HDF5 @ SOLEIL

Stéphane POIRIER Raphaël GIRARDOT

Agenda

History of NeXus/HDF5 @ SOLEIL

Current status

Recording services

Data analysis services

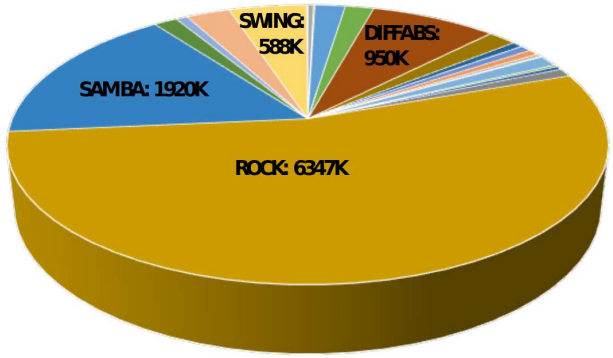
History of NeXus/HDF5 @ SOLEIL

- Up to a recent past, the Synchrotron scientific community (very conservative people!) used to write data in simple ASCII files.
- Before begining of operations in 2005 we take the decision to choose a data format that:
  - allow storing metadata along with experimental data
  - can deal with large data sets
  - accept any kind of data (n-Dimention, wide range of data types)
  - propose self-described data sets (data sets properties)
  - is efficient (obviously!)
- We looked at CBF, NetCDF, NeXus and choose this later with HDF5 as underlying physical format. Note that NeXus is just a nomenclature, a set of names related to some scientific domain of experiences.

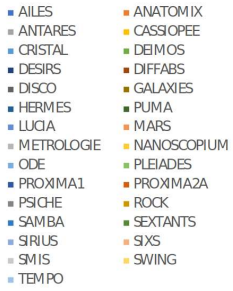
Current status

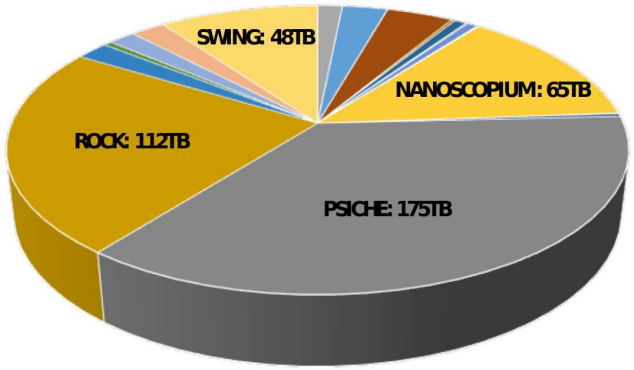
➔ NeXus/HDF5 data format is the « standard » at SOLEIL

- > 10 millions files
- Almost all beamlines (25 out of 30) record data in NeXus/HDF5 files
  - in 15 beamlines: systematically for raw data
  - in the last 10: depending on the context (instrument, ...)



Number of files





Data volume

Storage System status,05/13/2019

Current status

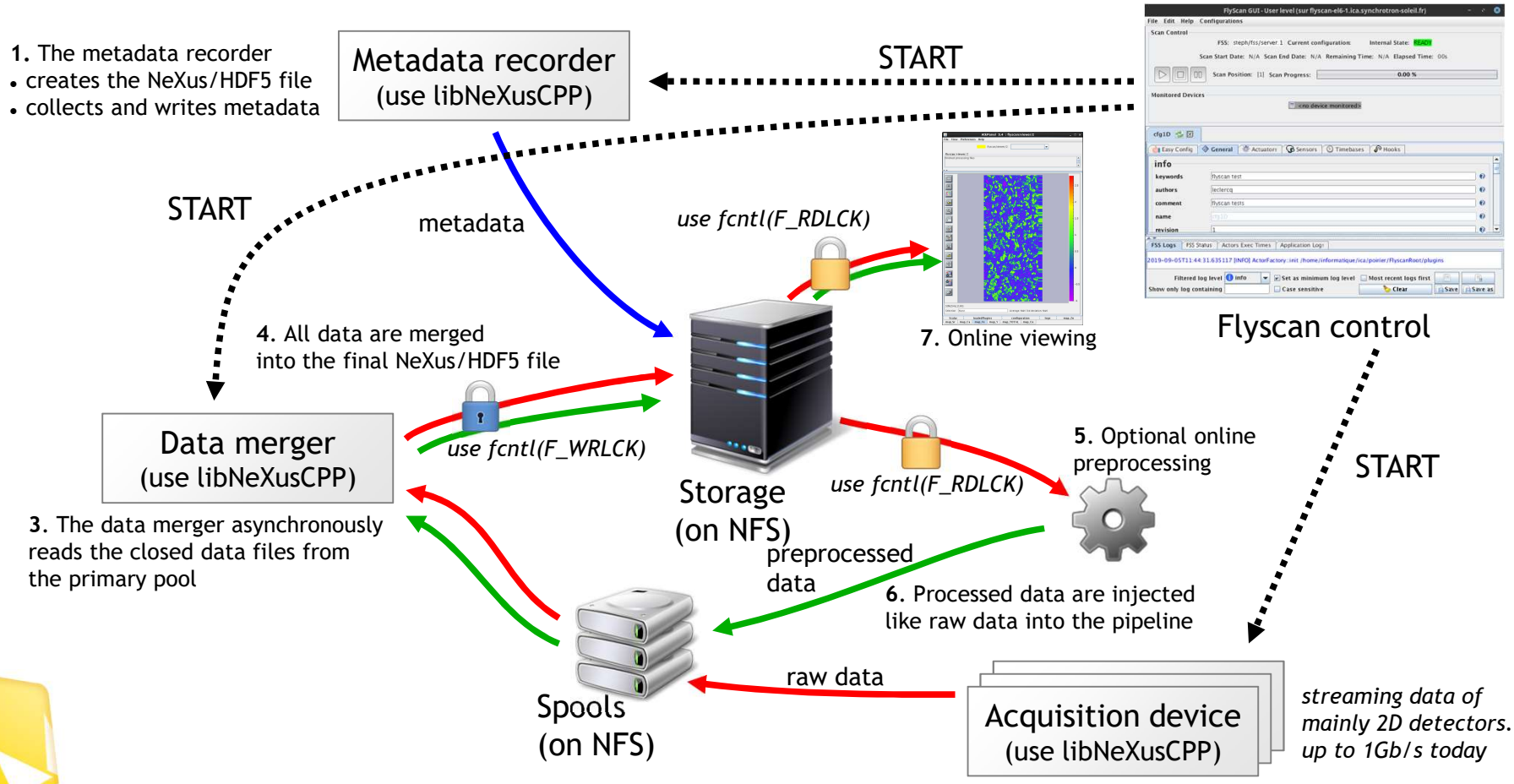
- ➔ SOLEIL use Tango as its control system for the accelerators and the beamlines,
- ➔ Recording is managed using a set of Tango 'devices'
  - 'devices' are pieces of software able to communicate to each other using on the software bus and to control physical equipements
- ➔ Each device that need to record experimental data or metadata uses a C++ library: libNexusCPP, developped at SOLEIL.

Recording service

- ➔ Originally built on top of the libNeXus provided by the NeXus Advisory Commetee (NIAC).
- ➔ But the NIAC had stopped the devloppement of this library. Therefore the libNeXusCPP was re-written on top of the CPP HDF5 lib.
- ➔ The current version
  - is based on HDF5 1.8.x
  - Can create/read/write Nexus files
  - For experimental data, propose synchronous or asynchronous streaming API
  - Can use Posix locks to manage concurrent access

Recording service

Experimental & contextual data flow for continuous scans (Flyscan)



Recording service: Conclusion/Next steps

- ➔ HDF5 as a very fast/efficient/reliable API and data container
- ➔ Switching to 1.10.x but all client applications need to migrate first (it's on the way).
- ➔ Make use of a compression algorithm (probably LZ4)
  - At least for temporary files
- ➔ Using Virtual datasets (is it possible through the CPP API?)
  - Today we 'simulate' this functionnality for very large datasets
- ➔ Using HDF5 SWMR for better locking efficiency.
- ➔ Aligning HDF5 files to the NeXus Application Definitions

Data analysis

- ➔ **We use HDF5 as a very efficient API and data container**
- ➔ **We choose NeXus because it offer a standard data organization (at group level)**
  - We don't try to strictly respect dataset names as defined by the NIAC
- ➔ **Our experience showed that it's almost impossible to standardize dataset names accross all beamlines**
- ➔ **The CDMA (Common Data Model Access) API is an effective solution to this issue**
  - Data format abstraction toolkit based on a dictionary mechanism, for names and paths between data files and data processing applications

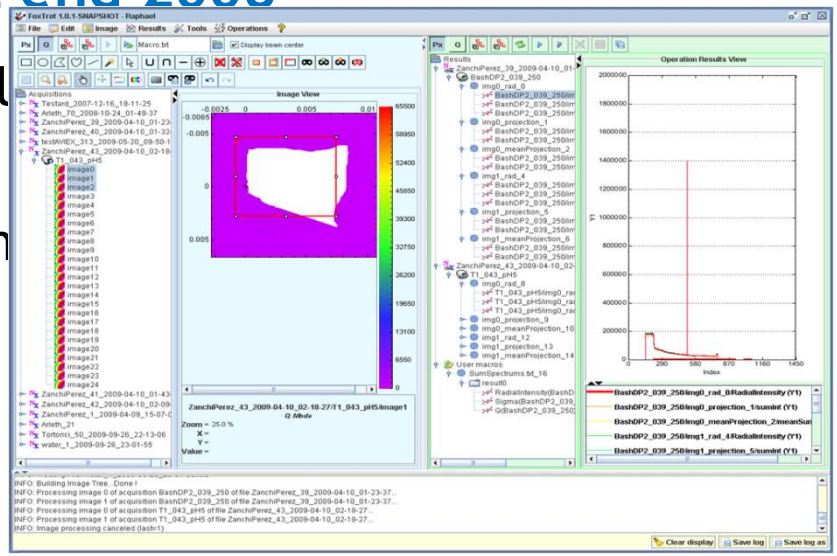
Data analysis services: history

➔ ActionJava used fors SAXS data analysis until 2008

- Developed by a single person on a beamline
- Integrates Tango control functionalities
- Hard to maintain and not integrated in ICA standard software

➔ ICA decided to develop Foxtrot end 2008

- Main decision: this software wil
  NeXus/HDF5 files
- Foxtrot V1 delivered on Septem



Data analysis services: history

➔ **ICA group is in charge of mainting Foxtrot since 2009**

- Migration to COMETE graphical framework with Foxtrot V2 (2010)
- Feedbacks done by SWING beamline

➔ **More flexibility offered to beamline users**

- Possibility to define their own ImageJ macros for data analysis
- Possibility to launch application on personnal computer
- Application can be downloaded by users outside of SOLEIL

➔ **Difficulties appeared:**

- Application is based on the very architecture of the NeXus/HDF5 files.
- It is not practical to use it « as is » on other beamlines.

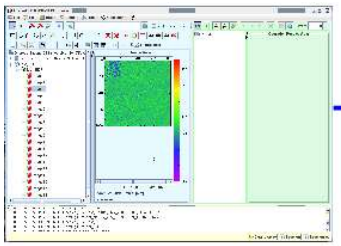
Data analysis services: Problems

- How may my application work with other NeXus/HDF5 files that don't have the same architecture as mine ?
- I want to collaborate with people that don't use HDF5 files, but need the same kind of application as mine to work with their files. What can I do ?

Data analysis services: CDMA

➔ In 2010, the CDMA project started in collaboration with ANSTO

« Scientific » keyword dictionary of the application



keywords Declaration file

```
<data-def name="Experiment name">
<!-- ex: EXAFS, SAXS,... →

  <item key="mono_energy">
  <item key="mono_type">

</data-def>
```

CDMA

Files format plugin

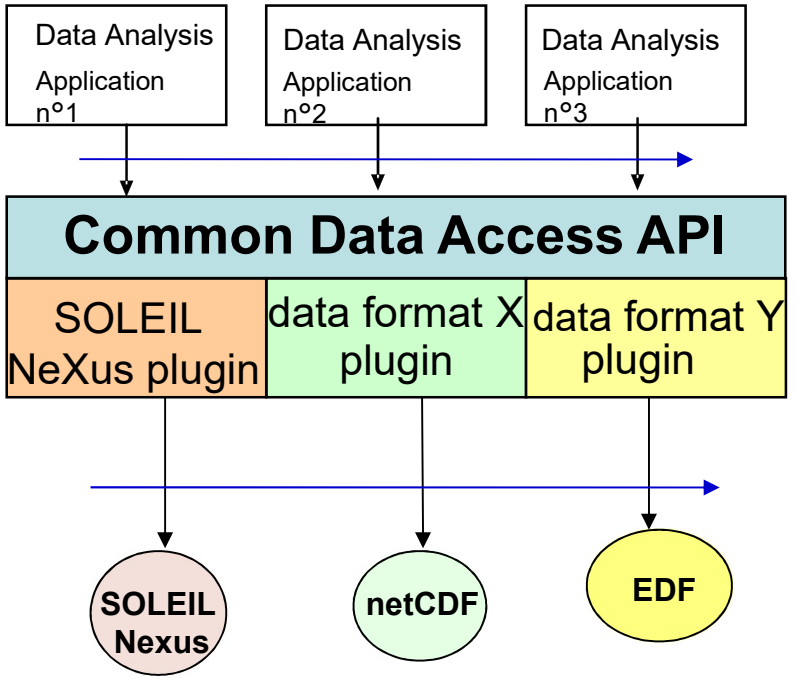
0100110
1001110
0100110
11110...

keywords Mapping file

```
<map-def name="Experiment name">
<!-- ex: EXAFS, SAXS,... -->

  <item key="mono_energy">
    <path>path/to/user/mono/energy</path>
  </item>
  <item key="mono_type">
    <path>path/to/user/mono/type</path>
  </item>
  ...
</map-def>
```

Mapping dictionary



Data analysis services: other software

➔ **Flamenco software development started in 2010**

- Relies on CDMA and Comete
- Uses the same ImageJ macros extension module as Foxtrot
- Works with spectrum stacks instead of image stacks.

➔ **As the data analysis need became stronger and stronger, with custom software, on various beamlines, it was decided to develop the « Fusion » data reduction framework**

- Graphical components Library
- Data treatment library
- Common way to use CDMA

Data analysis services: current state

- **ImageReducer** (core of Foxtrot) **and SpectrumReducer** (core of Flamenco)
  - Deployed on all beamlines
  - Operational, according to the disponibility of nexus files and the adapted dictionary
  - Some beamlines have their custom data reduction software (specific data reduction plugins) : ANTARES, CRISTAL, DIFFABS, HERMES, LUCIA, ROCK, SEXTANTS, SIRIUS, SWING
- **Java version of CDMA migrated to HDF5 1.10**
  - Use of SWMR for parallel data treatment

Data analysis services: next steps

- ➔ Move to HDF5 1.10.X for C/C++ CDMA APIs
- ➔ Align C++ CDMA API to the current java version
- ➔ Implement a Python 3 CDMA API
- ➔ Management of datasets splitted into many HDF5 files
- ➔ Work on global performances

Questions?