HDRI - Work Package 1: Data Format and Data Management

Coordinator: Rainer Gehrke[1]

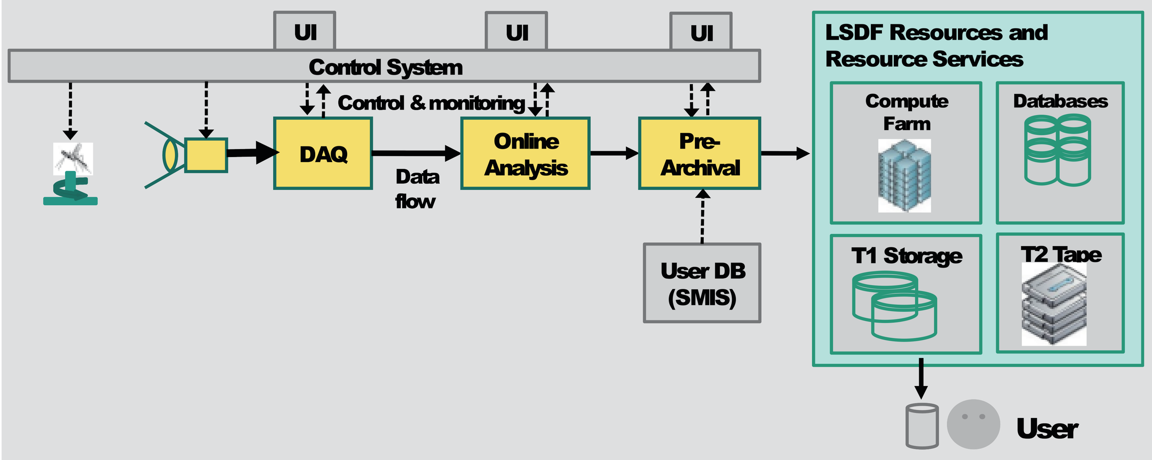
E. Wintersberger[1], H. Pasic[2], R. Gehrke[1]

[1] *DESY, Hamburg, Germany*
[2] *Karlsruher Institut fuer Technologie (KIT), Karlsruhe, Germany*

HDRI - High Data Rate Analysis and Processing

The application of 2D detectors in combination with high frame-rates and/or long frames-series increased the data-rates and data volumes involved in PNI experiments. The HDRI project tries to deal with this challenges along the entire data life-cycle.
The project is split into three work-packages:

- WP1 - data format and data management
- WP2 - real time data processing
- WP2 - data analysis, modelling and simulation



The heart of the WP-1 of the HDRI-project is a common data format based on HDF5 and Nexus. While HDF5 provides a simple way to access data in a binary format, Nexus adds the semantics. Nexus provides standard data fields with a predefined meaning as well as a well defined way how to structure a file.

HDF5: a new approach of storing data

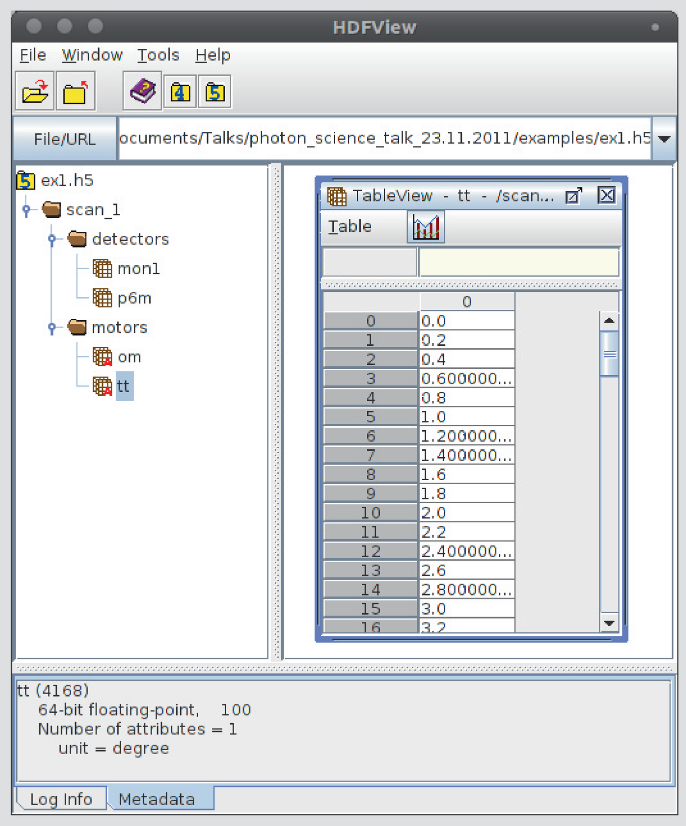
HDF5 is a data format whose features address a lot of the problems appearing at high data-rates

- binary format - fast enough to store large detector frames
- bindings to C/C++, FORTRAN, Java, Python, Matlab, IDL, etc.
- in-line compression of individual data-sets
- large user community (NASA,ESO, ...)

Data is organized like in a file system:

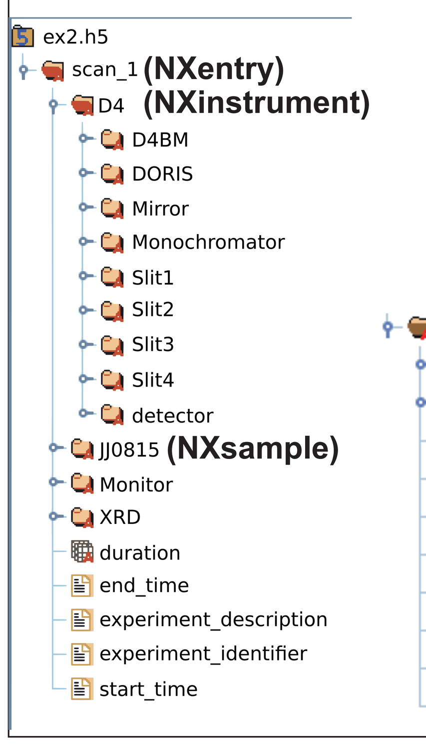
- **Groups** - Directories
- **Datasets** - Files
- **Attributes** for groups and data-sets
- **Links** to groups and data-sets within a file or a different file

HDF5 is the fundament of the common data format within the HDRI project.



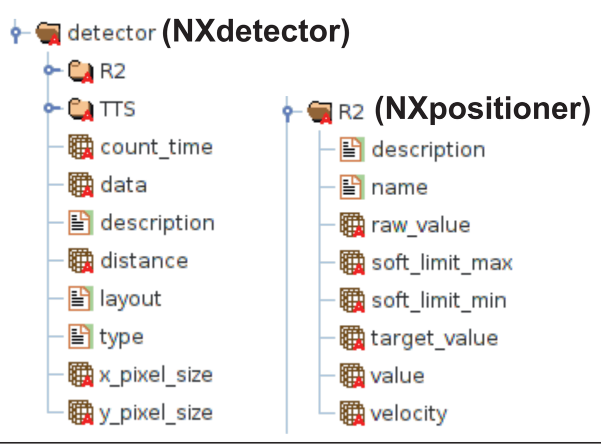
Nexus: making HDF5 aware of PNI

HDF5 has no special idioms to handle data from PNI experiments. The Nexus format provides conventions how to structure an HDF5 and rules how to store particular objects (motors, detectors, slits, etc.) that appear in an PNI experiment. These conventions and rules are summarized in the Nexus base classes (see Nexus manual on www.nexusformat.org).



Basic structure of a Nexus file (fundamental base classes):

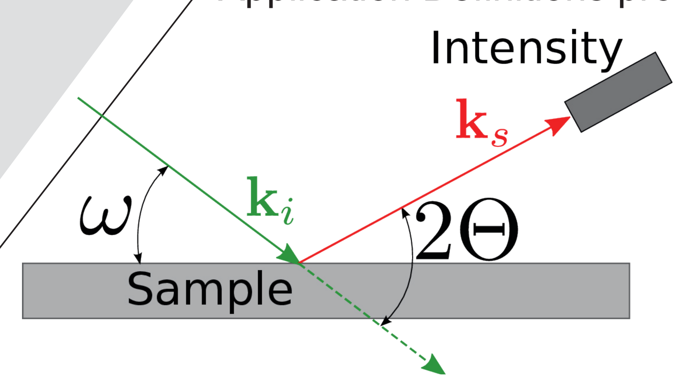
. **NXentry** - top level class for each scan/run
. **NXinstrument** - holds most of the beamline's components
. **NXsample** - sample environment

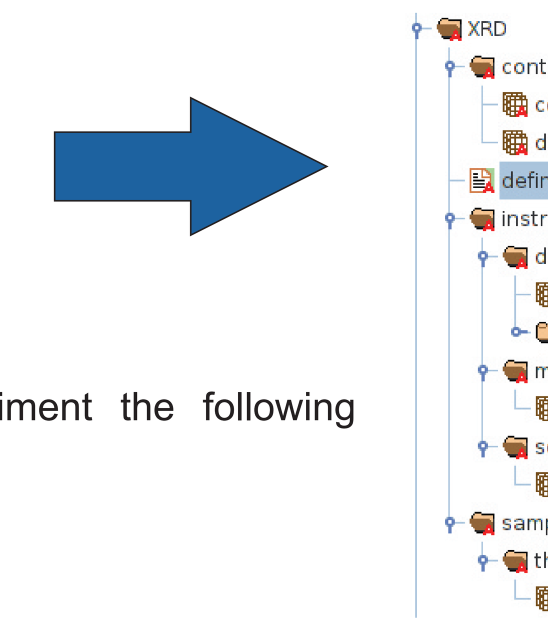


With base classes a complete description of the experiment can be stored along with the data in the same file.

Nexus Application Definitions

Application Definitions provide a method based view on the data.





For a simple XRD (as depicted below) experiment the following information is required to evaluate the data:

- 2Θ - the detector angle
- ω - angle of incidence
- the detector data
- the wavelength
- monitor data for normalization

Application Definitions provide shortcuts to method specific data!

Common Data-Format

HDF5 + Nexus

The heart of WP-1

HDF5 is easy to use

This simple example shows how to read data from an HDF5 file

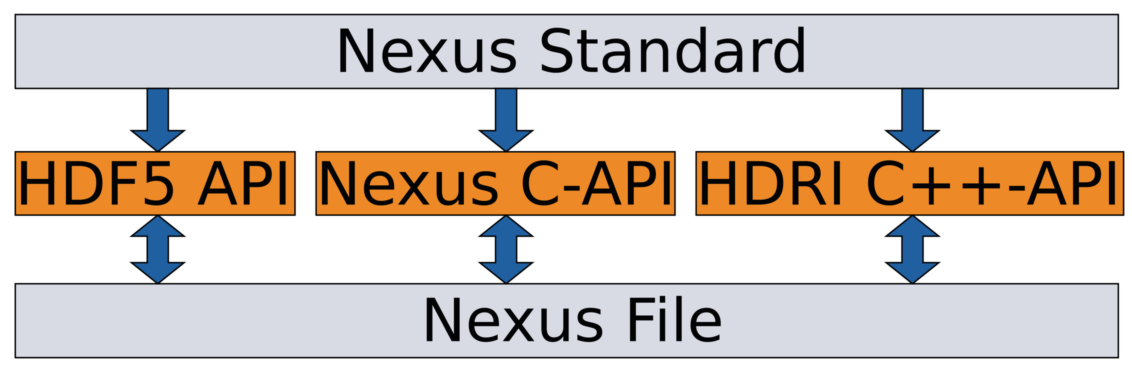
```
import h5py
import numpy

f = h5py.File("ex1.h5","w")
om = f["/scan_1/motors/om"][:]
omu = f["/scan_1/motors/om"].attrs["unit"]
tt = f["/scan_1/motors/tt"][:]

f.close()
```

As only native modules are used no additional code must be maintained in order to access data. HDRI will provide a C++ API for Nexus with HDF5 which adds Nexus semantics to HDF5 objects.

HDRI - Nexus/HDF5 API



Although there are several ways how to create a Nexus file the HDRI-project has decided to provide its own C++ API which has several advantages over the existing solutions:

- further simplifies the native HDF5 C or C++ API
- thread safe by nature (still under development)
- provides C++ features like iterators and template methods/functions
- provides archiving features not included in the standard Nexus C-API

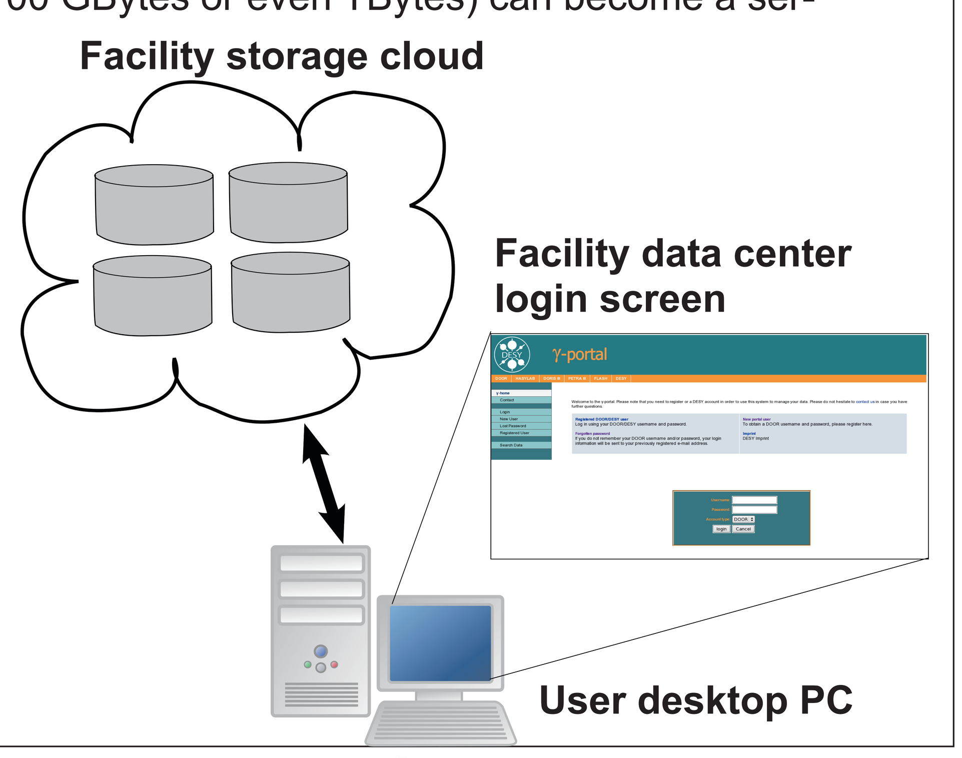
Data management - facility storage

Handling large amounts of data (in the order of 100 GBytes or even TBytes) can become a serious problem for users, in particular for the two following issues:

- transporting (moving data from the facility)
- safe storage of data (backup)

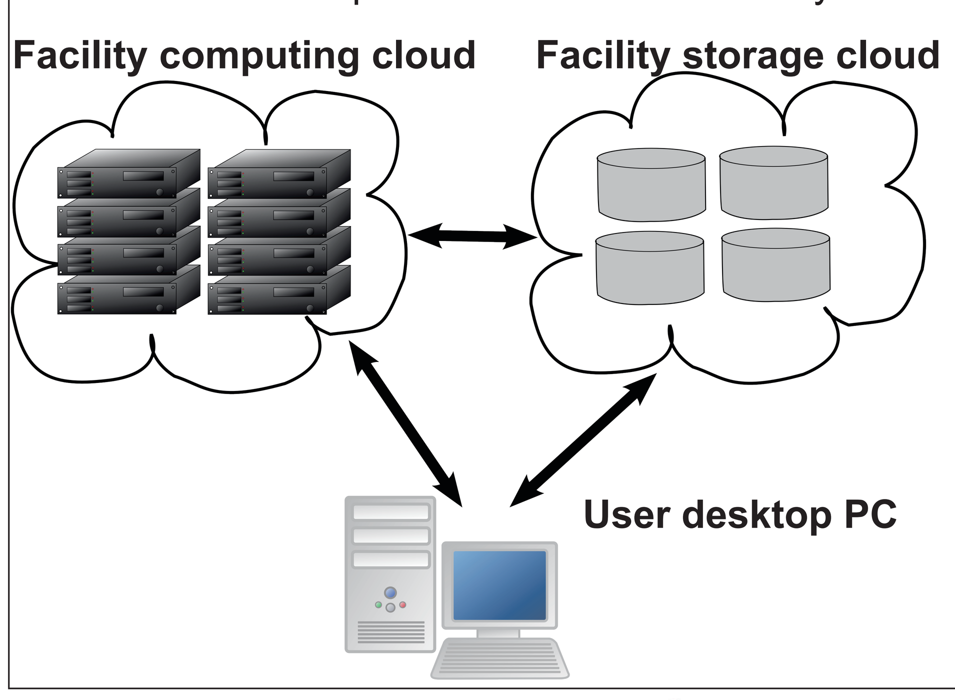
Thus facilities can provide storage services including:

- short time storage on disks (fast access)
- long time archiving on tapes (keep data safe)
- remote access for users
- search engines to look for particular datasets
- publishing data to the public domain
- make data cite-able



Remote data evaluation

Analysing large data-sets is nothing for the fainthearted. Not only appropriate computing resources must be provided but also the IO-system should perform seriously in order to prevent disk-IO from becoming the bottleneck of the entire analysis process. In particular for small research institutions it is difficult to provide such resources by themselves.



Within the HDRI project remote evaluation clusters are planned consisting of conventional computer hardware or even of GPUs. These hardware resources will be hosted by the research facilities involved in the HDRI project. Users can login from their home offices using a web interface and start evaluation runs on their data using standard analysis software installed on the clusters.