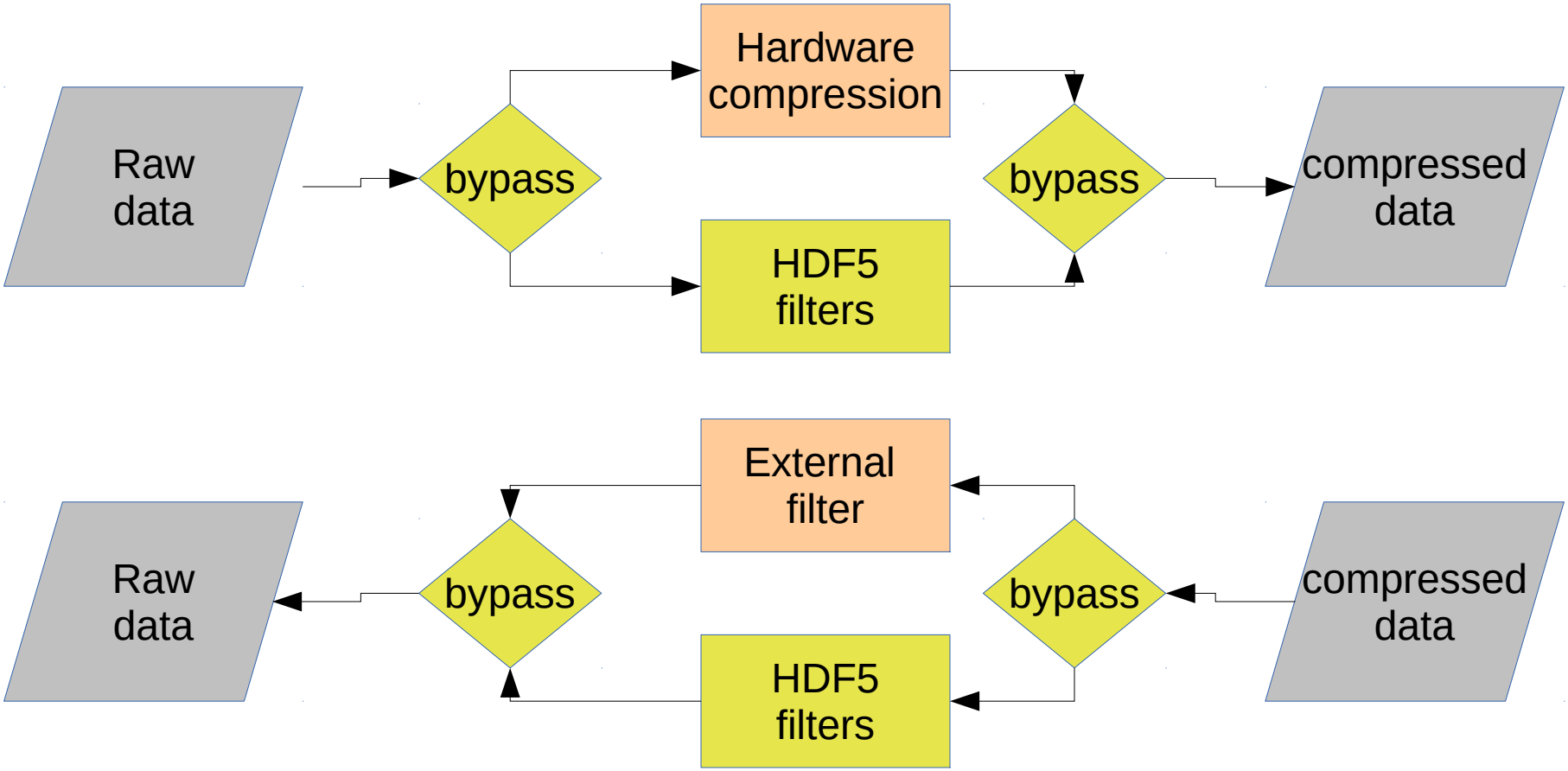
1. Bypassing the HDF5 filter chain (since 1.8.11)
2. Advanced external filter interface (since 1.8.11)
3. Single writer multiple reader support (currently negotiated)

1. and 2. have been funded by DECTRIS and DESY respectively

DESY

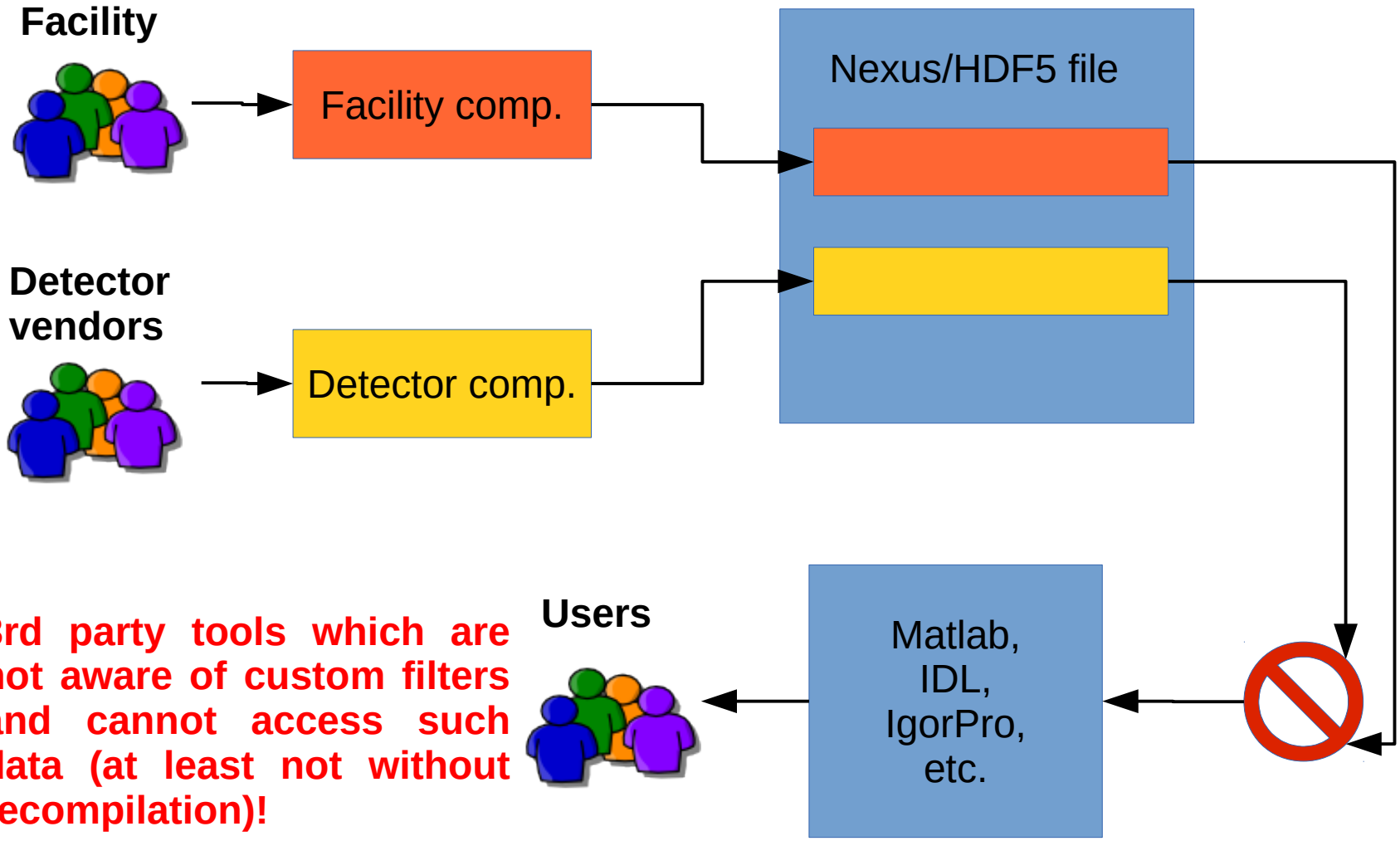
Bypass the HDF5 filter chain ...

Compression can sometimes done much better in hardware (FPGAs) → need to bypass the HDF5 filter chain!



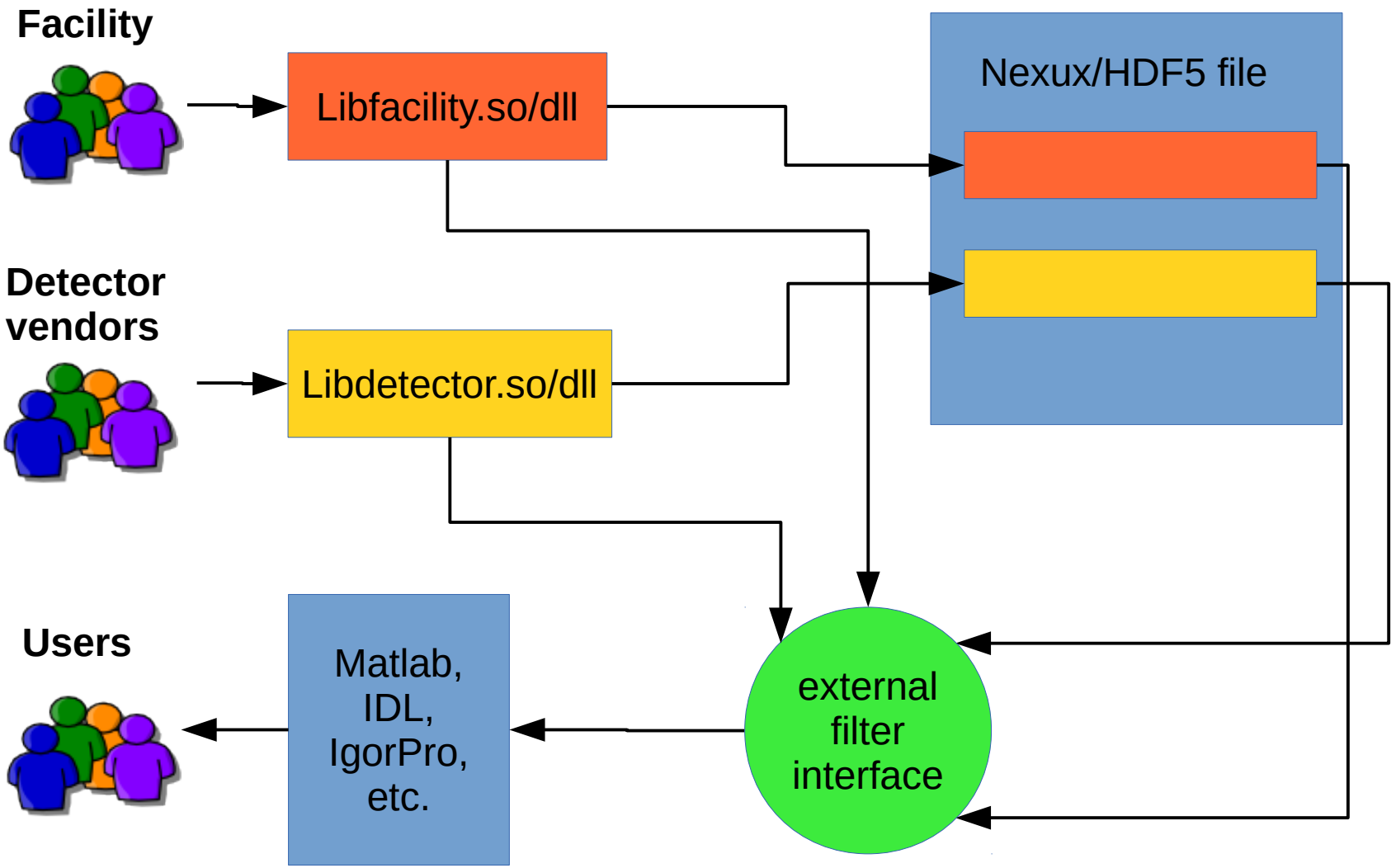
DESY

External filters cause problems for users ...



3rd party tools which are not aware of custom filters and cannot access such data (at least not without recompilation)!

Solution: interface for external filters ...



Nexus

add meaning to HDF5

Why Nexus?

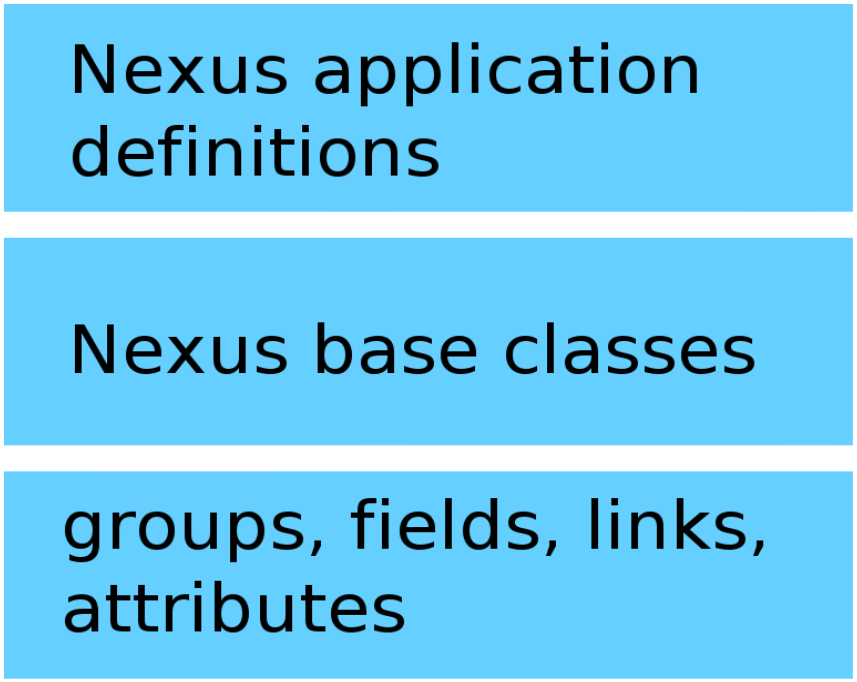
HDF5 is entirely agnostic about what data is stored!

Nexus is a set of rules how data should be organized in an HDF5 file.

Level 3: experimental technique specific definition

Level 2: individual beamline components

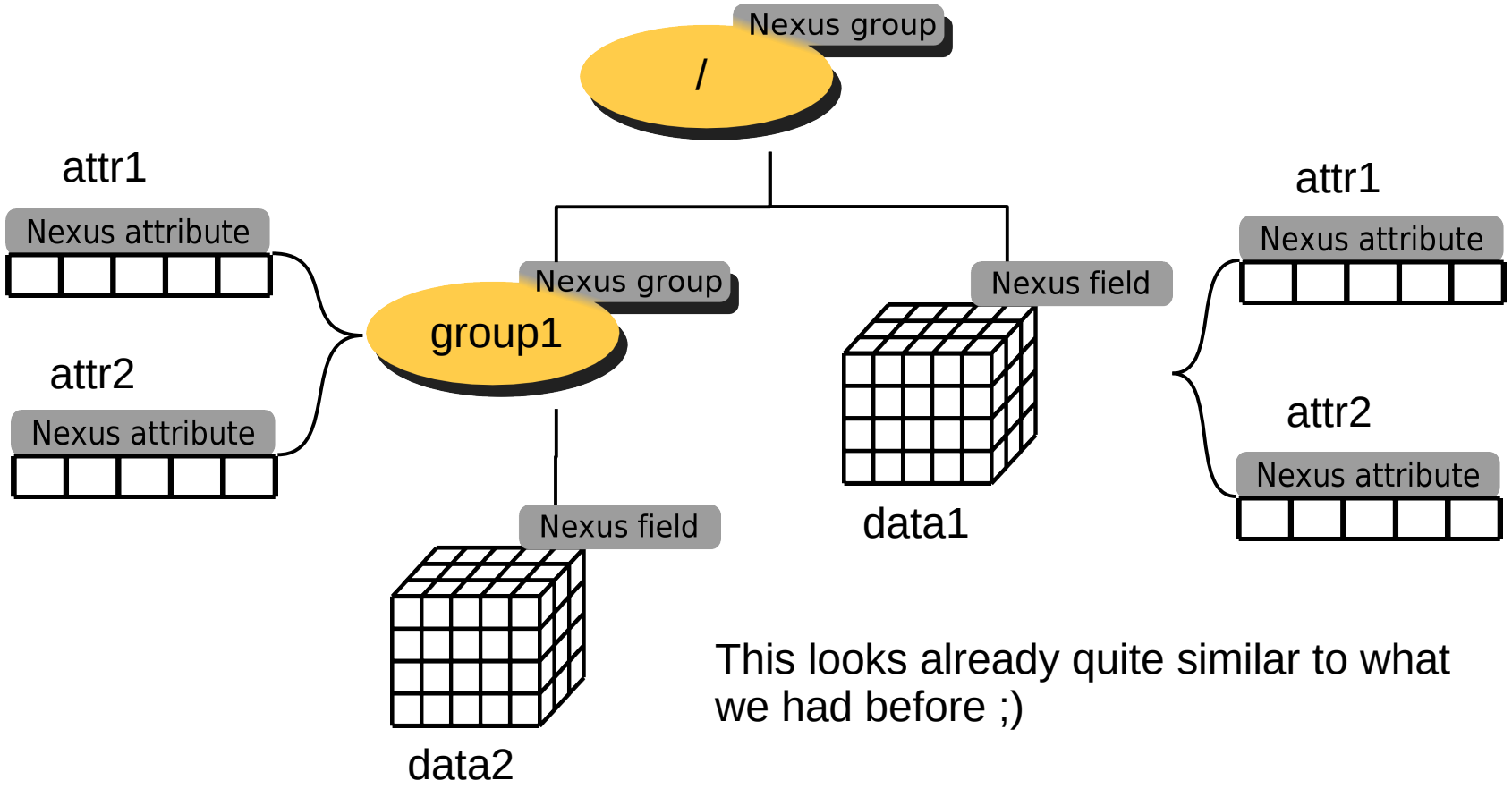
Level 1: basic objects



DESY

In Nexus data is organized in trees ...

Just start with the base classes:



This looks already quite similar to what we had before ;)

Nexus Definition Language (NXDL) – the key to semantics

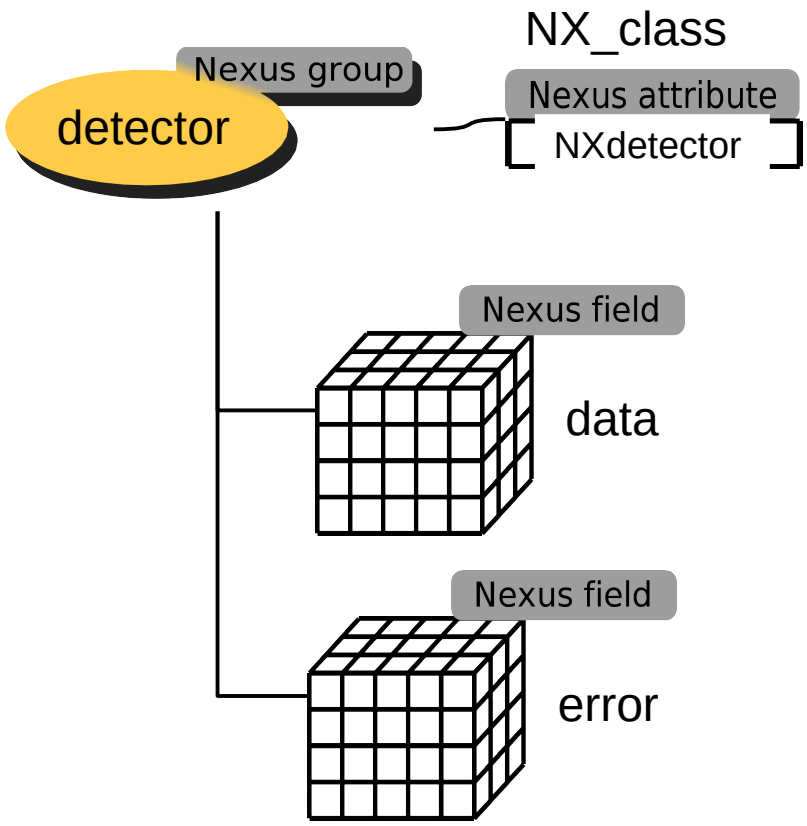
Smallest information unit: base class definied by an NXDL file.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="nxdlformat.xsl" ?>
…
<definition category="base" extends="NXobject" name="NXdetector"
            svnid="$Id$"
            type="group" version="1.0"
            xsi:schemaLocation="http://definition.nexusformat.org/nxdl/@NXDL_RELEASE@ ../nxdl.xsd"
            xmlns="http://definition.nexusformat.org/nxdl/@NXDL_RELEASE@"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:ns="http://definition.nexusformat.org/nxdl/@NXDL_RELEASE@"
            >
… code omitted …

  <doc>
    Template of a detector, detector bank, or multidetector.
  </doc>

  <field name="time_of_flight" type="NX_FLOAT" units="NX_TIME_OF_FLIGHT">
    <doc>Total time of flight</doc>

   … code omitted …
  </field>

  <field name="data" type="NX_NUMBER" units="NX_ANY">
    <doc>Data values from the detector.</doc>

   … code omitted …


  </field>
  … most of the code omitted …
</definition>
```


DESY

From NXDL to a base class tree ...

Representation of an NXDL file as a tree …. this is trivial.

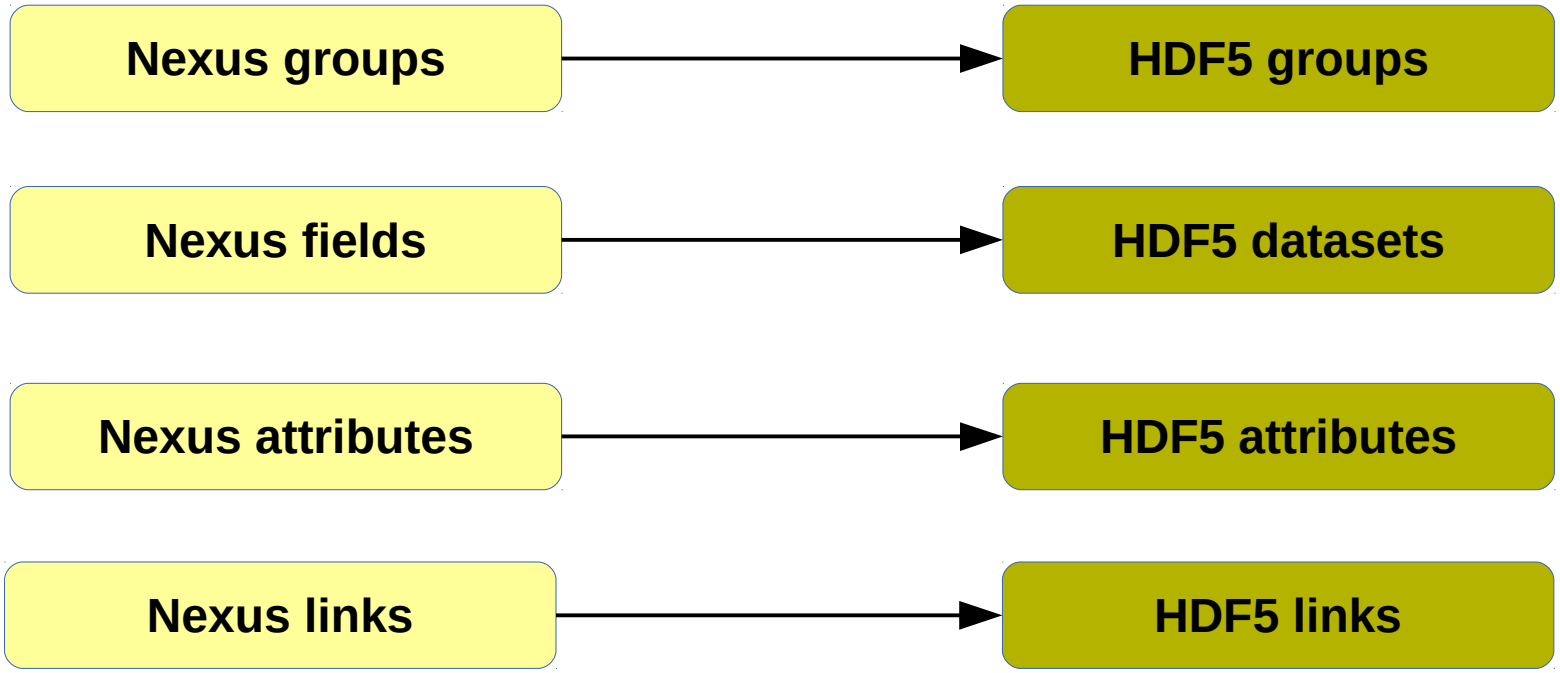


A group is turned into a class by setting the *NX_class* attribute.

Fields of standardized names and meaning within a class.

How to obtain a Nexus file ...

Nexus is just a set of rules how data should be organized → a physical file format is needed to store the data to disk!



This leads to a simple conclusion ...

A Nexus file IS an HDF5 file within which data is organized according to the Nexus rules!

DESY