MANTID

The Mantid Project

The challenges of delivering flexible HPC for novice end users

Nicholas Draper

SOS18

ISIS

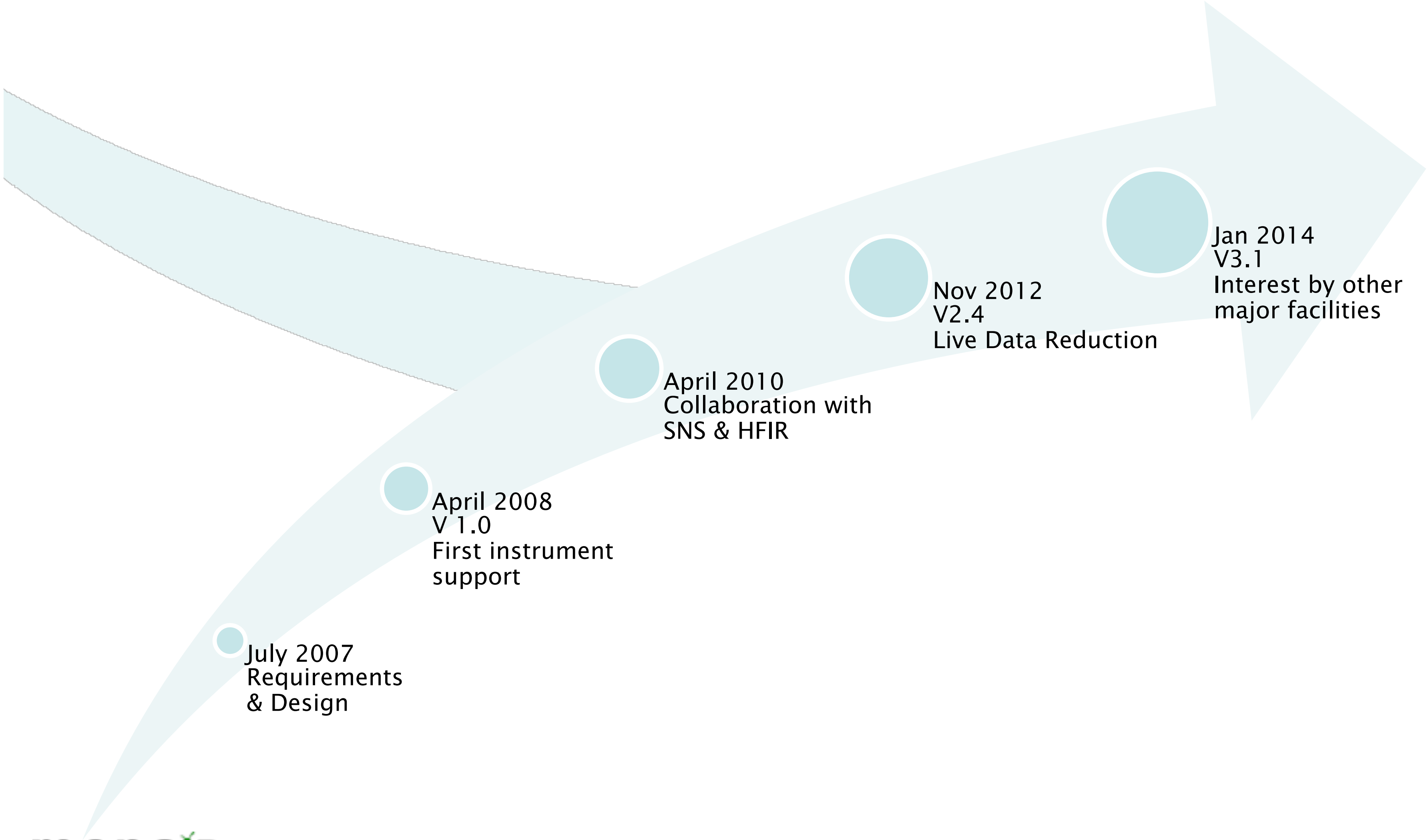
Tessella
Technology & Consulting

SNS
SPALLATION NEUTRON SOURCE

What Is Mantid

- A framework that supports high-performance computing and visualisation of scientific data.
- Manipulate and analyse Neutron and Muon data.
  - Neutron Scattering
    - Diffraction, spectroscopy, small angle, reflectometry
  - Muon Spectroscopy
  - Could be applied to other techniques
- Open Source
- Multiple Platforms
  - Windows, Linux, Mac

Project History



July 2007
Requirements
& Design

April 2008
V 1.0
First instrument
support

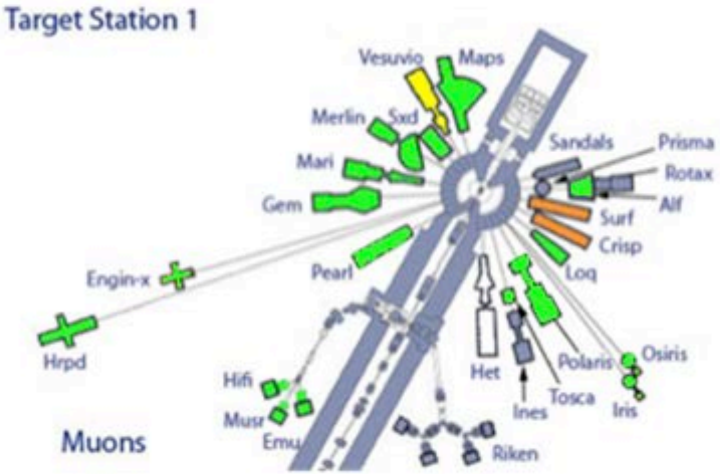
April 2010
Collaboration with
SNS & HFIR

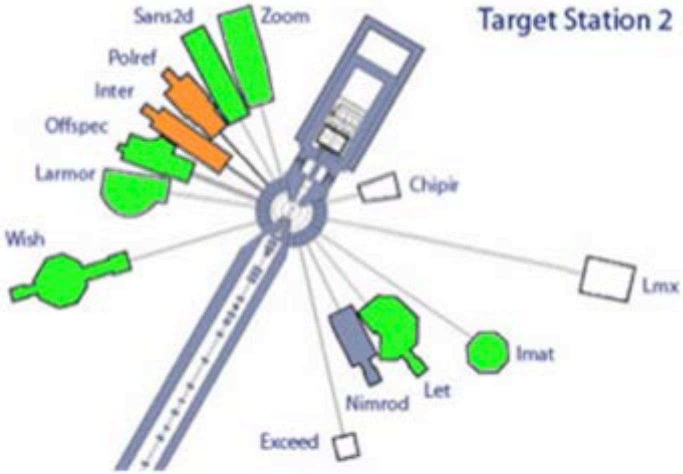
Nov 2012
V2.4
Live Data Reduction

Jan 2014
V3.1
Interest by other
major facilities

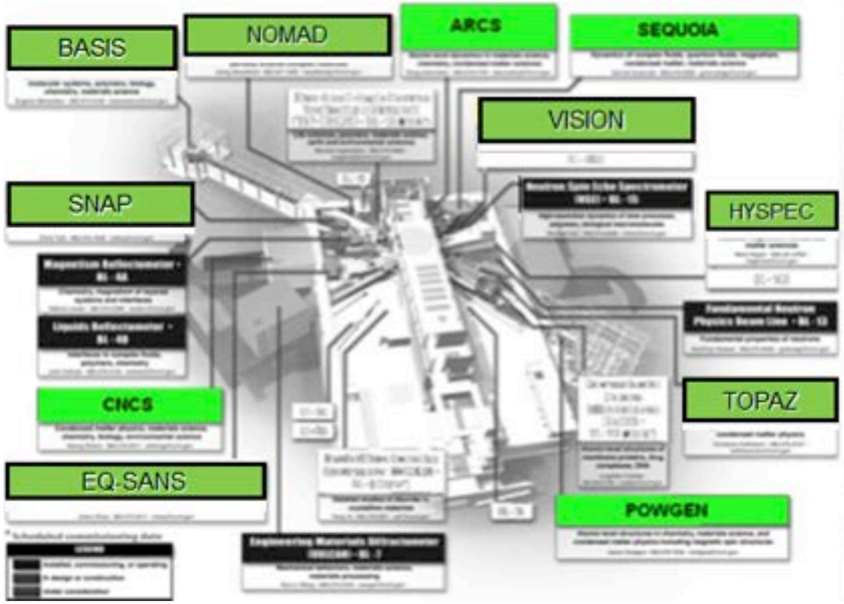
Members

ISIS

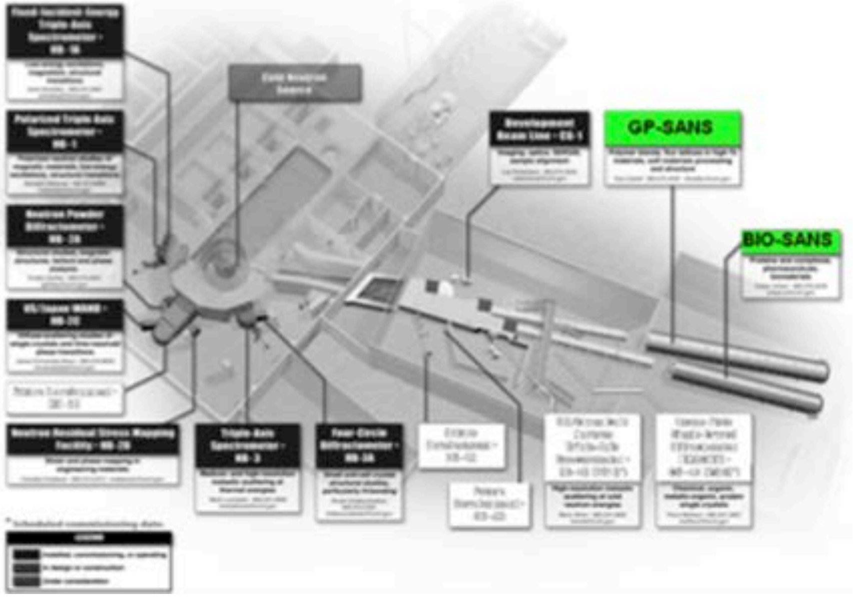




SNS



HFIR



MANTID

Contributors

ill
NEUTRONS
FOR SCIENCE®

ansto

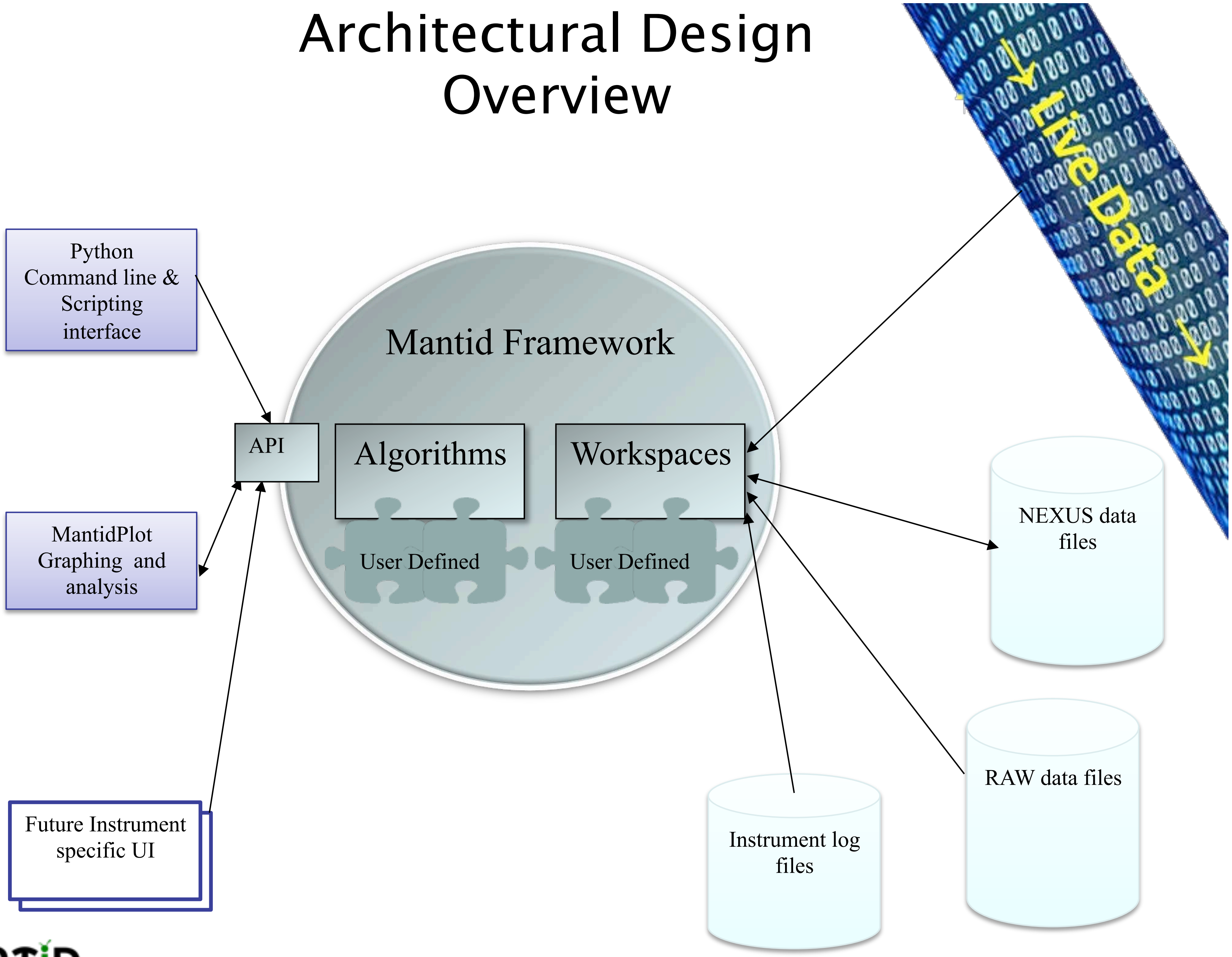
McStas
n

PAUL SCHERRER INSTITUT
PSI

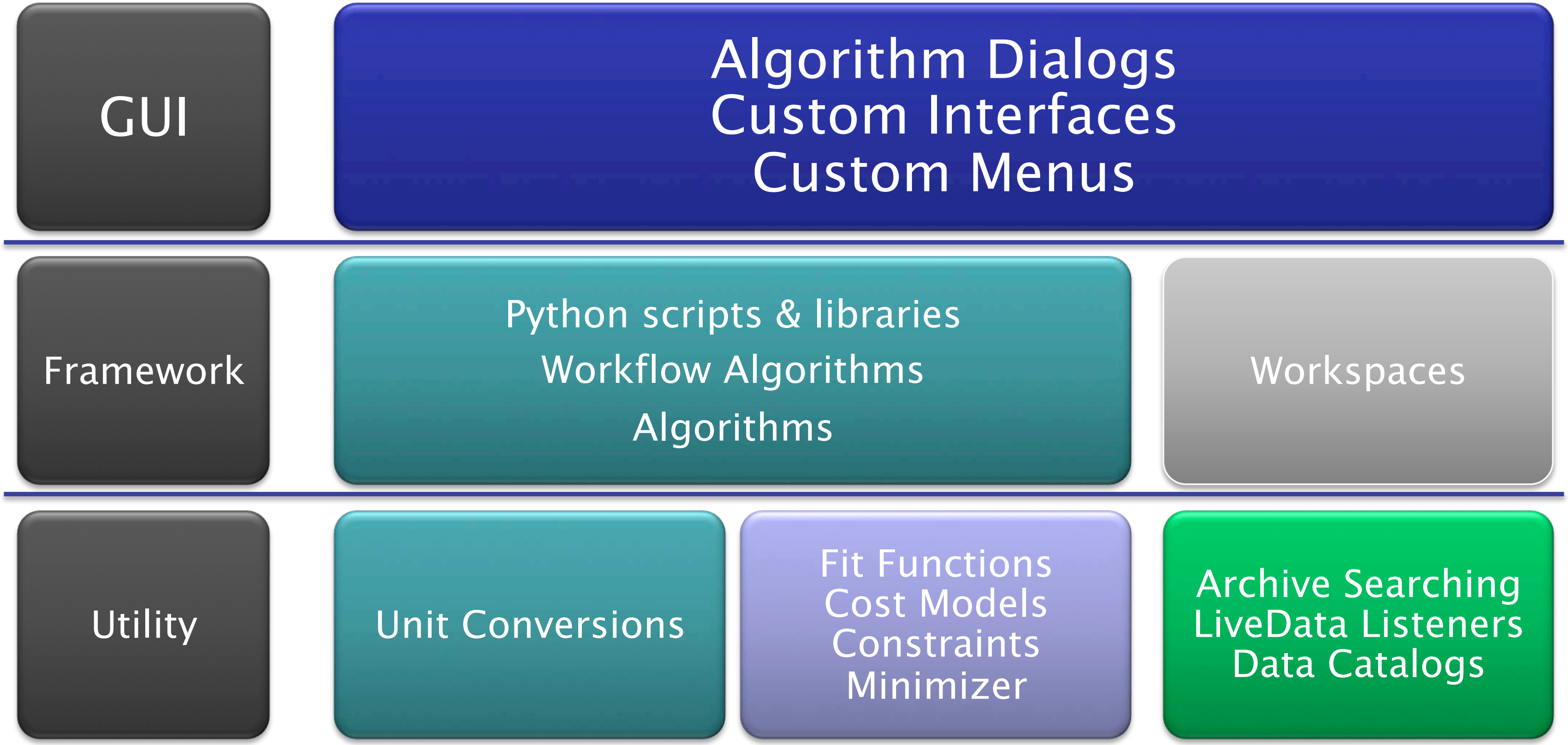
HZB Helmholtz
Zentrum Berlin

FRM II
Forschungs-Neutronenquelle
Heinz Maier-Leibnitz

Architectural Design Overview



Plug in extensions



GUI

Algorithm Dialogs
Custom Interfaces
Custom Menus

Framework

Python scripts & libraries
Workflow Algorithms
Algorithms

Workspaces

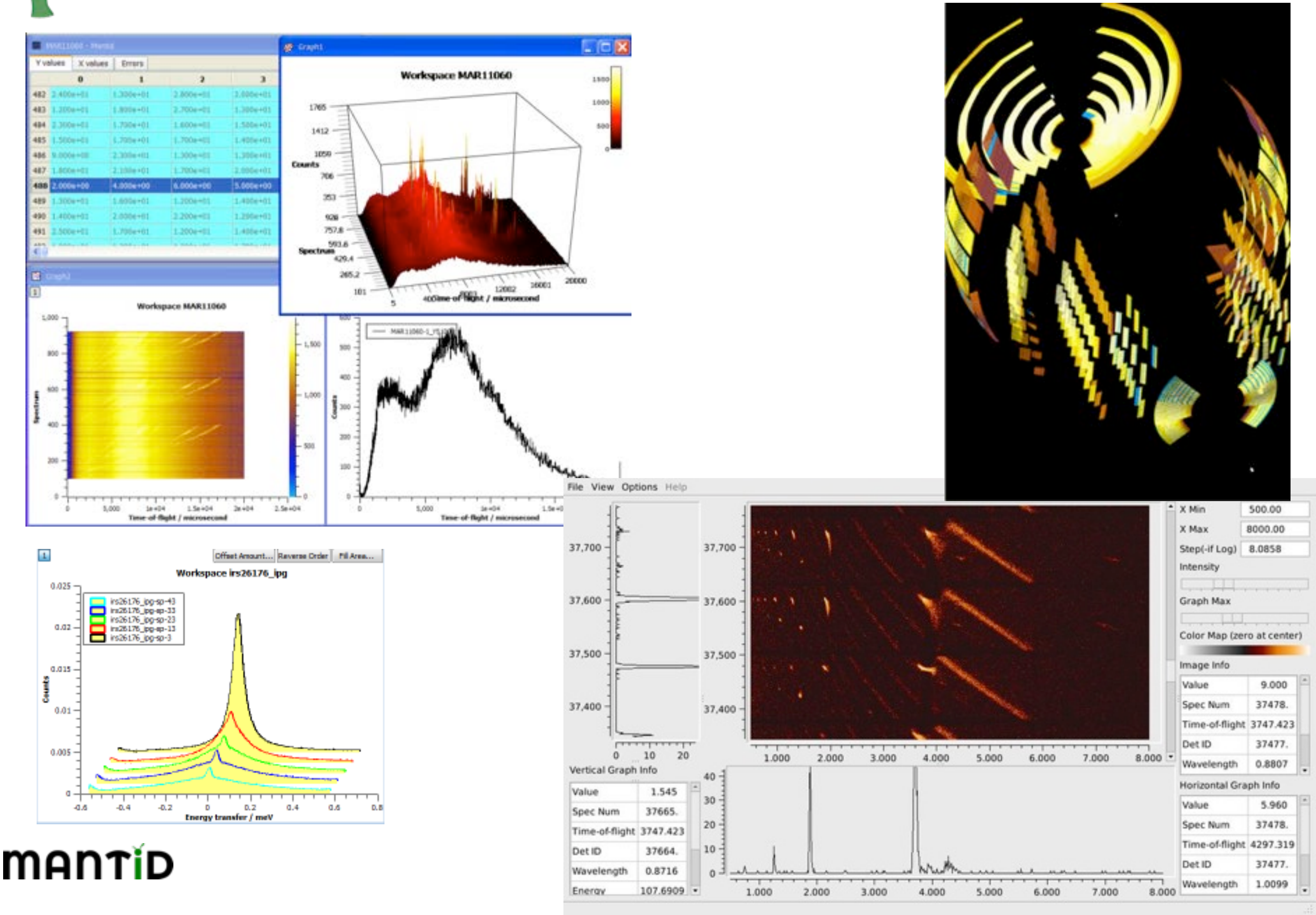
Utility

Unit Conversions

Fit Functions
Cost Models
Constraints
Minimizer

Archive Searching
LiveData Listeners
Data Catalogs

What can we do - Visualization



	0	1	2	3
482	2.400e+01	1.300e+01	2.800e+01	3.000e+01
483	1.200e+01	1.800e+01	2.700e+01	1.300e+01
484	2.300e+01	1.700e+01	1.600e+01	1.500e+01
485	1.500e+01	1.700e+01	1.700e+01	1.400e+01
486	9.000e+00	2.300e+01	1.300e+01	1.300e+01
487	1.800e+01	2.100e+01	1.700e+01	2.000e+01
488	2.000e+00	4.000e+00	6.000e+00	5.000e+00
489	1.300e+01	1.600e+01	1.200e+01	1.400e+01
490	1.400e+01	2.000e+01	2.200e+01	1.200e+01
491	2.500e+01	1.700e+01	1.200e+01	1.400e+01

X Min 500.00

X Max 8000.00

Step(-if Log) 8.0858

Color Map (zero at center)

Image Info

Value	9.000
Spec Num	37478.
Time-of-flight	3747.423
Det ID	37477.
Wavelength	0.8807

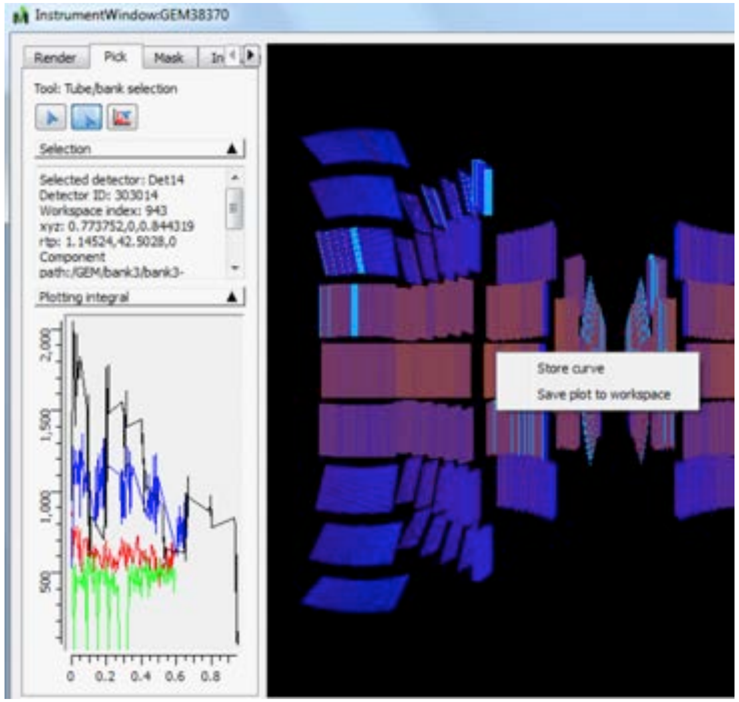
Horizontal Graph Info

Value	5.960
Spec Num	37478.
Time-of-flight	4297.319
Det ID	37477.
Wavelength	1.0099

Vertical Graph Info

Value	1.545
Spec Num	37665.
Time-of-flight	3747.423
Det ID	37664.
Wavelength	0.8716
Energy	107.6909

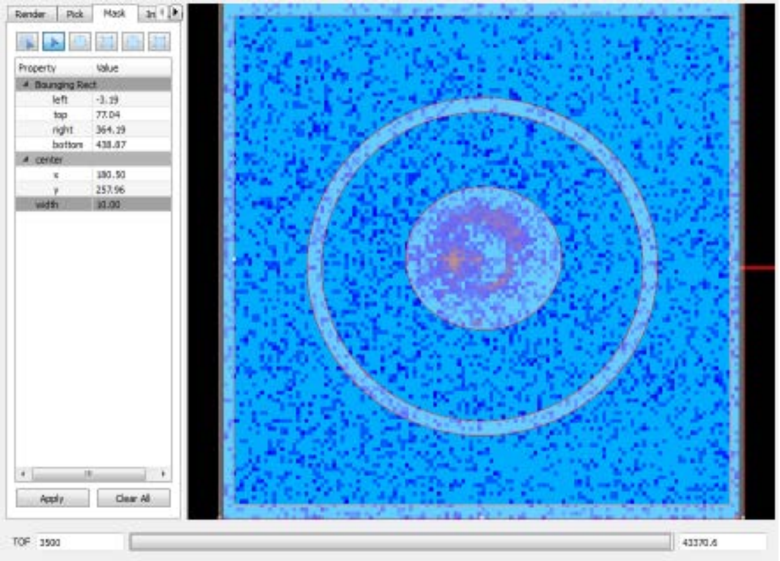
What we can do - Tools



InstrumentWindow:GEM38370

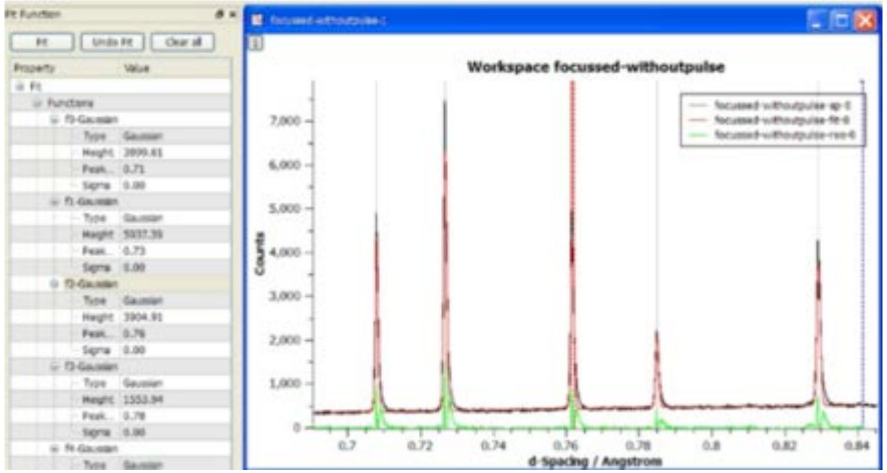
Tool: Tube/bank selection

Selected detector: Det14
Detector ID: 303014
Workspace index: 943
xyz: 0.773752,0,0.844319
rtp: 1.14524,42.5028,0
Component
path:/GEM/bank3/bank3-

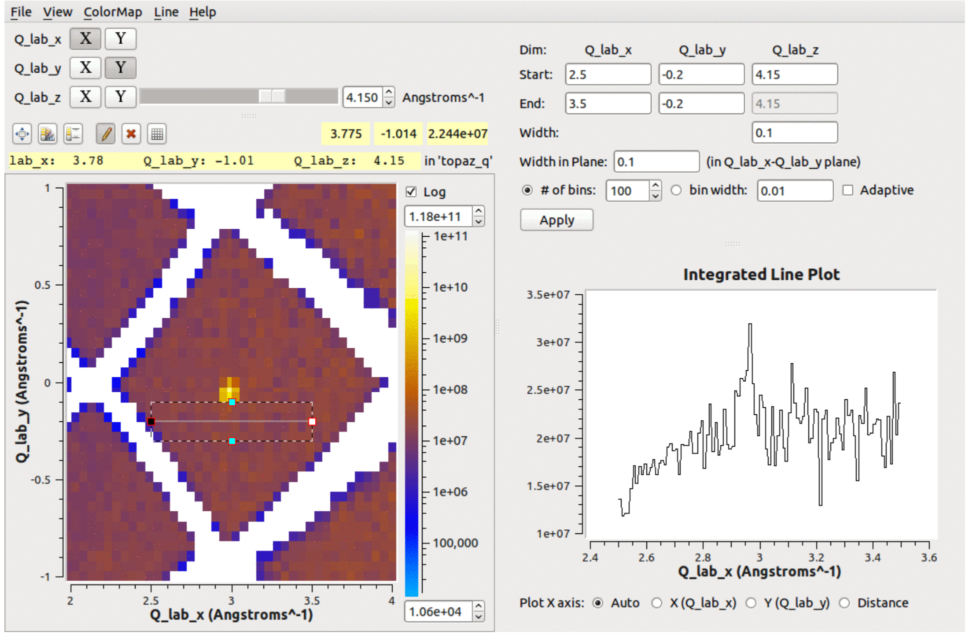


Property	Value
Bounging Rect	
left	-3.19
top	77.04
right	364.19
bottom	438.87
center	
x	180.50
y	257.96
width	10.00

TOF 3500 43370.6



Property	Value
Fit	
Functions	
f0-Gaussian	
Type	Gaussian
Height	2899.41
Peak...	0.71
Sigma	0.00
f1-Gaussian	
Type	Gaussian
Height	5037.35
Peak...	0.73
Sigma	0.00
f2-Gaussian	
Type	Gaussian
Height	3904.91
Peak...	0.76
Sigma	0.00
f3-Gaussian	
Type	Gaussian
Height	1553.94
Peak...	0.78
Sigma	0.00
f4-Gaussian	
Type	Gaussian



Q_lab_z 4.150 Angstroms^-1

3.775 -1.014 2.244e+07

lab_x: 3.78 Q_lab_y: -1.01 Q_lab_z: 4.15 in 'topaz_q'

Dim:	Q_lab_x	Q_lab_y	Q_lab_z
Start:	2.5	-0.2	4.15
End:	3.5	-0.2	4.15
Width:			0.1

Width in Plane: 0.1 (in Q_lab_x-Q_lab_y plane)

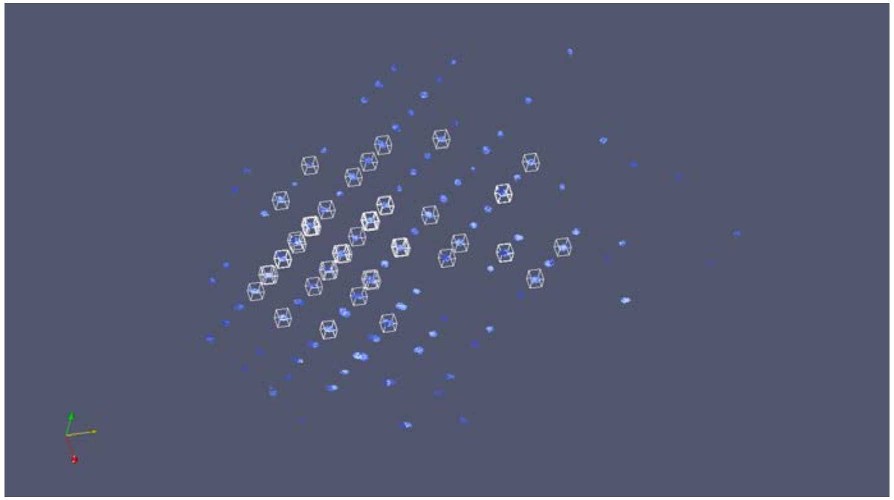
● # of bins: 100 ○ bin width: 0.01 □ Adaptive

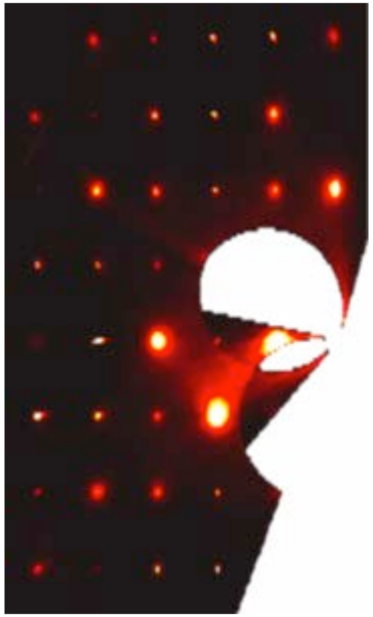
Apply

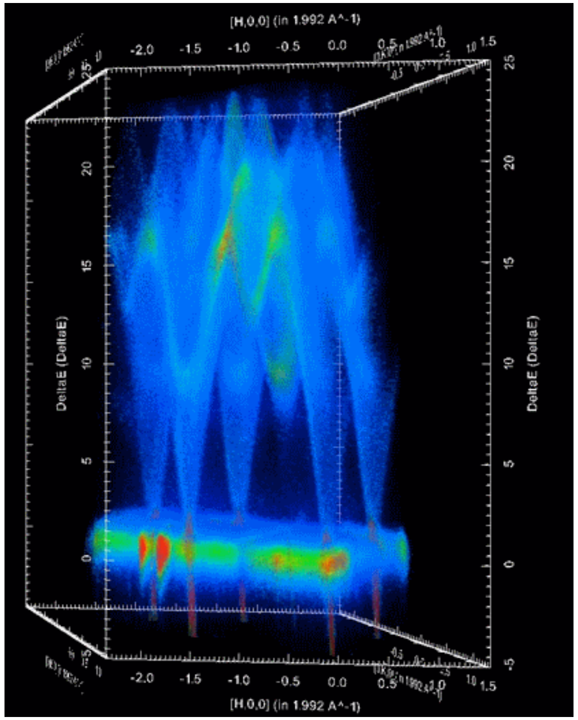
Plot X axis: ● Auto ○ X (Q_lab_x) ○ Y (Q_lab_y) ○ Distance

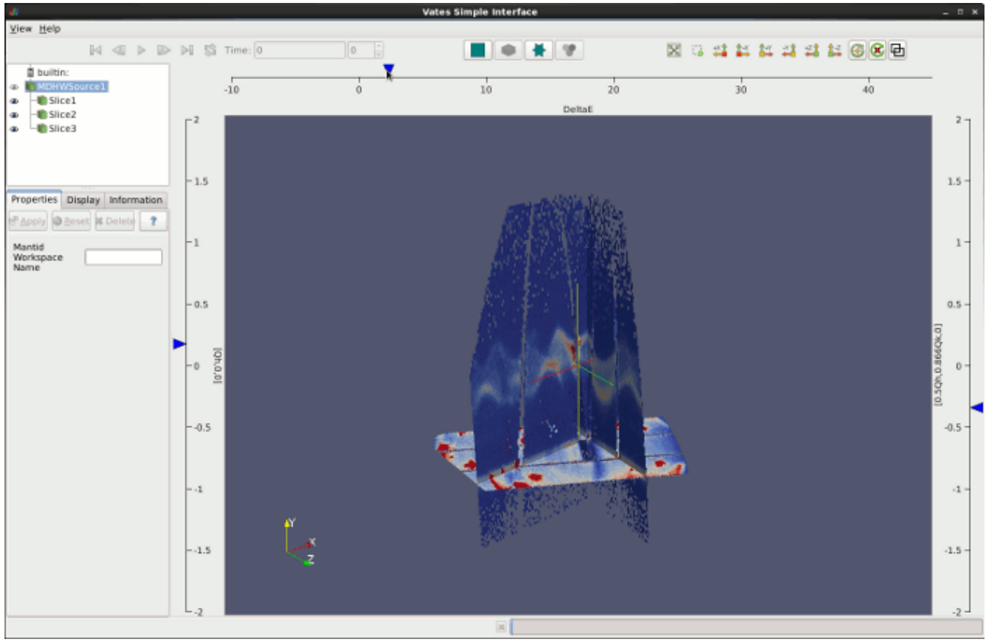
☑ Log

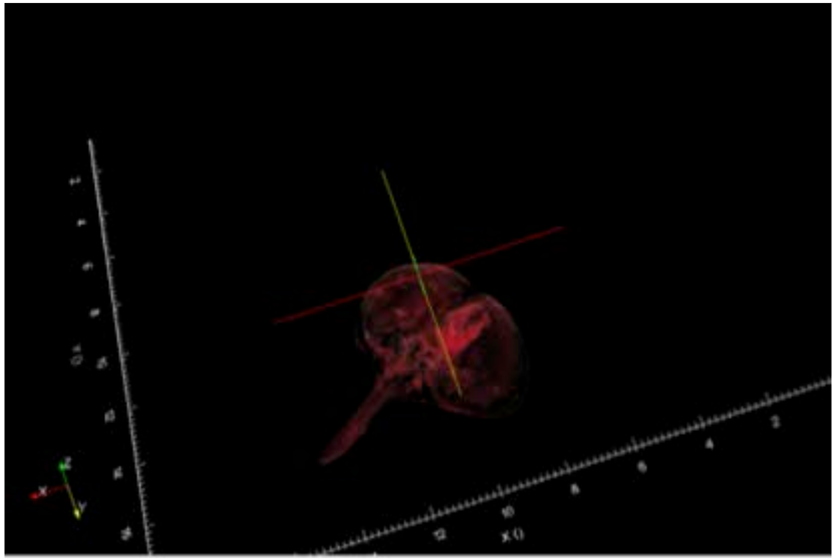
What can we do - Visualization





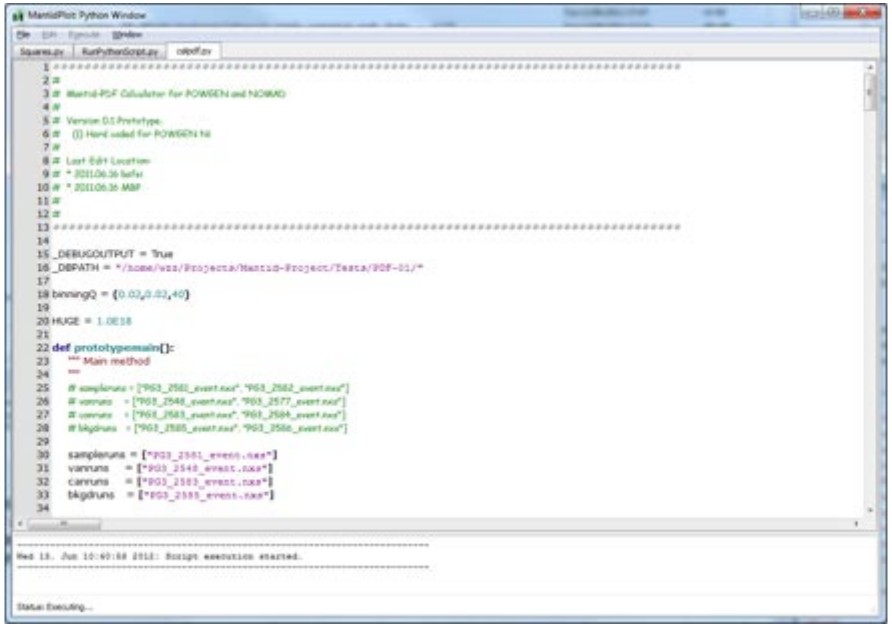






What we can do - Scripting

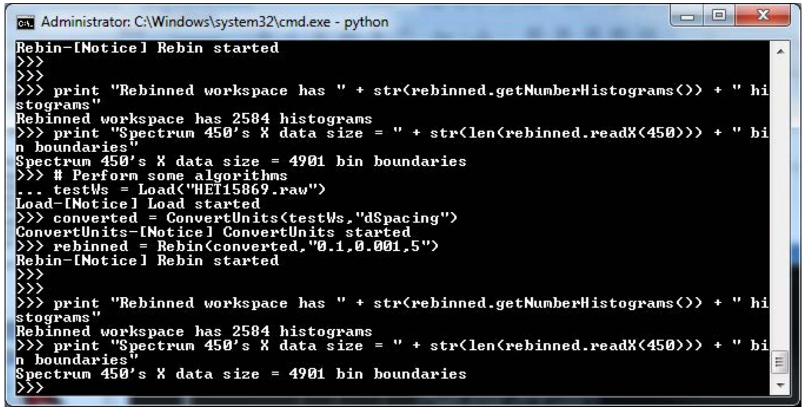
- Python Control
  - Within MantidPlot or command line
  - iPython Shell
  - Integrated numPy
  - Python Plugins
    - Algorithms
    - Fitting functions
    - User interfaces
  - Automated reduction



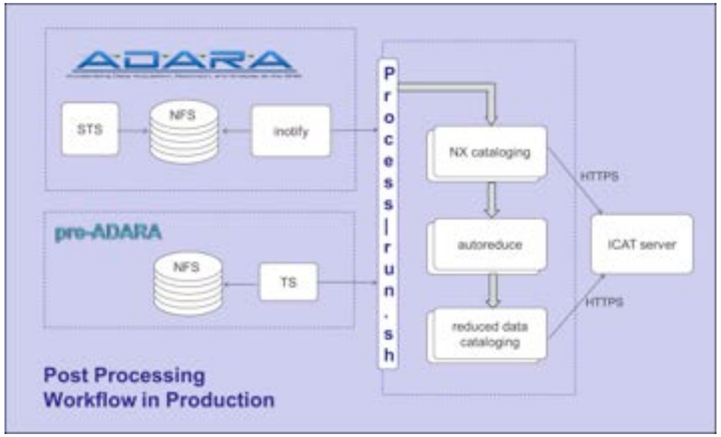
```
################################################################
#
#  Mantid-PDF Calculator for POWGEN and NOMAD
#
#  Version 0.1 Prototype
#    (1) Hard coded for POWGEN hi
#
#  Last Edit Location:
#  * 2011.06.16 befei
#  * 2011.06.16 MBP
#
#
################################################################

_DEBUGOUTPUT = True
_DBPATH = "/home/wzz/Projects/Mantid-Project/Tests/PDF-01/"

binningQ = (0.02,0.02,40)

HUGE = 1.0E18

def prototypemain():
    """ Main method
    """
    # sampleruns = ["PG3_2581_event.nxs", "PG3_2582_event.nxs"]
    # vanruns    = ["PG3_2548_event.nxs", "PG3_2577_event.nxs"]
    # canruns    = ["PG3_2583_event.nxs", "PG3_2584_event.nxs"]
    # bkgdruns   = ["PG3_2585_event.nxs", "PG3_2586_event.nxs"]

    sampleruns = ["PG3_2581_event.nxs"]
    vanruns    = ["PG3_2548_event.nxs"]
    canruns    = ["PG3_2583_event.nxs"]
    bkgdruns   = ["PG3_2585_event.nxs"]

```

```
------------------------------------------------------------------
Wed 13. Jun 10:40:58 2012: Script execution started.
------------------------------------------------------------------
```

Status: Executing...



```
Rebin-[Notice] Rebin started
>>>
>>>
>>> print "Rebinned workspace has " + str(rebinned.getNumberHistograms()) + " hi
stograms"
Rebinned workspace has 2584 histograms
>>> print "Spectrum 450's X data size = " + str(len(rebinned.readX(450))) + " bi
n boundaries"
Spectrum 450's X data size = 4901 bin boundaries
>>> # Perform some algorithms
... testWs = Load("HET15869.raw")
Load-[Notice] Load started
>>> converted = ConvertUnits(testWs,"dSpacing")
ConvertUnits-[Notice] ConvertUnits started
>>> rebinned = Rebin(converted,"0.1,0.001,5")
Rebin-[Notice] Rebin started
>>>
>>>
>>> print "Rebinned workspace has " + str(rebinned.getNumberHistograms()) + " hi
stograms"
Rebinned workspace has 2584 histograms
>>> print "Spectrum 450's X data size = " + str(len(rebinned.readX(450))) + " bi
n boundaries"
Spectrum 450's X data size = 4901 bin boundaries
>>>
```

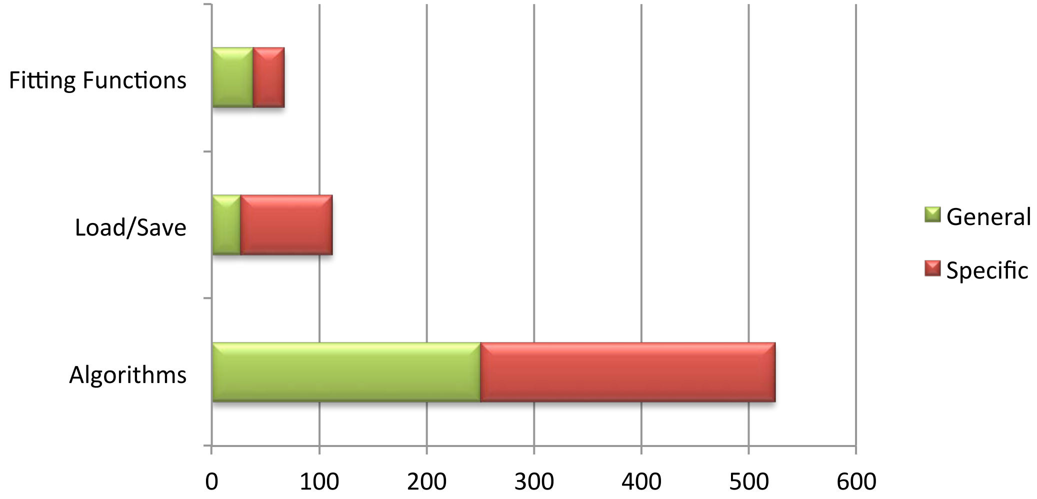


Post Processing Workflow in Production

What we can do - Algorithms

703

- Arithmetic
- Correction Functions
  - Absorption
  - Background
  - Efficiency
- Technique
  - Single Crystal
  - Inelastic
  - SANS
  - Powder diffraction
- Data handling



- Diagnostics
- Event Filtering
- Optimization
- Transforms
  - Masking
  - Grouping
  - Smoothing
  - Unit conversions

The Mantid Environment

- Users
  - From scientific experts, who will understand HPC to some extent, but have limited time.
  - To visiting scientists, who just want results, and have little time to understand systems or learn new processes.

- Compute environments
  - Local computer only
  - Powerful workstations
  - Facility HPC facilities
  - University facilities
  - Commercial Cloud resources
  - National HPC facility

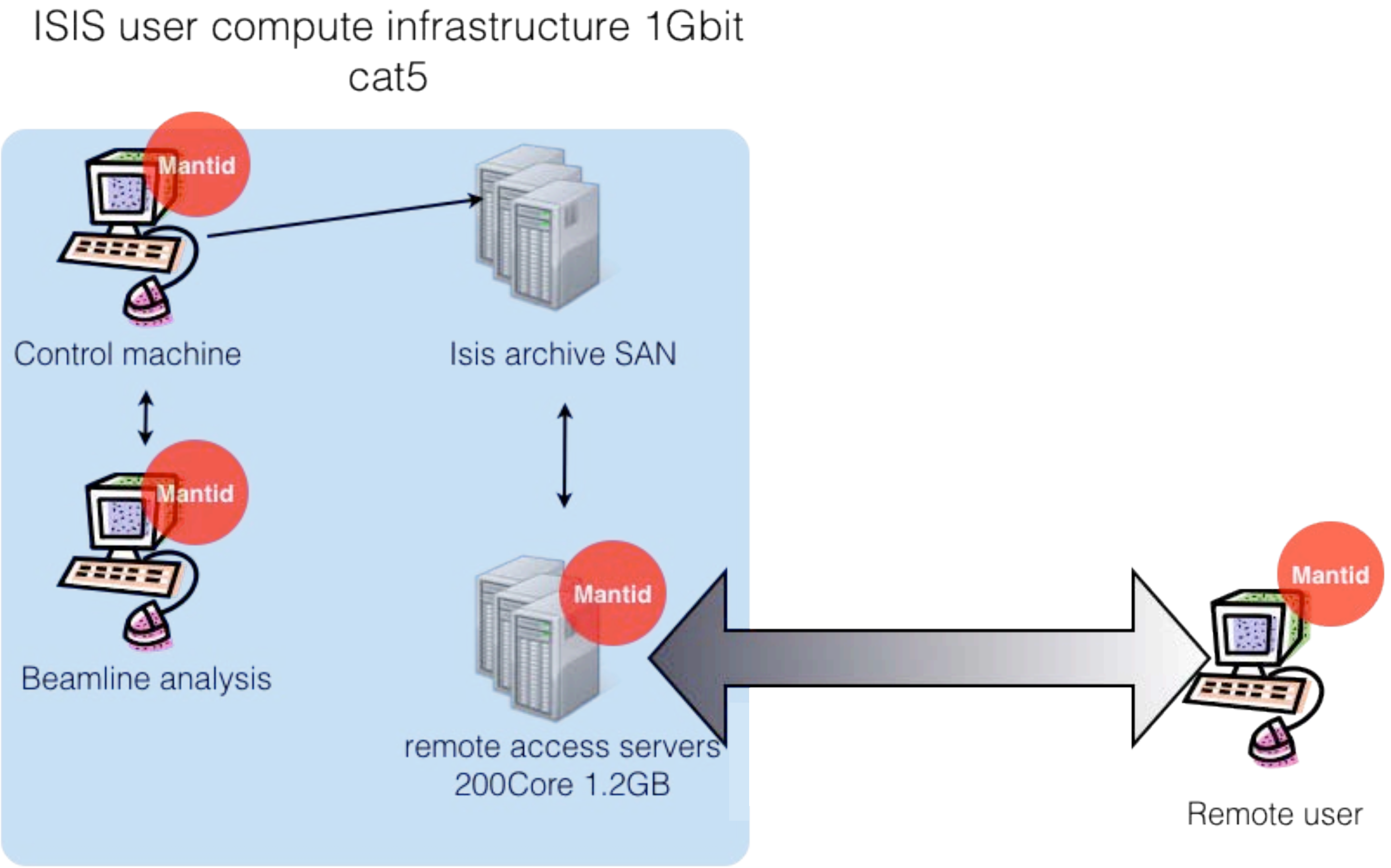
Mantid and Distributed Computing

- Not all operations in Mantid would benefit from distributed or HPC computing
  - Small data volume
  - Large data volume, local to client, simple operations
- For some operations the need is clear
  - Large Data Volume, fast access by cluster
  - Complex scalable operations
    - Monte Carlo instrument simulations
    - Absorption corrections
  - Use of third party codes
    - Molecular dynamics simulations
  - Computationally expensive optimisations of models to experimental data

What is a Mantid job?

- Input data
  - Shared File Location
  - Uploaded File
- Python Script
- Output data
  - Shared File Location
  - Downloaded File

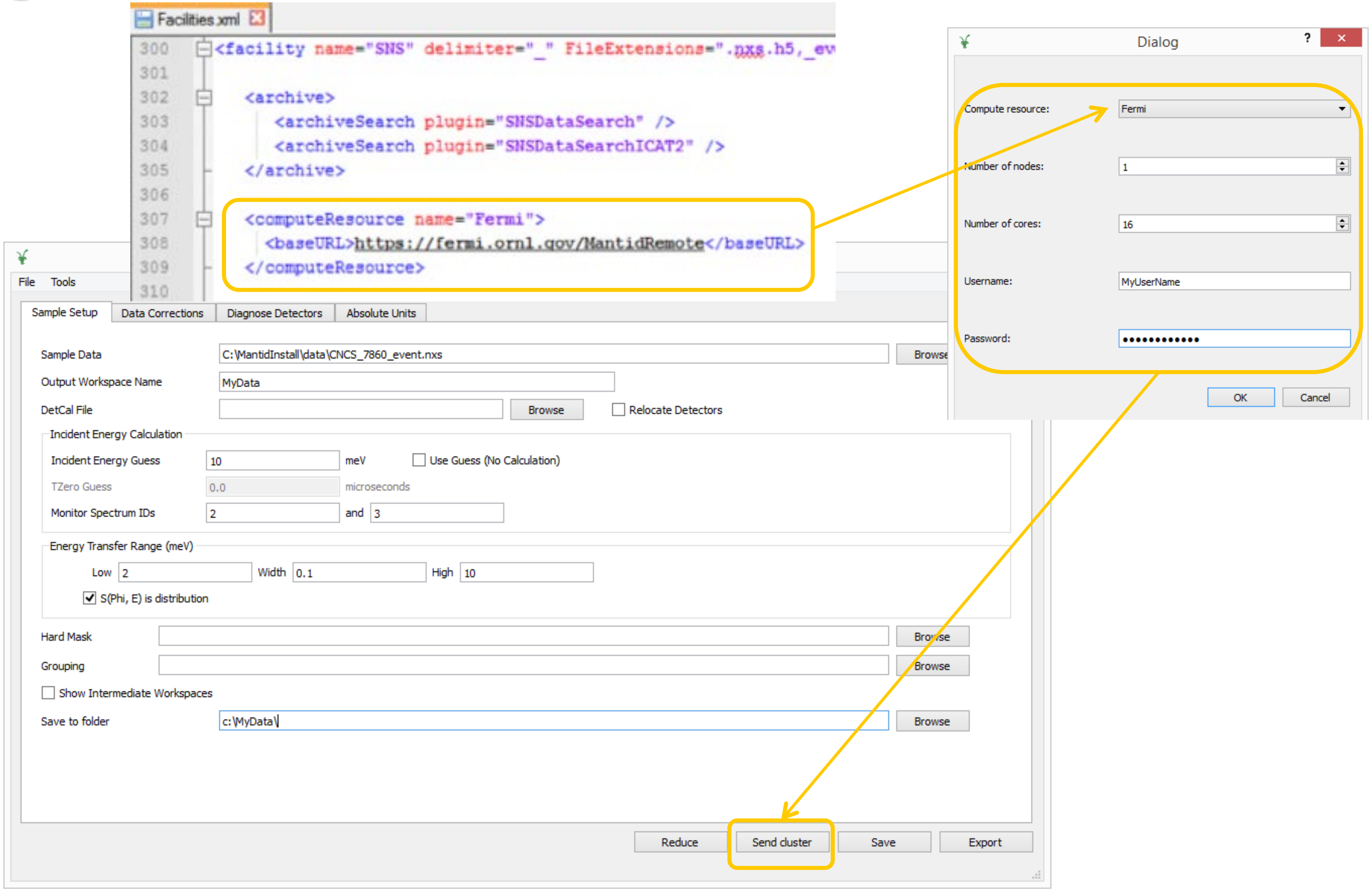
One possible Infrastructure



Challenges

- Different facility infrastructures
  - Authentication
    - Username/password, certificates, need to physically turn up with ID.
  - Job Schedulers
    - Few accessible via web services
    - Of those that do not few work well with C++ cross platform
  - Locating resources

Interfaces - Keep it simple



Facilities.xml

```
<facility name="SNS" delimiter="_" FileExtensions=".nxs,.h5,_ev

  <archive>
    <archiveSearch plugin="SNSDataSearch" />
    <archiveSearch plugin="SNSDataSearchICAT2" />
  </archive>

  <computeResource name="Fermi">
    <baseURL>https://fermi.ornl.gov/MantidRemote</baseURL>
  </computeResource>
```

Dialog

Compute resource: Fermi

Number of nodes: 1

Number of cores: 16

Username: MyUserName

Password: ••••••••••••

Sample Data: C:\MantidInstall\data\CNCS_7860_event.nxs

Output Workspace Name: MyData

DetCal File:

□ Relocate Detectors

Incident Energy Calculation

Incident Energy Guess: 10 meV

□ Use Guess (No Calculation)

TZero Guess: 0.0 microseconds

Monitor Spectrum IDs: 2 and 3

Energy Transfer Range (meV)

Low 2 Width 0.1 High 10

☑ S(Phi, E) is distribution

Hard Mask:

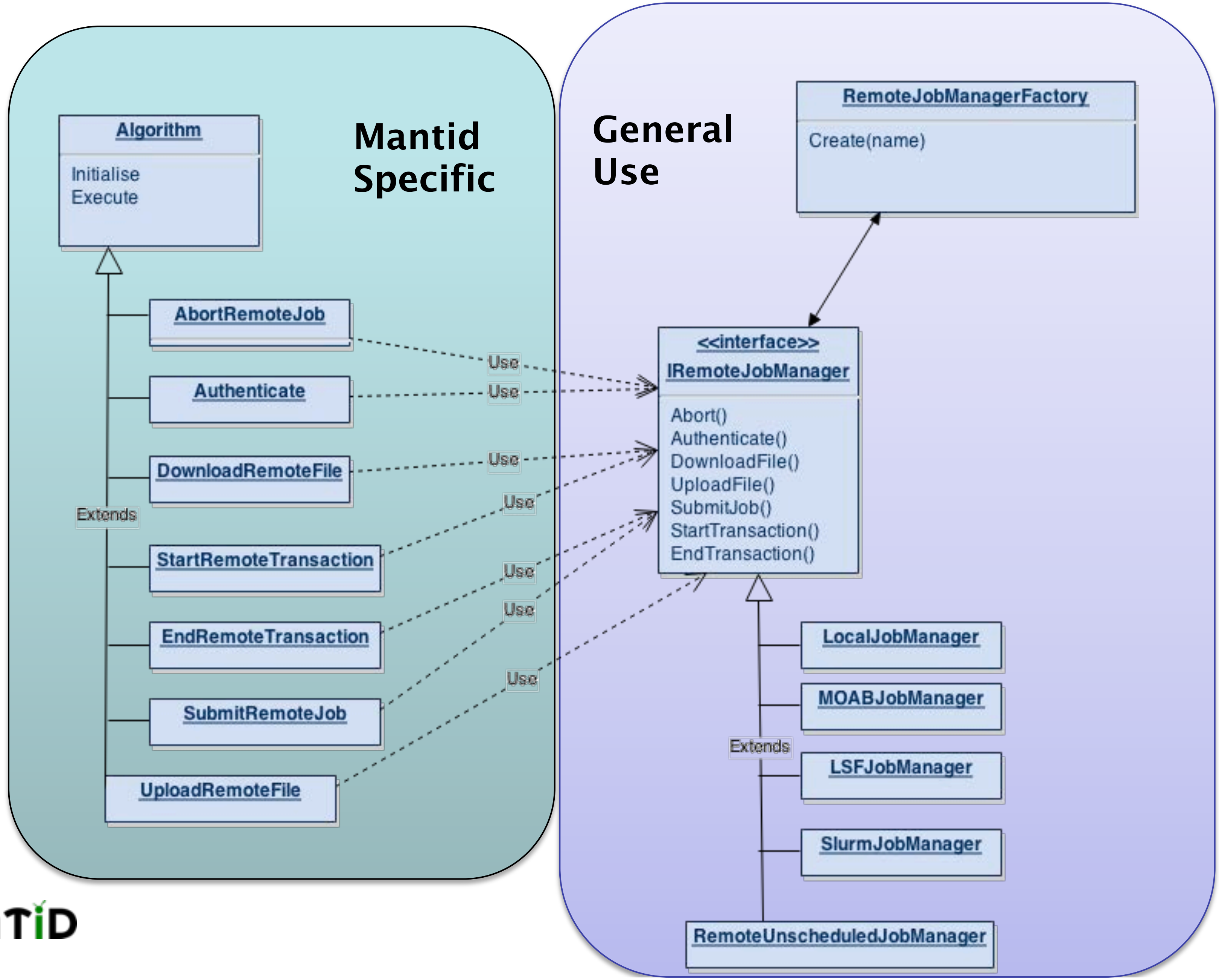
Grouping:

□ Show Intermediate Workspaces

Save to folder: c:\MyData\

MANTID

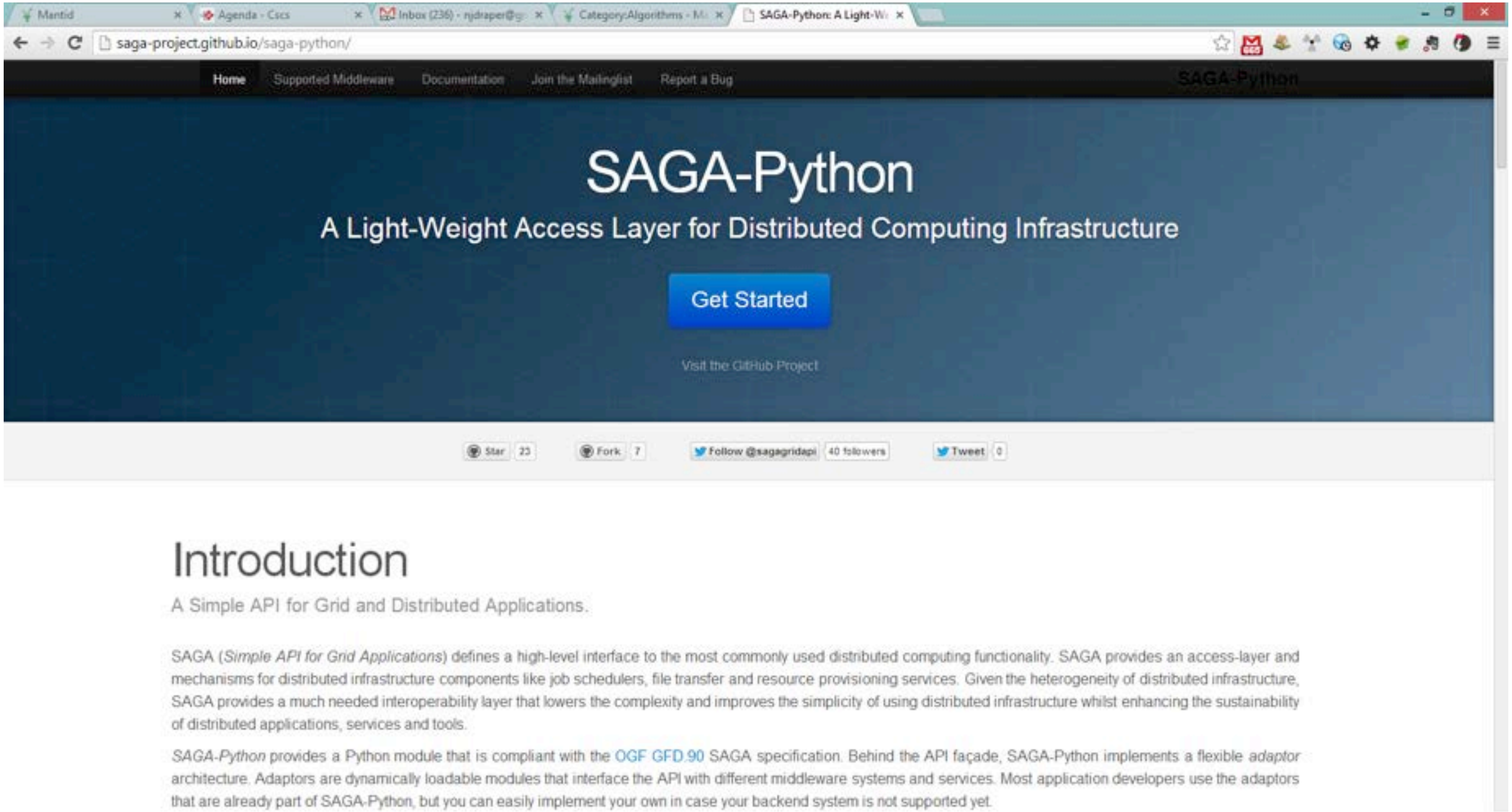
Under the bonnet



The Ideal Middleware

- Removes the need to care what job scheduler is in use.
- Compiles easily on Windows, Mac, Linux
- Packages easily for deployment
- APIs
  - C++, Python
  - others
- Needs to handle network proxies, firewalls etc
- Does not need Admin or special permissions
- Easy to add additional Job Managers
- Just works …

Doesn’t this sound a bit like … SAGA



SAGA-Python

A Light-Weight Access Layer for Distributed Computing Infrastructure

Introduction

A Simple API for Grid and Distributed Applications.

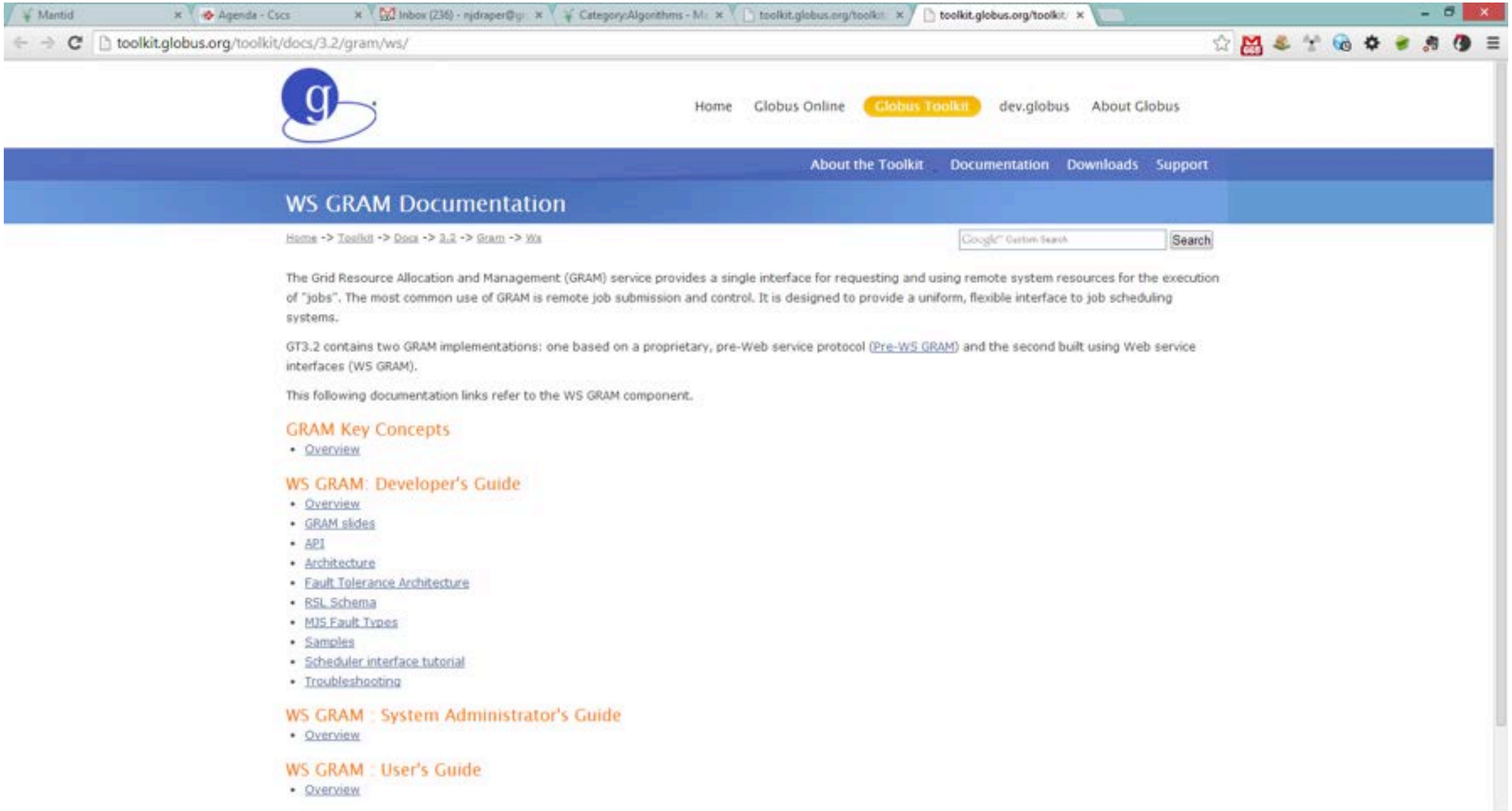
SAGA (*Simple API for Grid Applications*) defines a high-level interface to the most commonly used distributed computing functionality. SAGA provides an access-layer and mechanisms for distributed infrastructure components like job schedulers, file transfer and resource provisioning services. Given the heterogeneity of distributed infrastructure, SAGA provides a much needed interoperability layer that lowers the complexity and improves the simplicity of using distributed infrastructure whilst enhancing the sustainability of distributed applications, services and tools.

SAGA-Python provides a Python module that is compliant with the OGF GFD.90 SAGA specification. Behind the API façade, SAGA-Python implements a flexible *adaptor* architecture. Adaptors are dynamically loadable modules that interface the API with different middleware systems and services. Most application developers use the adaptors that are already part of SAGA-Python, but you can easily implement your own in case your backend system is not supported yet.

Doesn’t this sound a bit like … SAGA

- Removes the need to care what job scheduler is in use. ✓
- Compiles easily on Windows ✗, Mac ?, Linux ✓
- Packages easily for deployment
- APIs ← Development stopped 2 yrs ago
  - C++ ✗, Python ✓
  - others Java
- Needs to handle network proxies, firewalls etc SSH only
- Does not need Admin or special permissions ✓
- Easy to add additional Job Managers ✓
- Just works …?

Doesn’t this sound a bit like … WS_GRAM



WS GRAM Documentation

Home -> Toolkit -> Docs -> 3.2 -> Gram -> Ws

The Grid Resource Allocation and Management (GRAM) service provides a single interface for requesting and using remote system resources for the execution of "jobs". The most common use of GRAM is remote job submission and control. It is designed to provide a uniform, flexible interface to job scheduling systems.

GT3.2 contains two GRAM implementations: one based on a proprietary, pre-Web service protocol (Pre-WS GRAM) and the second built using Web service interfaces (WS GRAM).

This following documentation links refer to the WS GRAM component.

GRAM Key Concepts

- Overview

WS GRAM: Developer's Guide

- Overview
- GRAM slides
- API
- Architecture
- Fault Tolerance Architecture
- RSL Schema
- MJS Fault Types
- Samples
- Scheduler interface tutorial
- Troubleshooting

WS GRAM : System Administrator's Guide

- Overview

WS GRAM : User's Guide

- Overview

Doesn't this sound a bit like …
WS_GRAM

- Removes the need to care what job scheduler is in use. ? **Not local or remote workstation**
- Compiles easily on Windows, Mac, Linux
- Packages easily for deployment
- APIs
  - C++, Python
  - others
- Needs to handle network proxies, fire✓ etc
- Does not need Admin or special permissions✗
- Easy to add additional Job Managers?
- Just works …?

WS_GRAM
Web services ✓

File Transfer
needs GridFTP

MANTID

Further information

- Project web page - www.mantidproject.org

- Many Thanks to the Project Sponsors
- And the development team

ISIS

SNS
SPALLATION NEUTRON SOURCE

Tessella
Technology & Consulting

MANTID DEVELOPER WORKSHOP

Spallation Neutron Source • Oak Ridge National Laboratory • Oak Ridge, Tennessee, USA
January 23, 2014