

The Mantid Project

The challenges of delivering
flexible HPC for novice end users

Nicholas Draper
SOS18



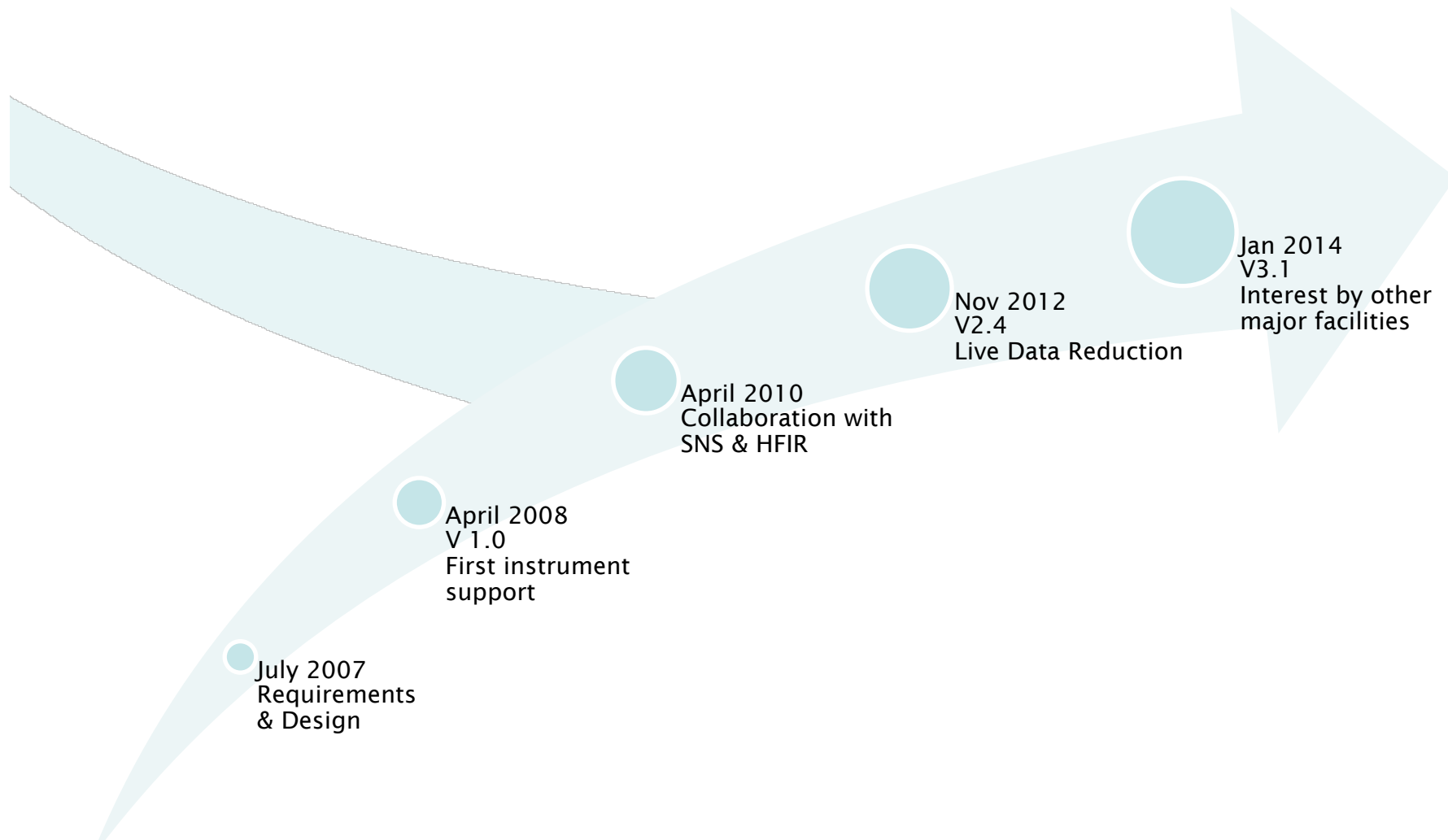
What Is Mantid

- 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





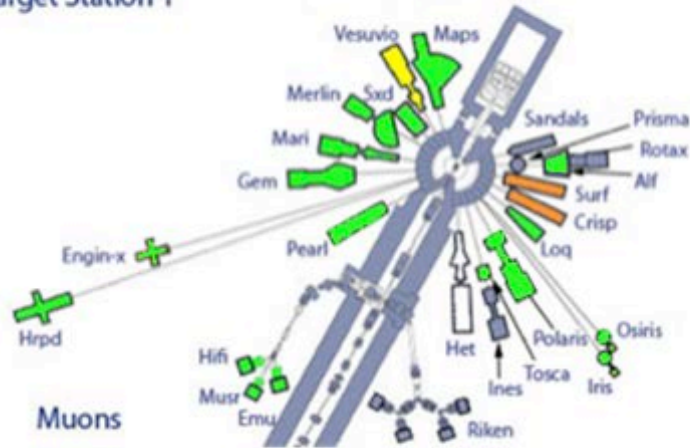
Project History





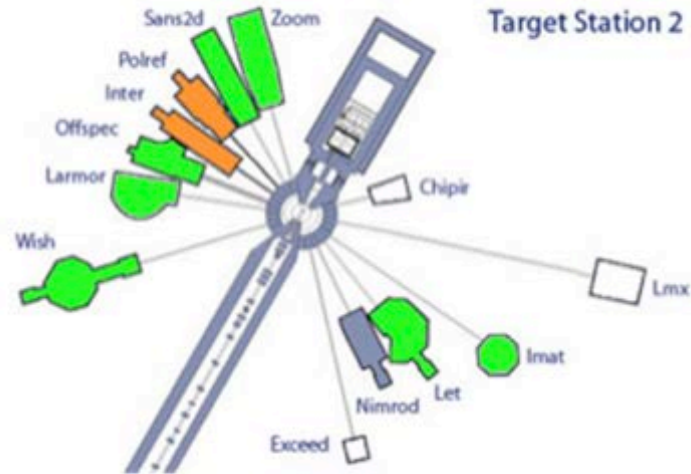
ISIS

Target Station 1

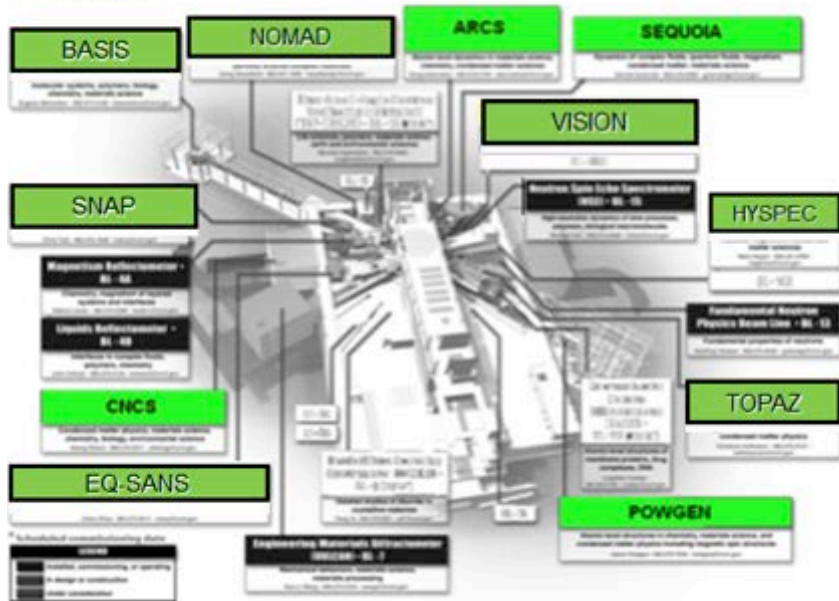


Members

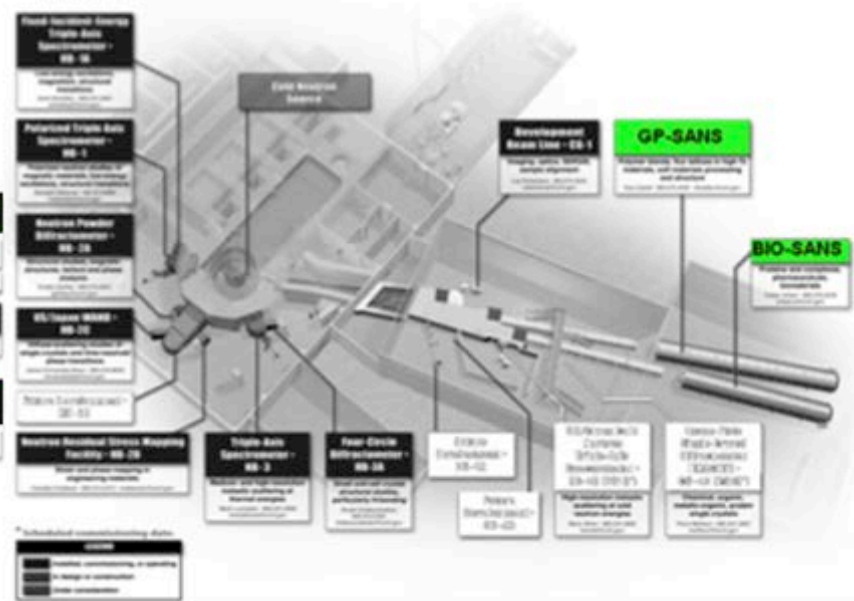
Target Station 2



SNS



HFIR





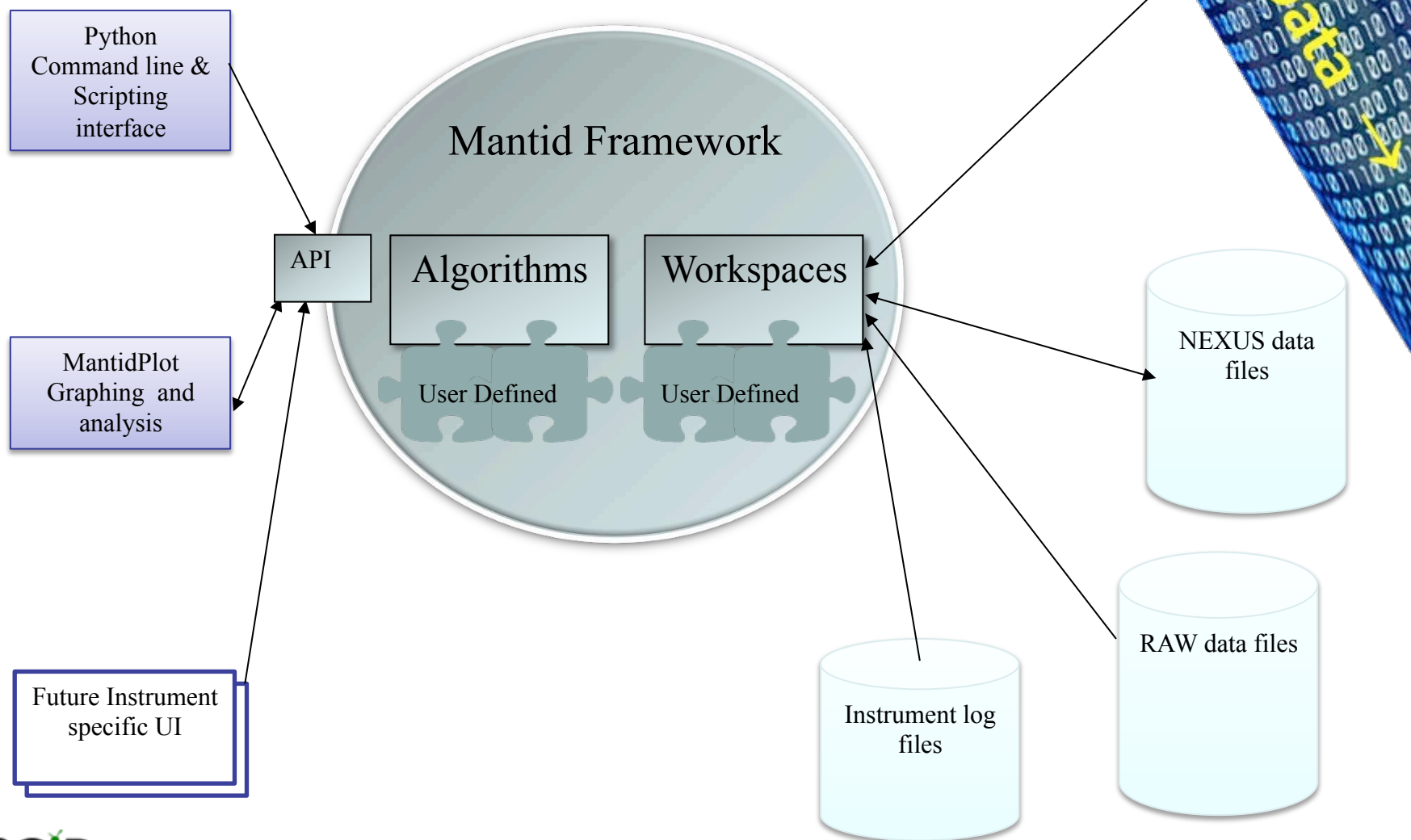
Contributors



MANTiD

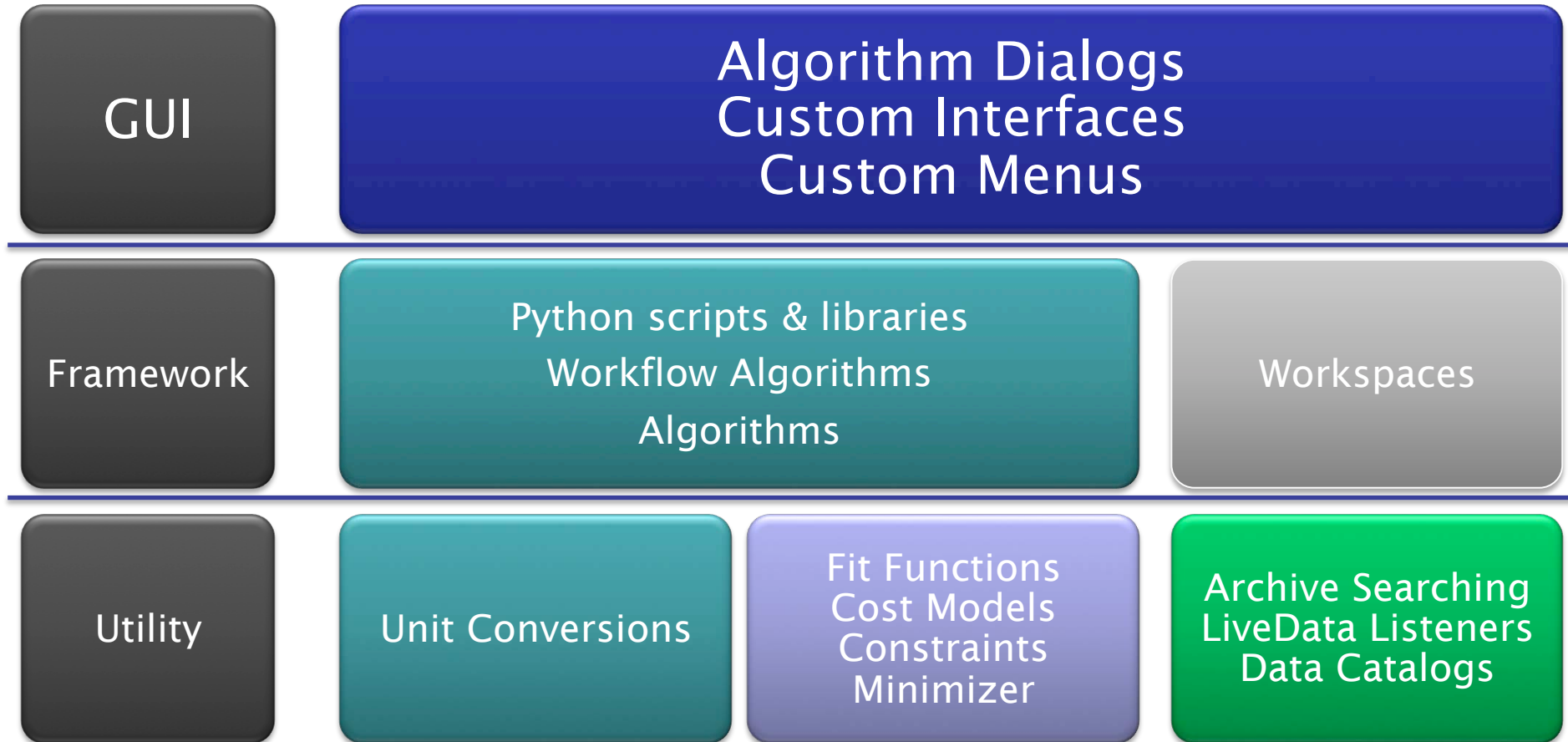


Architectural Design Overview



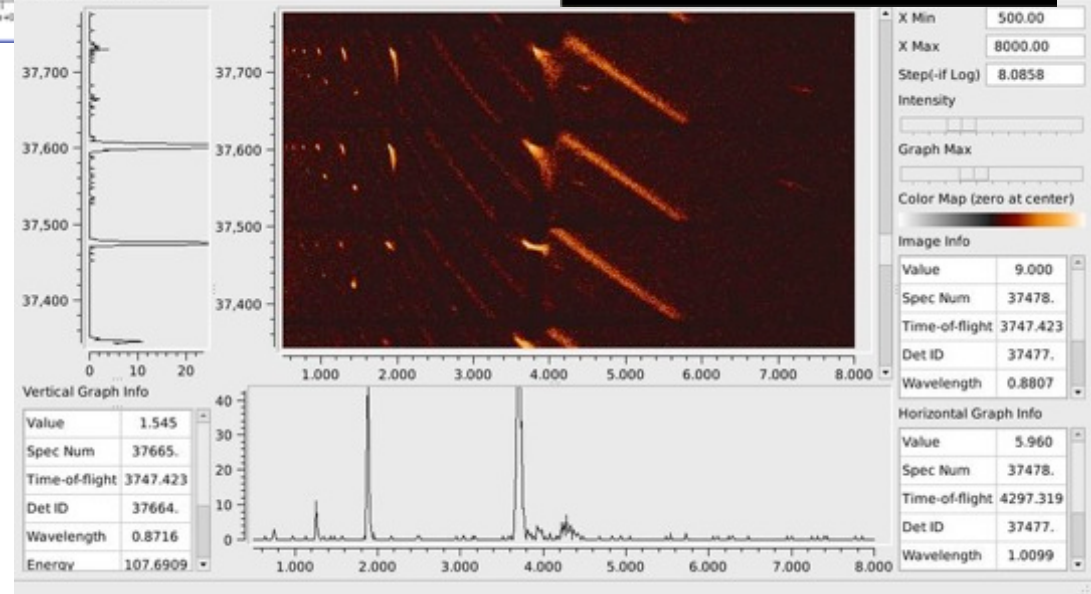
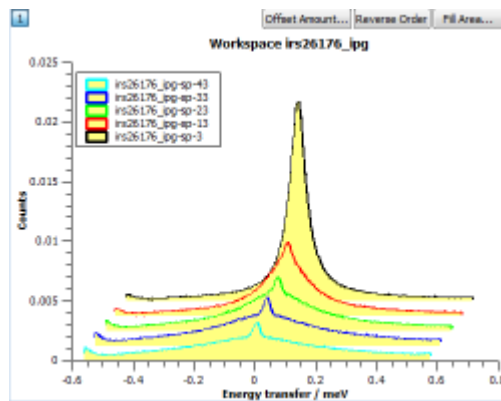
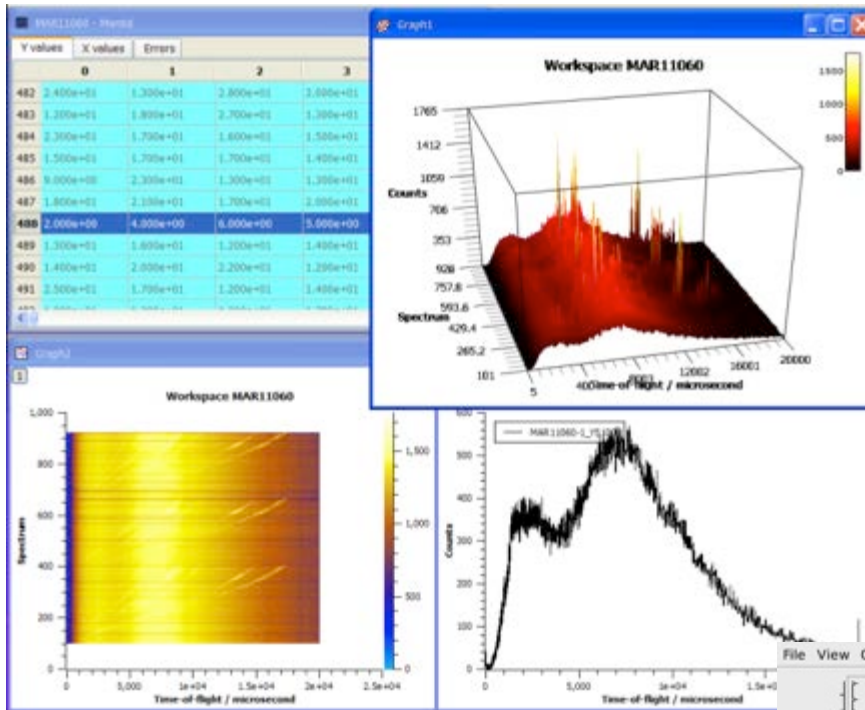


Plug in extensions



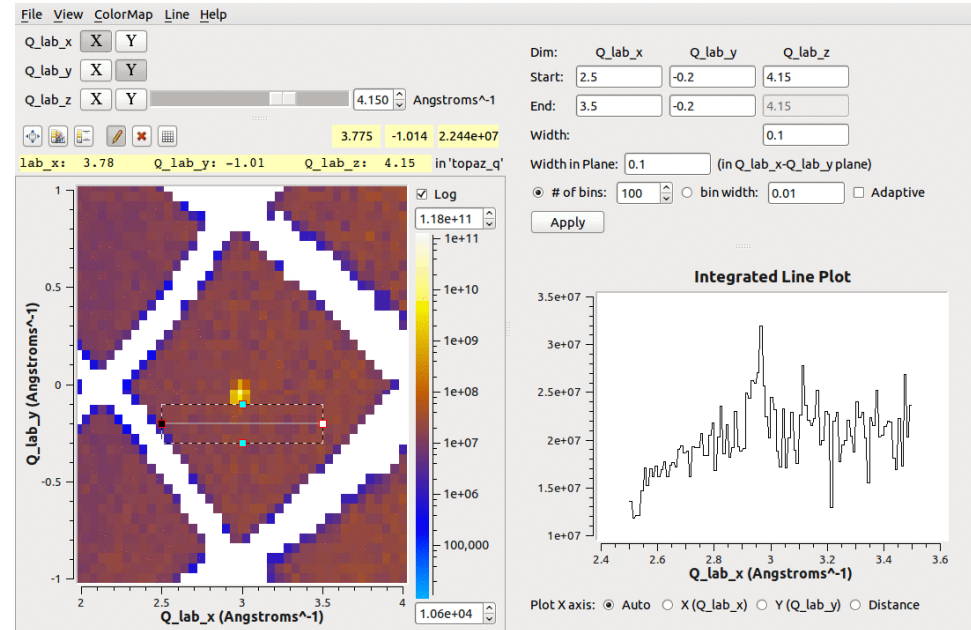
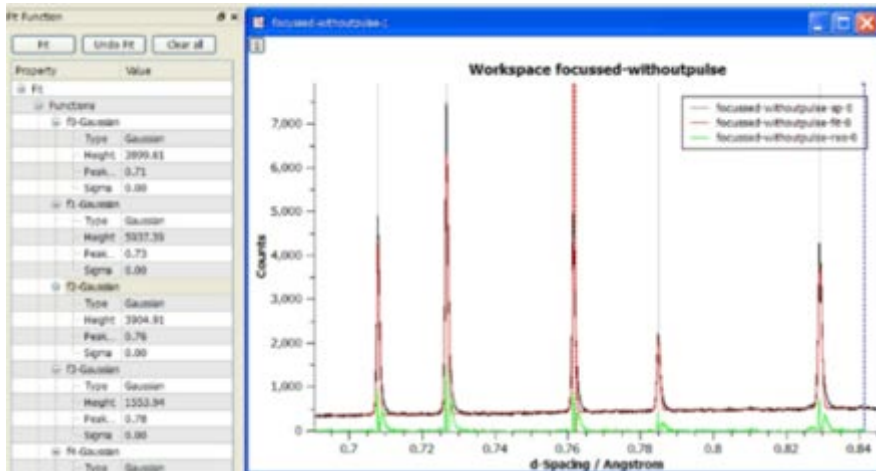
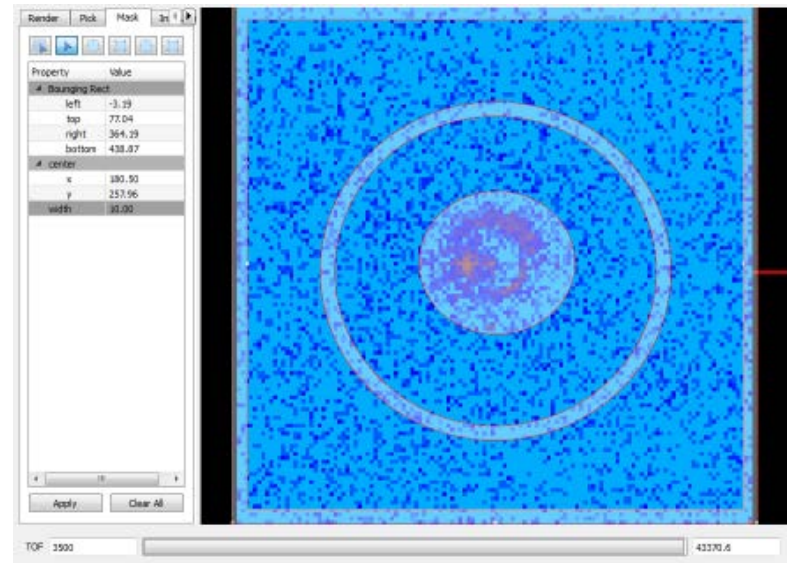
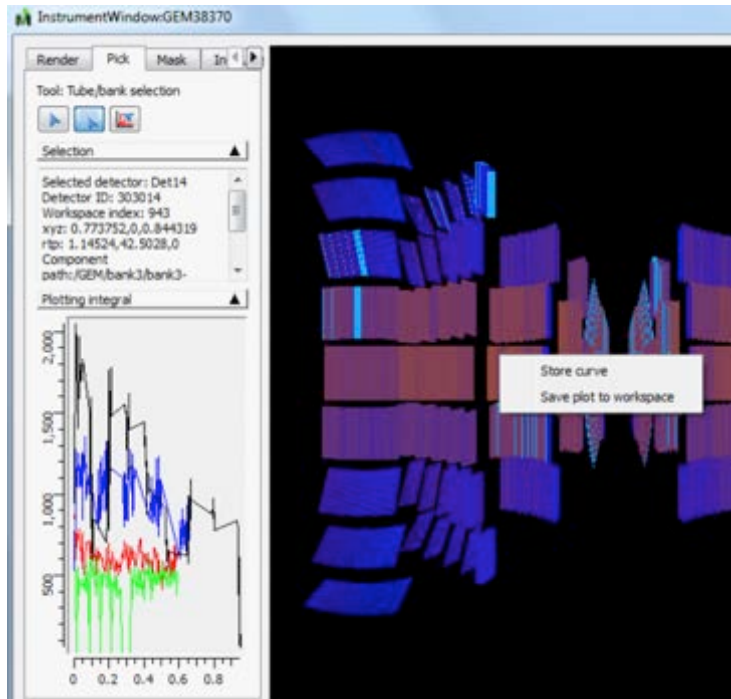


What can we do - Visualization



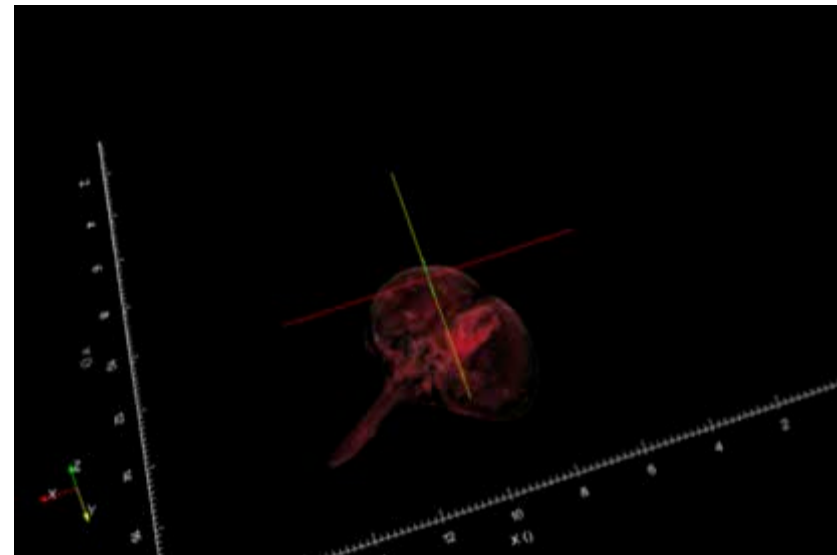
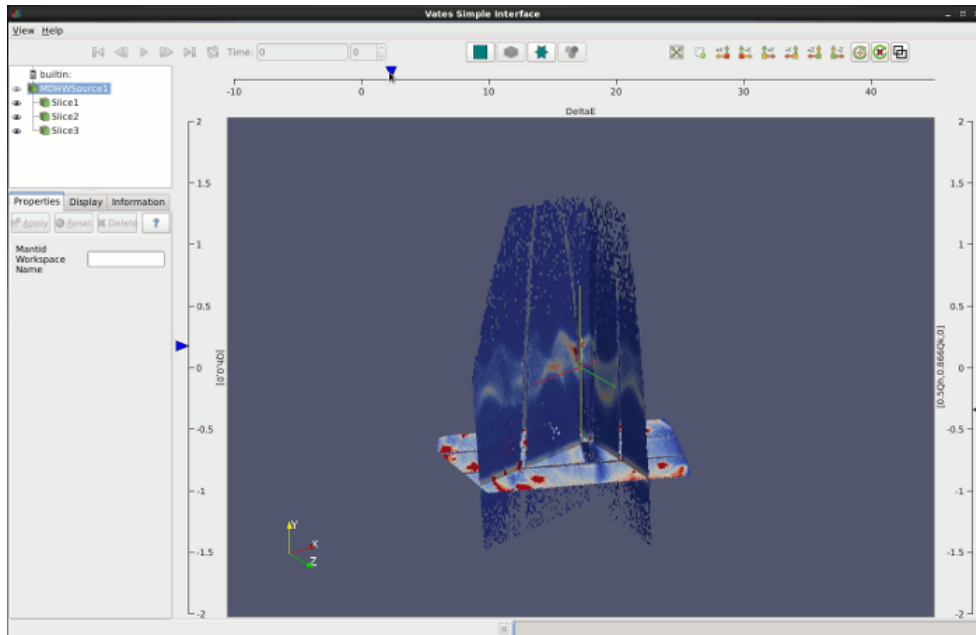
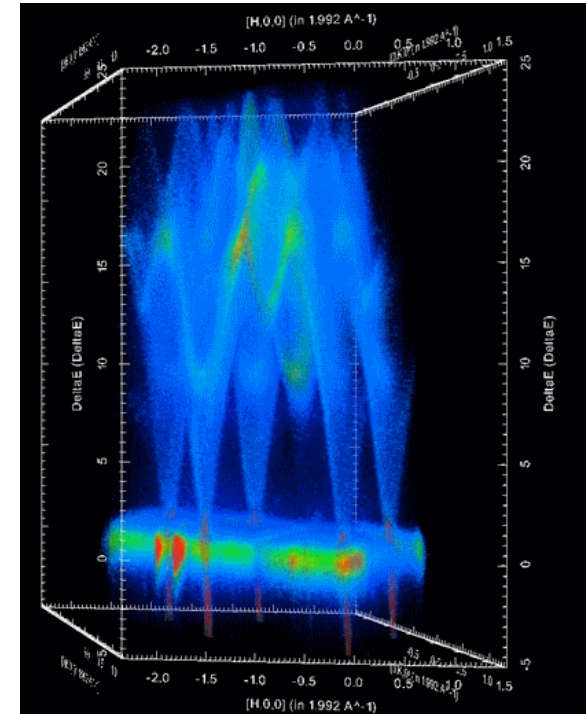
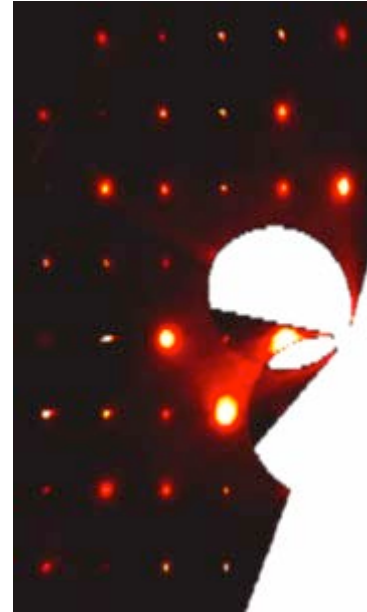
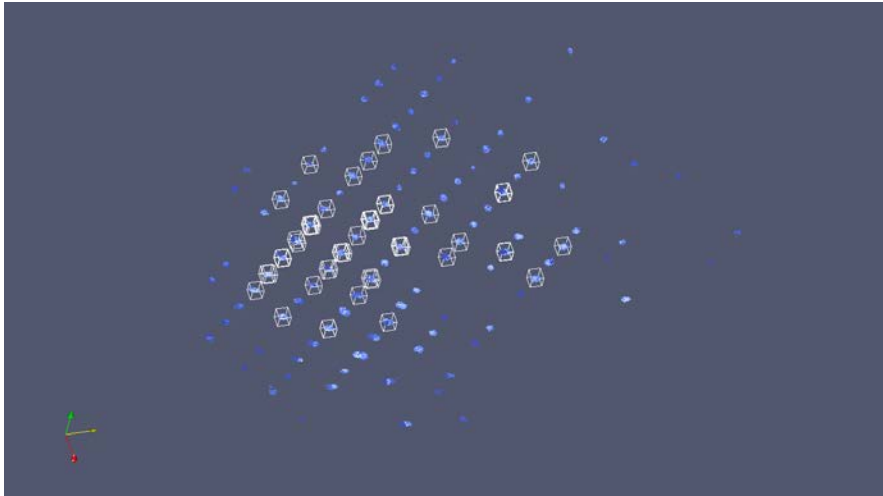


What we can do - Tools





What can we do - Visualization



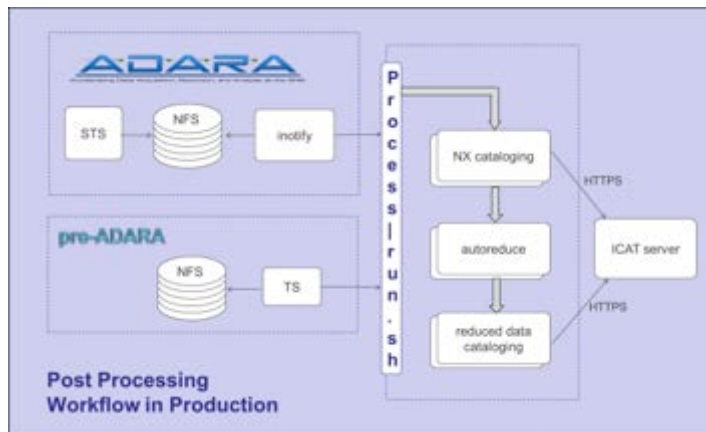


What we can do - Scripting

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

```
1 #####
2 #
3 # Mantid-PST Calculator for POWEN and NORMD
4 #
5 # Version 0.1 Prototype
6 # ( ) None coded for POWEN to
7 #
8 # Last Edit Location
9 # * 20110626 Safe
10 # * 20110626 MBF
11 #
12 #
13 #####
14
15 _DEBUGOUTPUT = True
16 _DEFPATH = "/some/path/Projects/Mantid-Project/Execs/90F-01/"
17
18 binning() = (0.02,0.02,40)
19
20 HSIZE = 1.0E55
21
22 def prototypemain():
23     """ Main method
24     """
25     # samples = ["901_2501_event.raw", "901_2502_event.raw"]
26     # samples = ["901_2548_event.raw", "901_2577_event.raw"]
27     # samples = ["901_2583_event.raw", "901_2594_event.raw"]
28     # bins = ["901_2583_event.raw", "901_2586_event.raw"]
29
30     samples = ["901_2381_event.raw"]
31     varuna = ["901_2349_event.raw"]
32     caruna = ["901_2383_event.raw"]
33     bgruna = ["901_2383_event.raw"]
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
Administrator: C:\Windows\system32\cmd.exe - python
Rebin-[Notice] Rebin started
>>>
>>> print "Rebinned workspace has " + str(rebinned.getNumberHistograms()) + " histograms"
Rebinned workspace has 2584 histograms
>>> print "Spectrum 450's X data size = " + str(len(rebinned.readX(450))) + " bin boundaries"
Spectrum 450's X data size = 4901 bin boundaries
>>> # Perform some algorithms
... testWs = Load("HEI15869.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()) + " histograms"
Rebinned workspace has 2584 histograms
>>> print "Spectrum 450's X data size = " + str(len(rebinned.readX(450))) + " bin boundaries"
Spectrum 450's X data size = 4901 bin boundaries
>>>
```

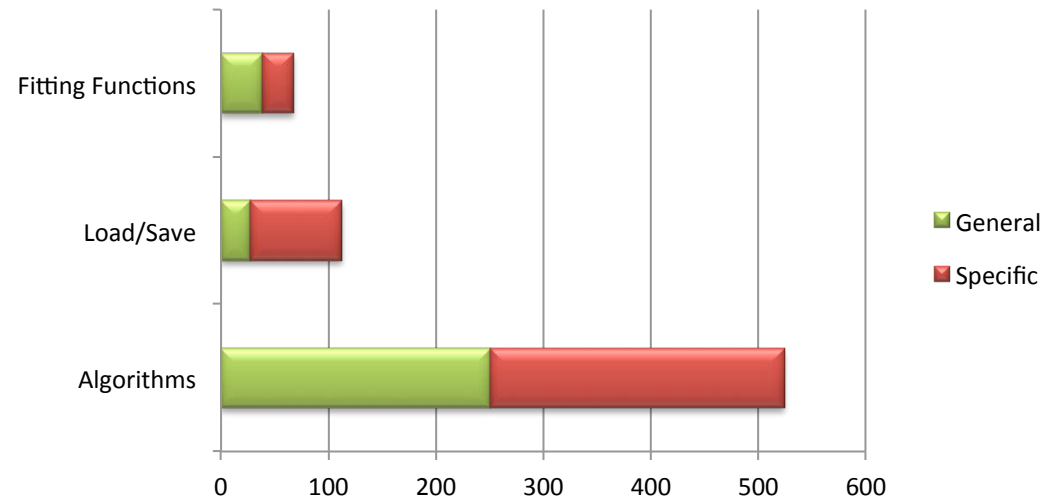




What we can do - Algorithms

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



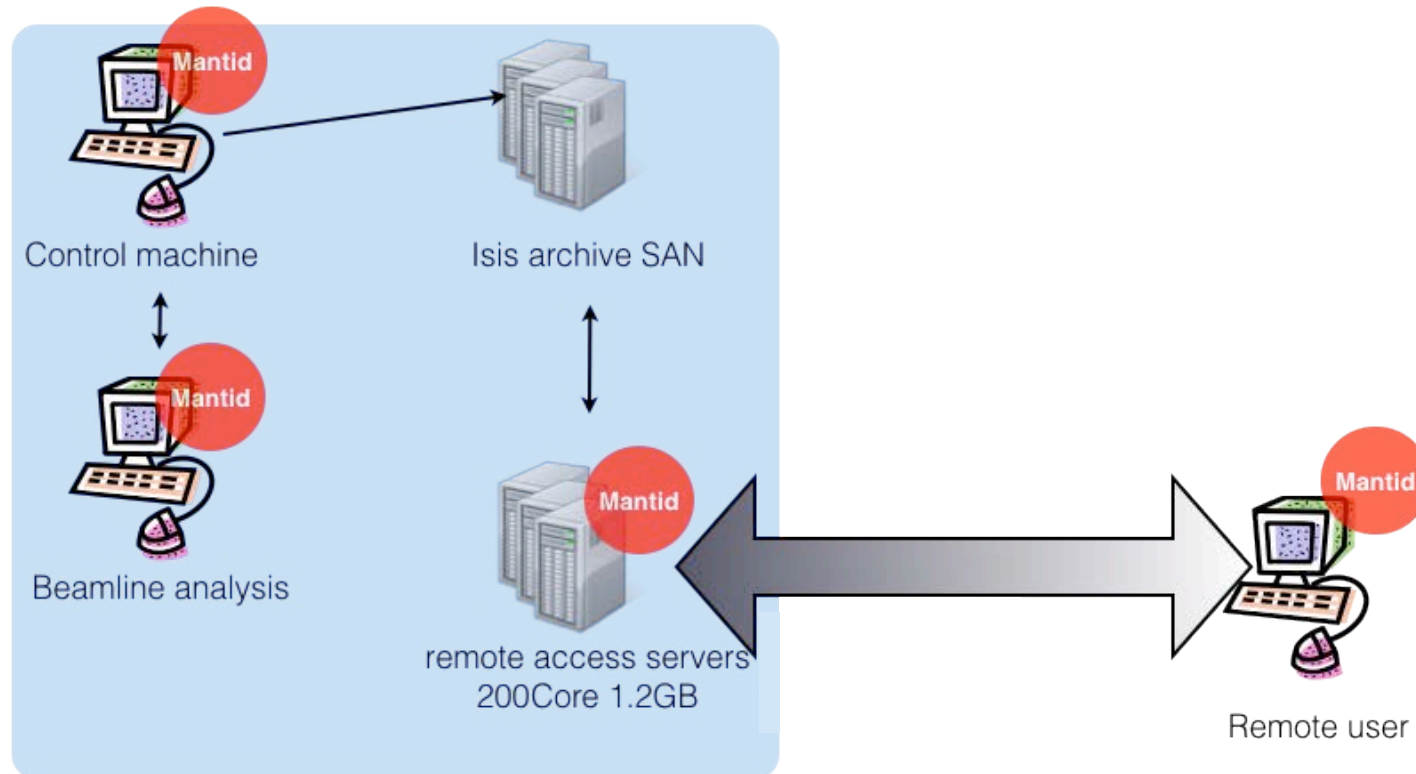
What is a Mantid job?

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



One possible Infrastructure

ISIS user compute infrastructure 1Gbit
cat5





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

The image displays the Mantid software interface. At the top, a window titled 'Facilities.xml' shows XML code for configuring a facility and compute resource. A yellow box highlights the following XML snippet:

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

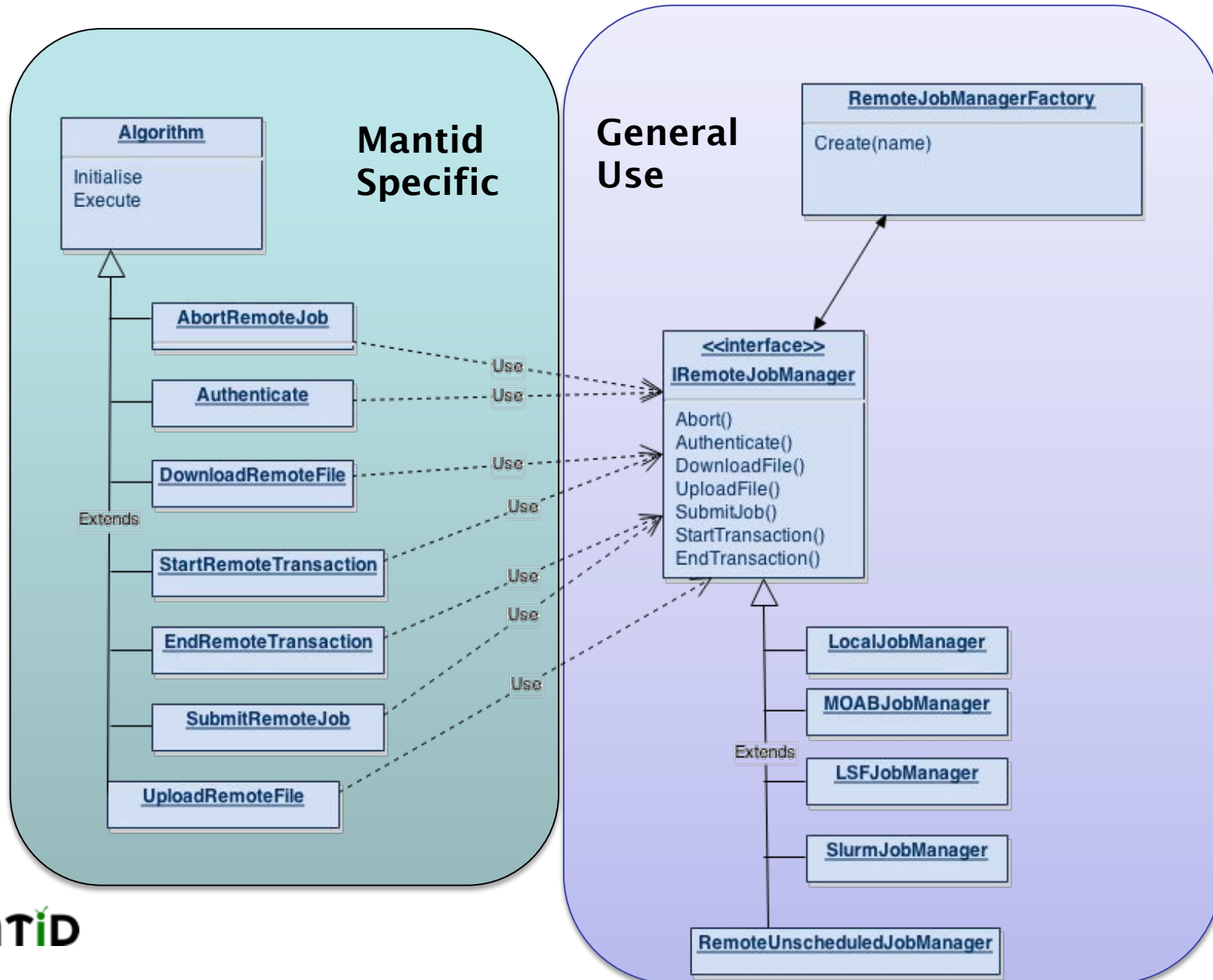
Below the XML editor is the main Mantid application window. It features several tabs: 'Sample Setup', 'Data Corrections', 'Diagnose Detectors', and 'Absolute Units'. The 'Sample Setup' tab is active, showing fields for 'Sample Data' (C:\MantidInstall\data\CNCS_7860_event.nxs), 'Output Workspace Name' (MyData), and 'DetCal File'. There are also sections for 'Incident Energy Calculation' (with fields for Incident Energy Guess, TZero Guess, and Monitor Spectrum IDs) and 'Energy Transfer Range (meV)' (with fields for Low, Width, and High). At the bottom of the application window, a yellow box highlights the 'Send cluster' button.

To the right, a 'Dialog' window is open, showing configuration for a compute resource. It includes a dropdown for 'Compute resource:' (set to 'Fermi'), spinners for 'Number of nodes:' (1) and 'Number of cores:' (16), text boxes for 'Username:' (MyUserName) and 'Password:' (masked with dots), and 'OK' and 'Cancel' buttons. A yellow box highlights the entire dialog window.

Yellow arrows indicate the flow of information: one arrow points from the highlighted XML code to the 'Compute resource:' dropdown in the dialog, and another arrow points from the 'Send cluster' button in the main application window to the dialog box.



Under the bonnet





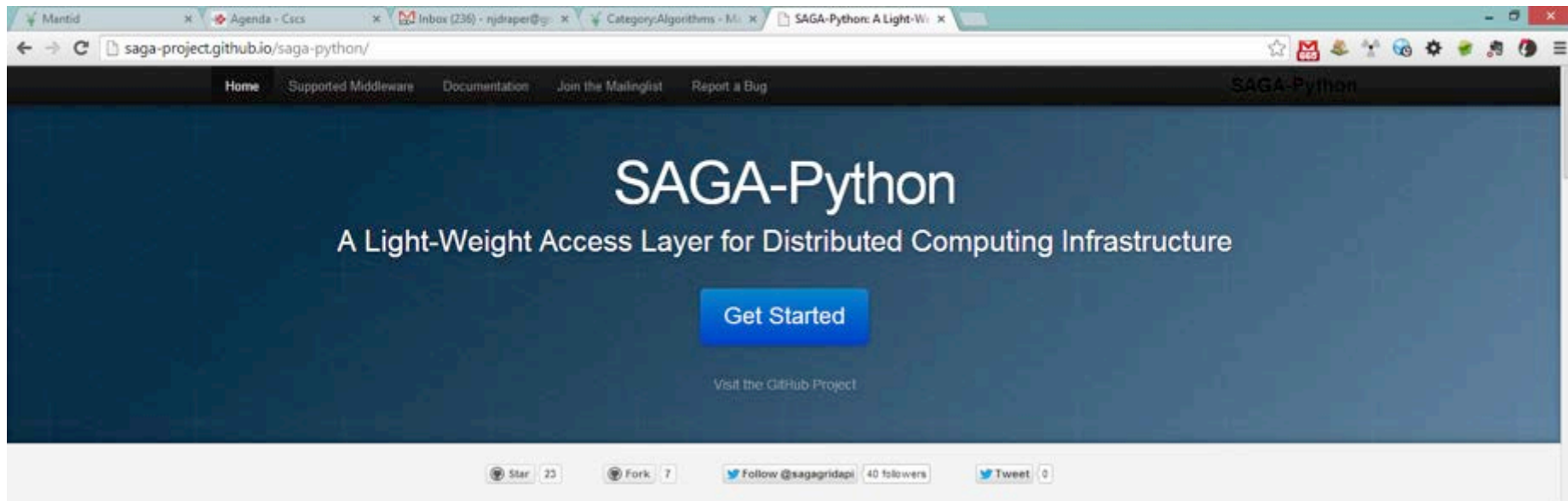
The Ideal Middleware

- 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



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

The screenshot shows a web browser window with the URL `toolkit.globus.org/toolkit/docs/3.2/gram/ws/`. The page features a blue header with the Globus logo and navigation links: Home, Globus Online, Globus Toolkit (highlighted), dev.globus, and About Globus. Below the header is a sub-header with links: About the Toolkit, Documentation, Downloads, and Support. The main content area is titled "WS GRAM Documentation" and includes a breadcrumb trail: Home -> Toolkit -> Docs -> 3.2 -> Gram -> Ws. A search bar is present with the text "Google Custom Search" and a "Search" button. The text on the page describes the Grid Resource Allocation and Management (GRAM) service, its common use for remote job submission, and mentions two implementations: a proprietary pre-Web service protocol (Pre-WS GRAM) and a Web service interface (WS GRAM). It also provides links to documentation for WS GRAM components, including Key Concepts, Developer's Guide, System Administrator's Guide, and User's Guide, each with an "Overview" link.

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

Google Custom Search

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).

The 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](#)

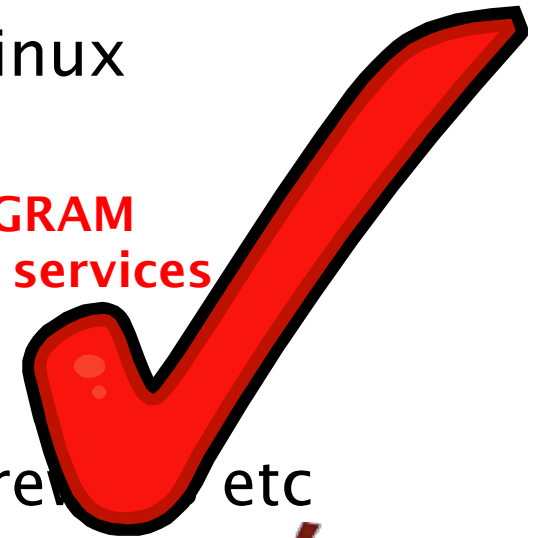


Doesn't this sound a bit like ...

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, firewalls etc
- Does not need Admin or special permissions **X**
- Easy to add additional Job Managers? **?**
- Just works ...? **?**

WS_GRAM
Web services



**File Transfer
needs GridFTP**



Further information

- Project web page - www.mantidproject.org
- Many Thanks to the Project Sponsors
- And the development team

