

DMSC Status

Thomas Holm Rod
Data Analysis and Modelling group, DMSC



www.europeanspallationsource.se

14 June, 2018

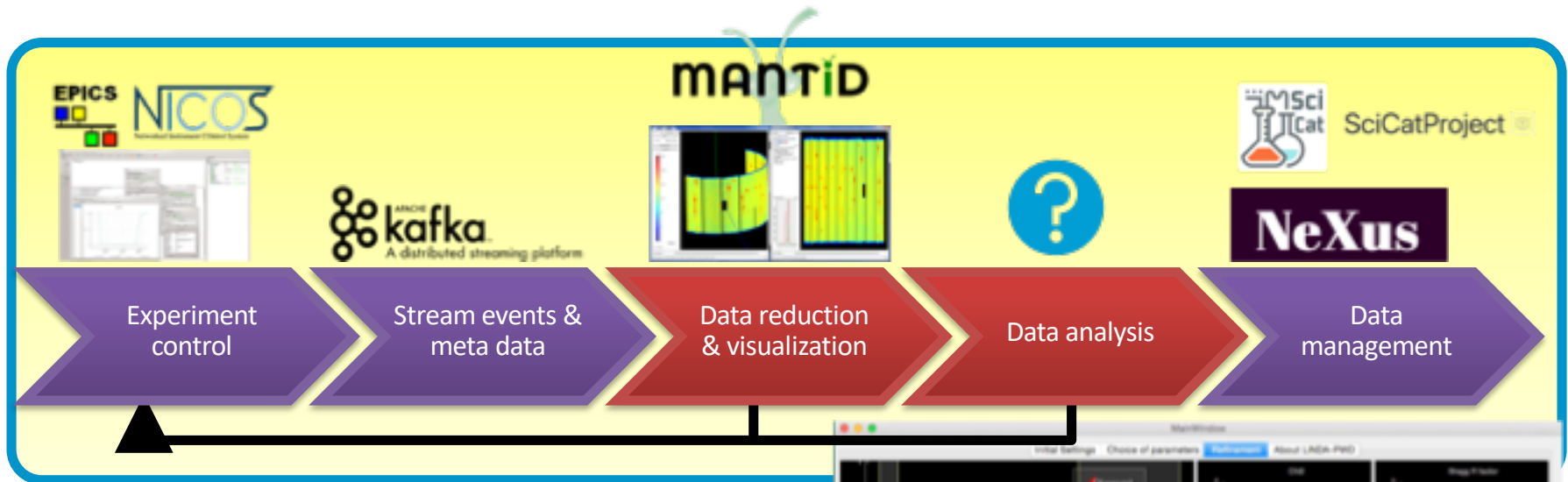
Data Management and Software Center



ESS Software stack

Functionality	Software	Partners
User office		Max IV
Experiment control	NICOS / EPICS	ISIS, PSI, FRMII
Data streaming	Kafka	ISIS, PSI
Data catalogue	SciCat	PSI, MAX IV
Data format	NeXus	ISIS
Data reduction	Mantid	ISIS (SNS, ILL)
Analysis & modelling		

Integrated data processing pipeline



(LINDA + Fullprof)

- Make *integrated* data processing pipeline work smoothly and seamlessly
- For each step provision *maintainable* software
 - For data analysis: build on *existing* software / frameworks

ESS strategy for maintainable and user-friendly software (non-functional requirements)

ESS should be able to maintain, sustain and (rapidly) extend chosen software.

Focus on *few* existing programs and ensure:

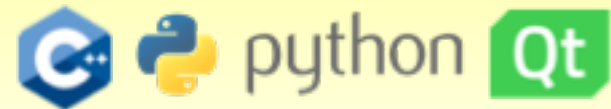
- Modularization / APIs
- CLI (Python) & GUI
- Facilitate 3rd party contributions
- Fully open source and free
- Quality assurance (testing)
- Documented both for users and developers
- No single point of failure
- Widely used (preferably the standard)

- *ESS may install software not maintained by ESS*
- *Users are free to use their own (analysis) software solutions*



Standard & Guidelines

Preferred languages



**Upgrade developers' skills
in UX design**

Analysis: Rietveld refinement

Rietveld Program	Language(s)
Fullprof	Fortran
Jana2006	Fortran
GSAS II	Python/C++
Maud (presentation tomorrow)	Java
Topas	

Preferred solution for powder diffraction

Work with *ILL* to convert *FullProf* into sustainable software:

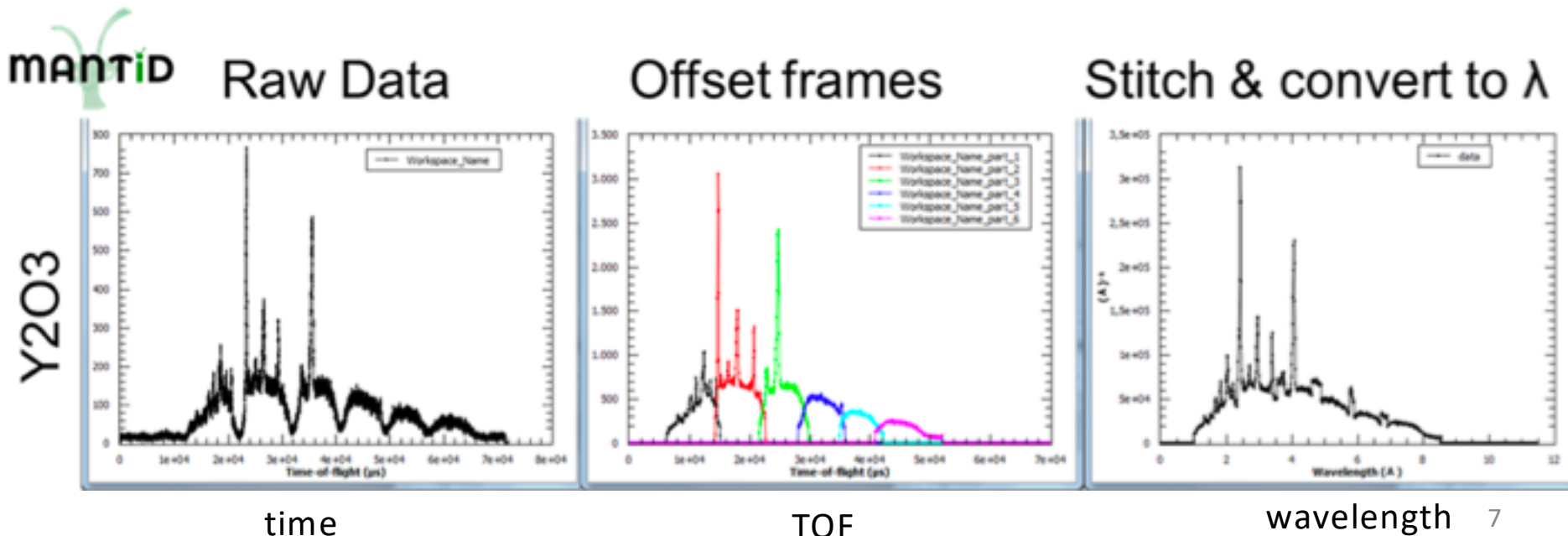
- Python scripting interface
- Qt based GUI
- 2D Rietveld
- Texture analysis
- Documentation
- Modularize
- Quality assurance
- Fully Open Source

- and make other major codes available as well

**Additional resources
from NMIStar
– if funded!**

Data reduction & V20: WFM stitching

- ESS data reduction team involved with V20 since 2016
- Versioned pip packages now used for deployment
- Much simpler installation via single script as of June 2018
- Not possible so far to base WFM only on chopper settings



Data catalogue: SciCat

- **F**indable, **A**ccessible, **I**nteroperable, **R**eusable
- Links to Data Acquisition and Proposal System
- Data format: NeXus

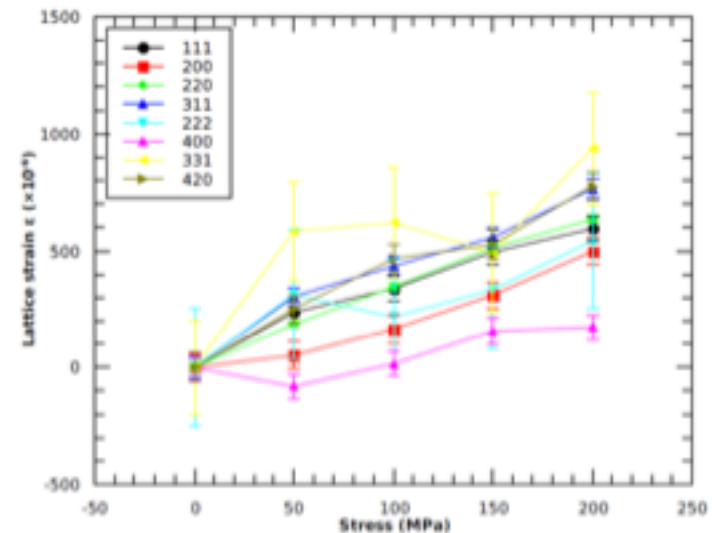
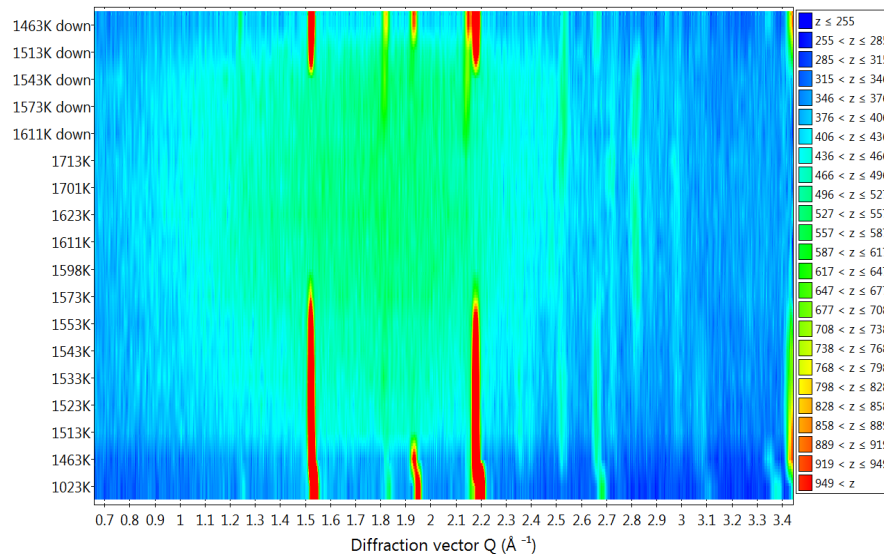


Data Curation Workshop at DMSC

- SciCat
- Reduction and Analysis Workflow
- Data Publication Workflow DOI

User interfaces during experiment: Visualization

Automated feedback from real-time data analysis to experiment control. How should the user operate and monitor this?



To be updated on the fly

Plots from BEER Concepts of Operations

User interfaces during experiment

Automated feedback from real-time data analysis to experiment control. How should the user operate and monitor this?

Tentative suggestion:

Experiment Control UI

- Sample environment state (meta-data)
- Events on detector
- Histogram(s) [as function of meta-data or time]

Analysis UI

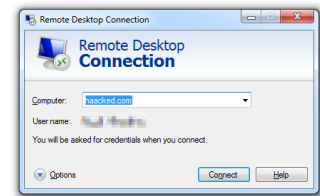
- Histogram(s) as function of meta-data or time
- Result from analysis (e.g. lattice constant) as function of meta data or time

Both UIs to be updated on the fly

Users also need access to Mantid so that settings for data reduction can be changed

User interfaces after experiment

- Users can access data remotely through SciCat
- Other scientists can access data after three years through SciCat
- Users can remotely process data at ESS.
Technologies considered are:
 - Remote desktops
 - Jupyter notebooks
 - Web interfaces (SciCat)



Questions for you

1. Which programs do you or your users depend on?
2. Which data bases and other external services do they depend on?
3. Which CPU demanding tasks do users need / want to run?
4. What are the biggest bottlenecks?

Questions for me?

