Jupyter Examples for Neutron Scattering Facilities

Jonathan Taylor
European Spallation Source

Where is my Data?

How do I get it?

How do I look at it?

What does it mean?

Is it right?

REDUCING THE BOTTLENECK EFFECT:

"What we're trying to do here is **expedite the time to discovery.** Scientists should be able to **focus on their science** without having to become experts in data management."

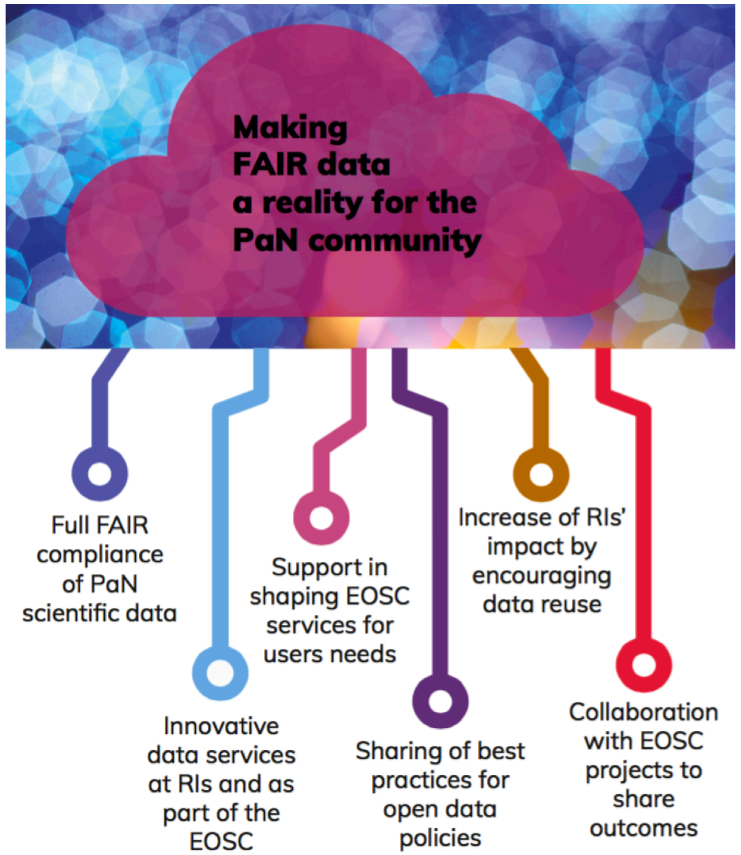
—Shawn McKee
research scientist in physics

* Facility users are not expected to be experts in scientific computing

Summary

- Jupyter is a powerful tool for scientific computing at user facilities.
- Facility users need a range of tools
- Facility users have a range of experience
- Jupyter provides a unique data management function - can capture rich meta data
- Some examples of what can be developed in a short (ish) space of time.

PaNOSC



Analyse

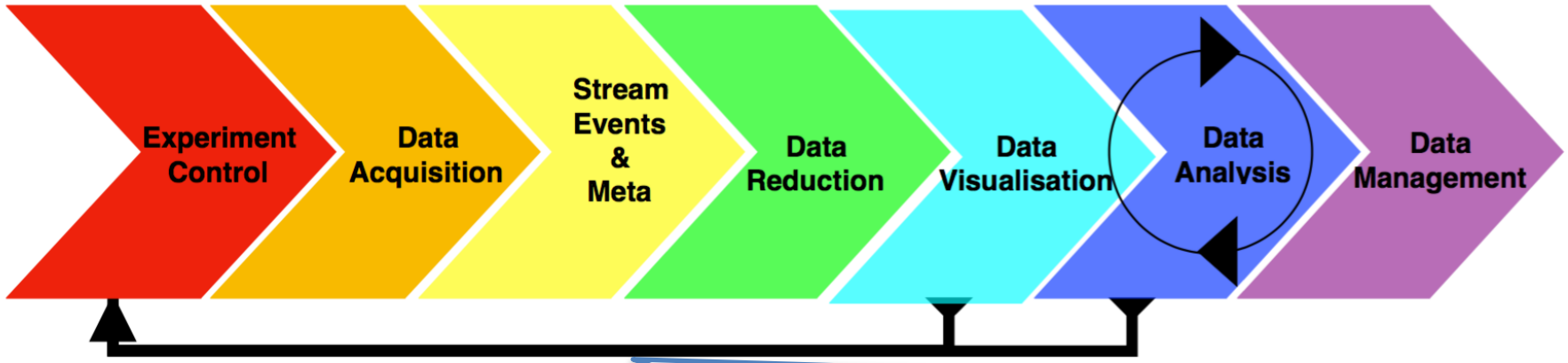
- PaNOSC (Photon and Neutron Open Science Cloud) links petabytes of data between CERIC, ELI, ESRF, ESS, ILL, XFEL
- Scientists can analyse and visualise cross-facility data online with Jupyter notebooks
- Hit the ground running, no need to wait for lengthy data downloads, or slow software installation
- Be ready for EOSC, the European Open Science Cloud

ExPaNDS
European Open Science Cloud Photon and Neutron Data Services

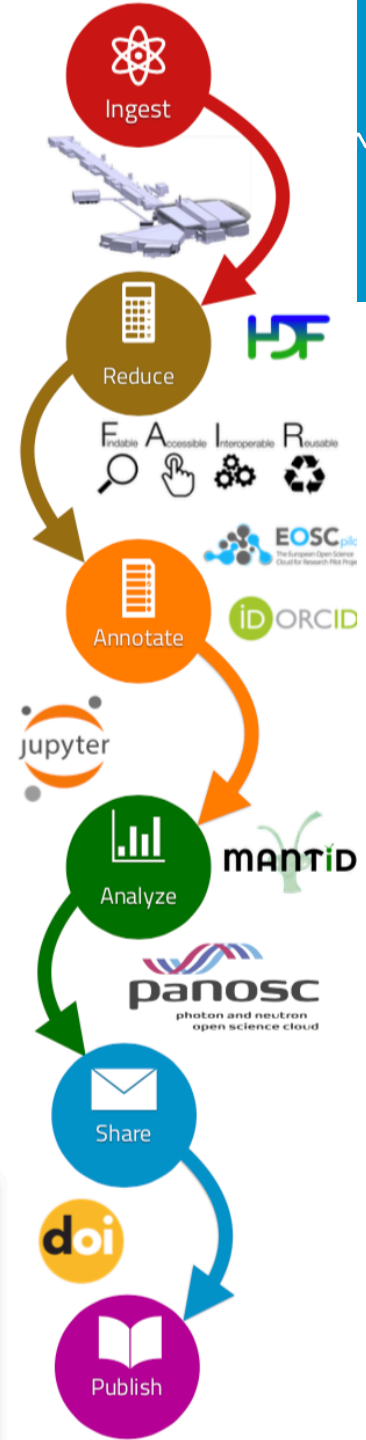
Vision- PaNOSC will help fastrack new Photon and Neutron facilities, improve data services for existing users, create a new class of virtual users of open data, help build and integrate the PaN RIs to the **EOSC**.

delivery consortium

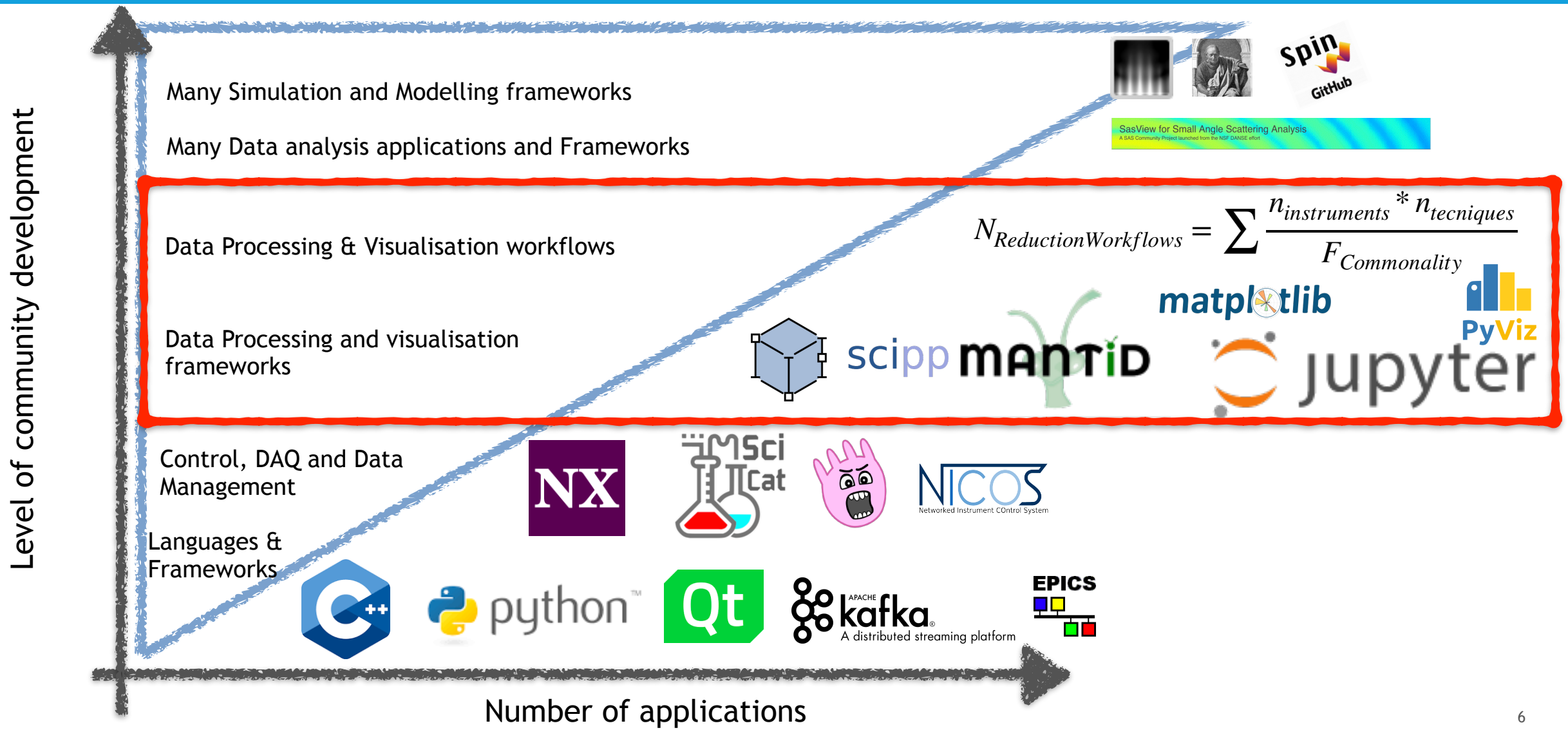
Data Pipelines



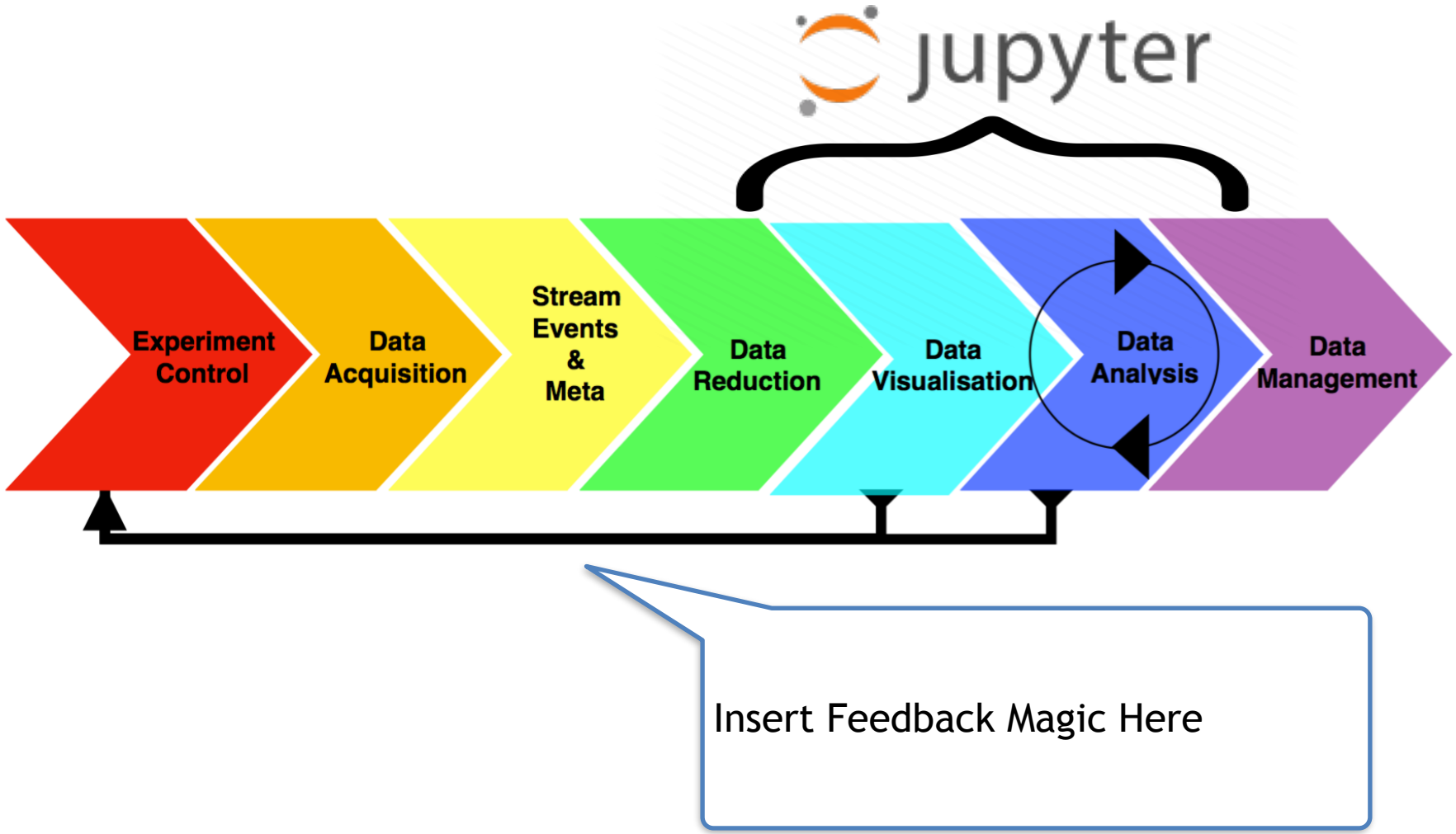
Insert Feedback Magic Here

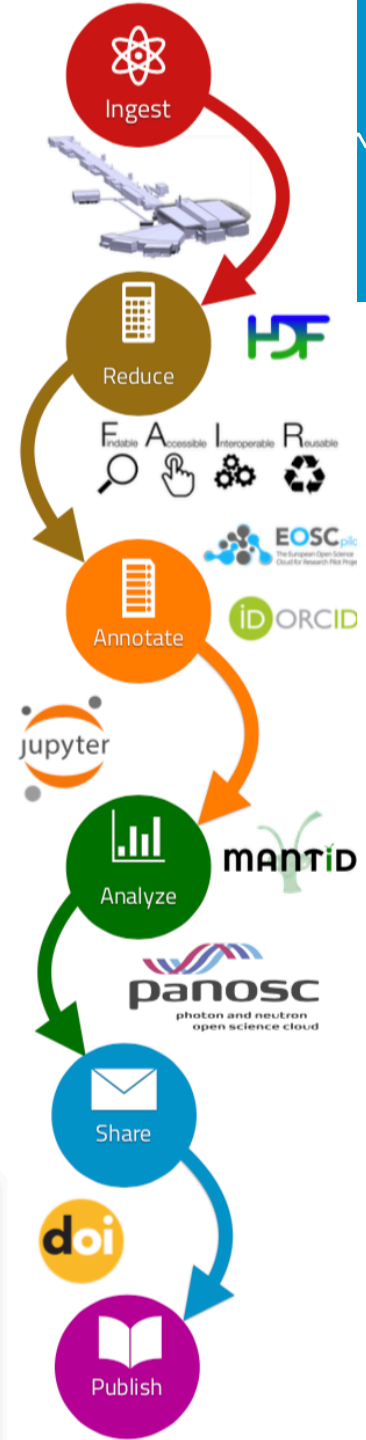


Scientific Application Stack ESS (& Neutron scattering)



Data Pipelines





The jupyter EcoSystem

EUROPEAN SPALLATION SOURCE

MENU nature International journal of science

Subscribe Search Login

TOOLBOX · 30 OCTOBER 2018

Why Jupyter is data scientists' computational notebook of choice

An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.

Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

Language of choice

Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.

Share notebooks

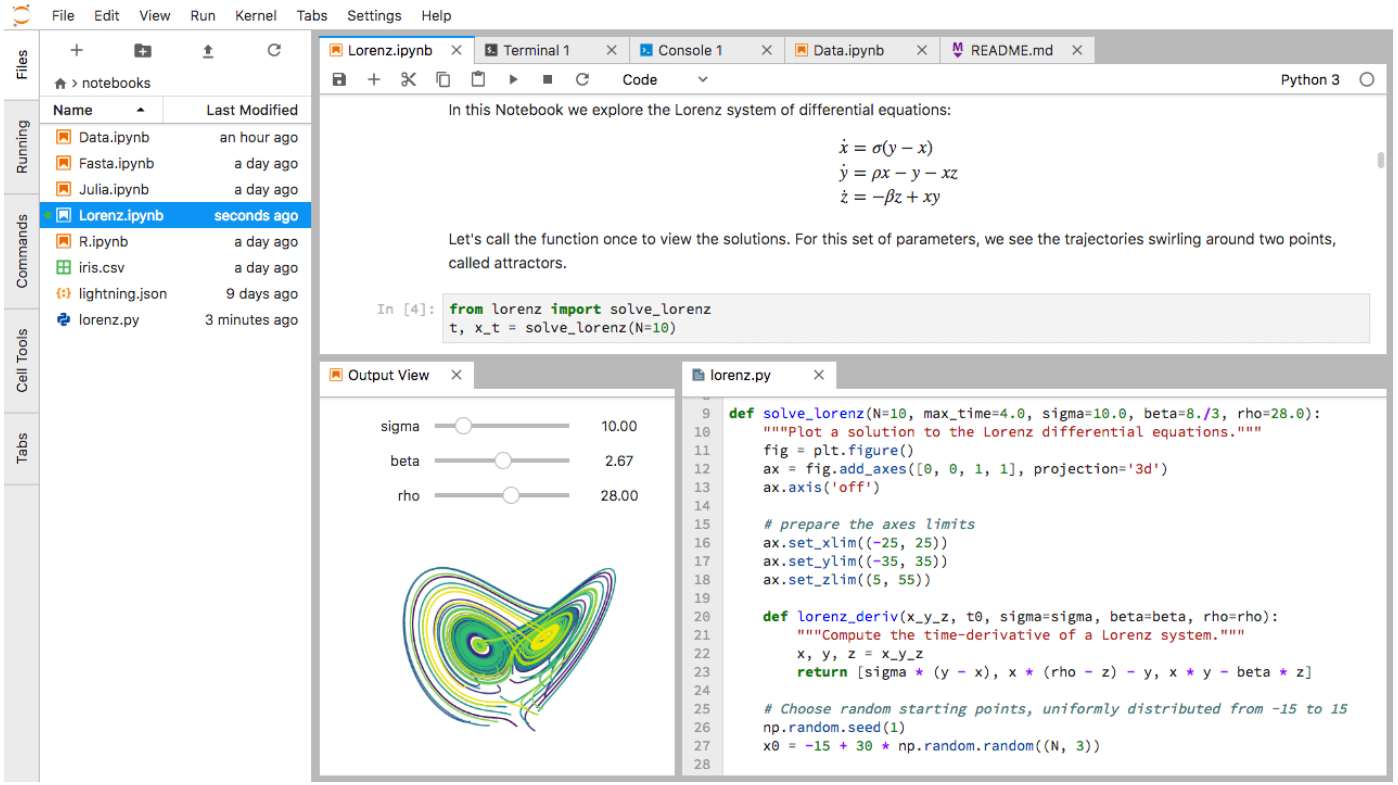
Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.

Interactive output

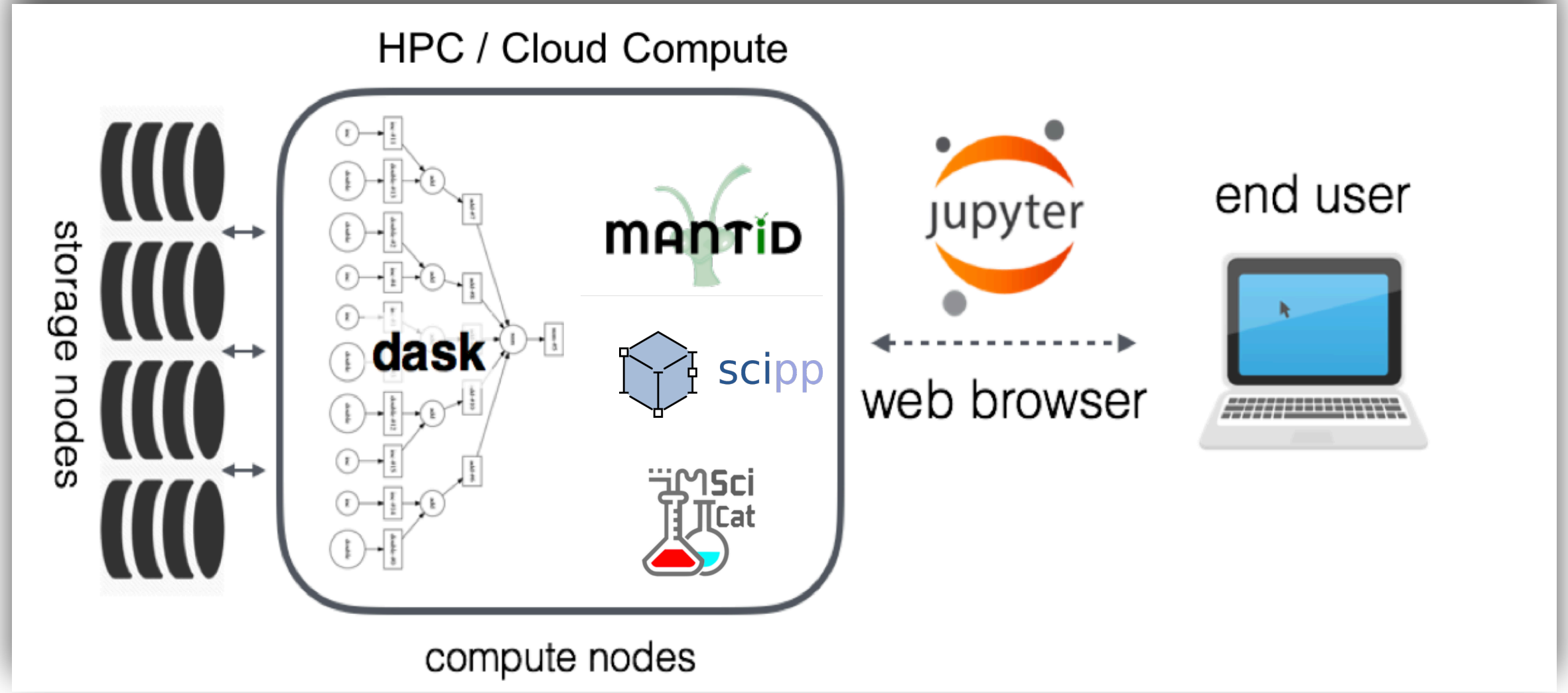
Your code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.

Advantages.

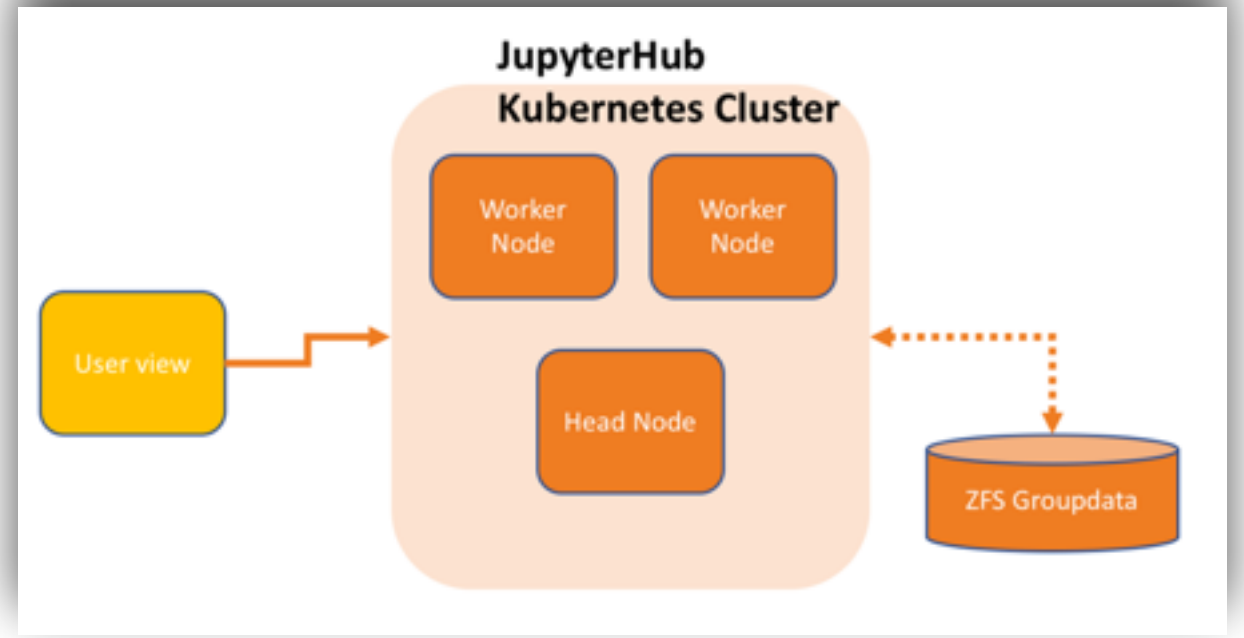
- Clean Flexible Architecture
  - Compute & storage can be close
  - No need to transfer large data volumes
  - Facility and user developed methods can coexist
- Web Interface
  - Good UX
  - High level of community commitment / development
  - Code development and execution
  - 1D, 2D, 3D Visualisation of data
- Data Management / FAIR
  - Persistence for analysis workflow & rich meta data
    - Text, Math, Citations, Links …
  - Interfaces to catalogue both directions and storage



How could this look for ESS

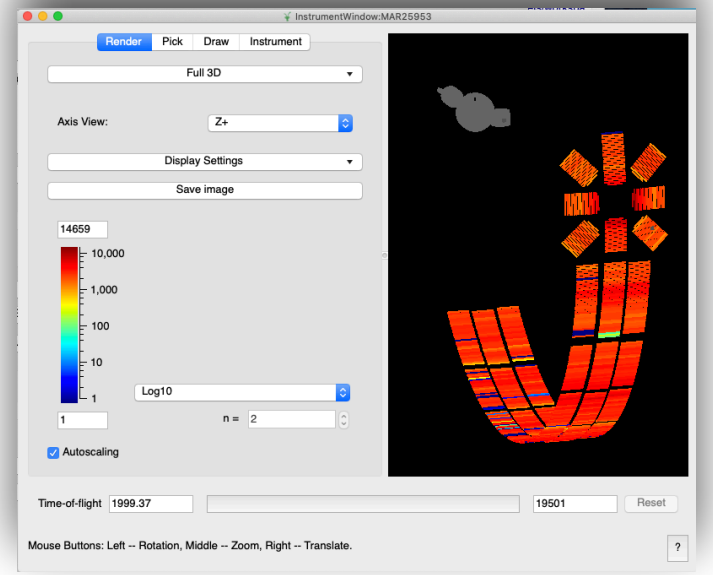


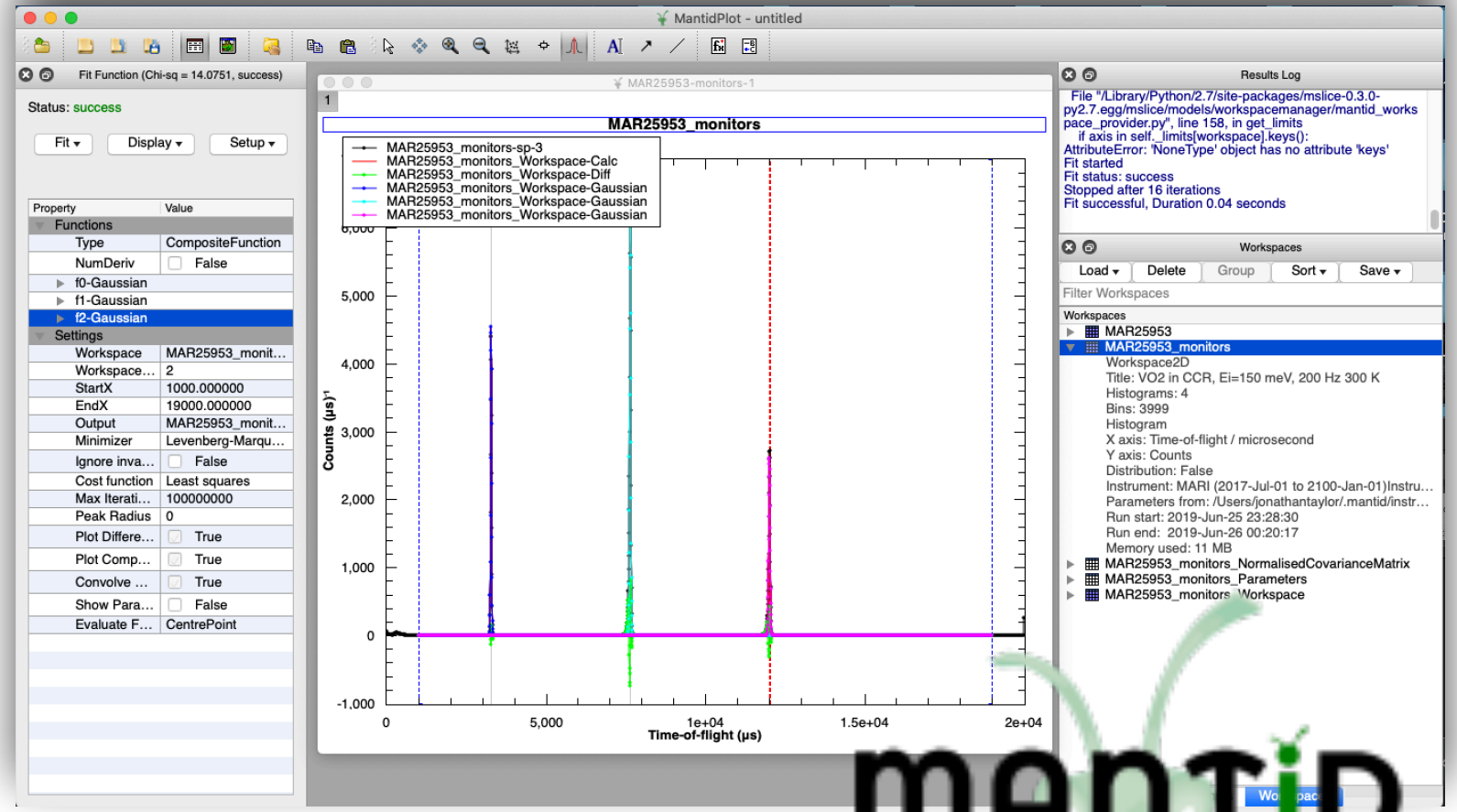
Jupyter Hub at DMSC



What Are (Most) Users Comfortable With.

- GUI for interacting with data.
- Relies on code and data format to persist data history
- No opportunity to persist Rich information
- Some complexity with provisioning architecture





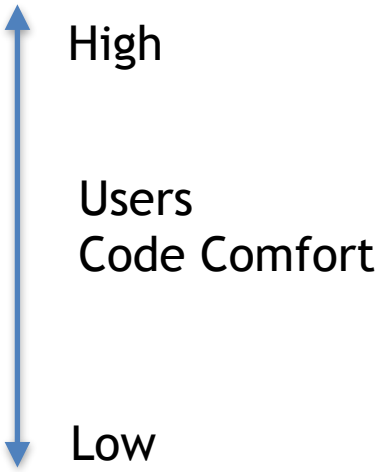
Questions

- Not all users are confident with a code / script based environment

- Can it be used as a direct replacement for traditional GUI delivery for facility reduction & analysis?

Jupyter examples for Neutron data.

- Some example concepts for deploying data processing using Jupyter for facility users
- Use Standard Notebooks - Populated with pre defined functions and scripts
- Decorate a note book with widgets to steer the user in a familiar directions
- Execute Notebooks as web applications to provide a GUI look and feel



Examples are all from MARI @ ISIS

- Code cell allows data reduction from Time of Flight to Energy Transfer

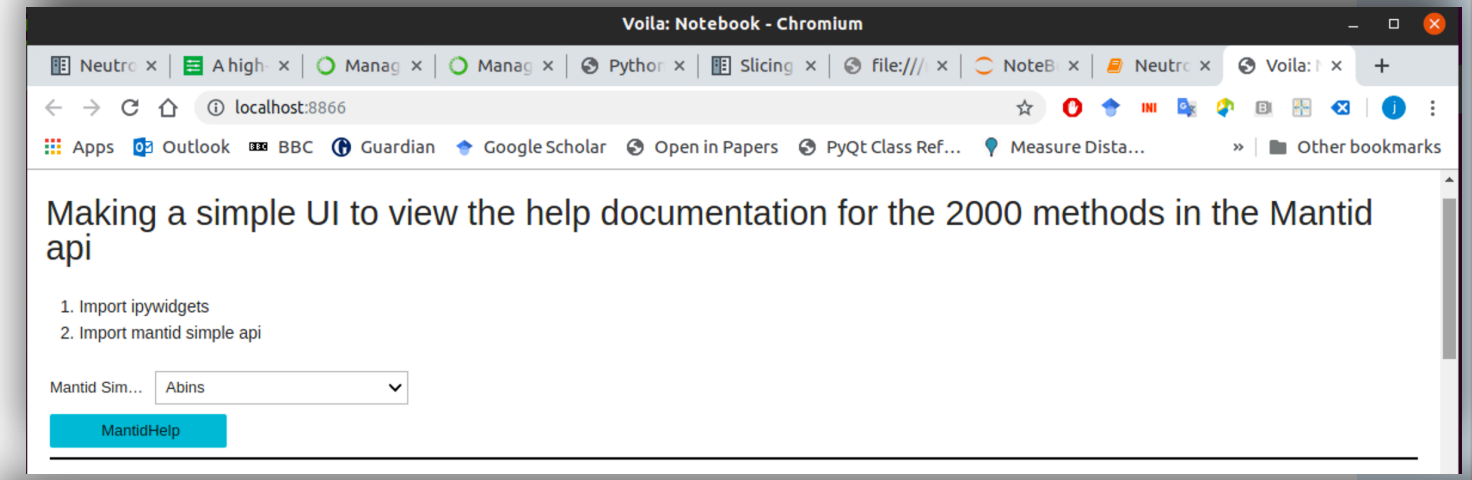
In [9]:

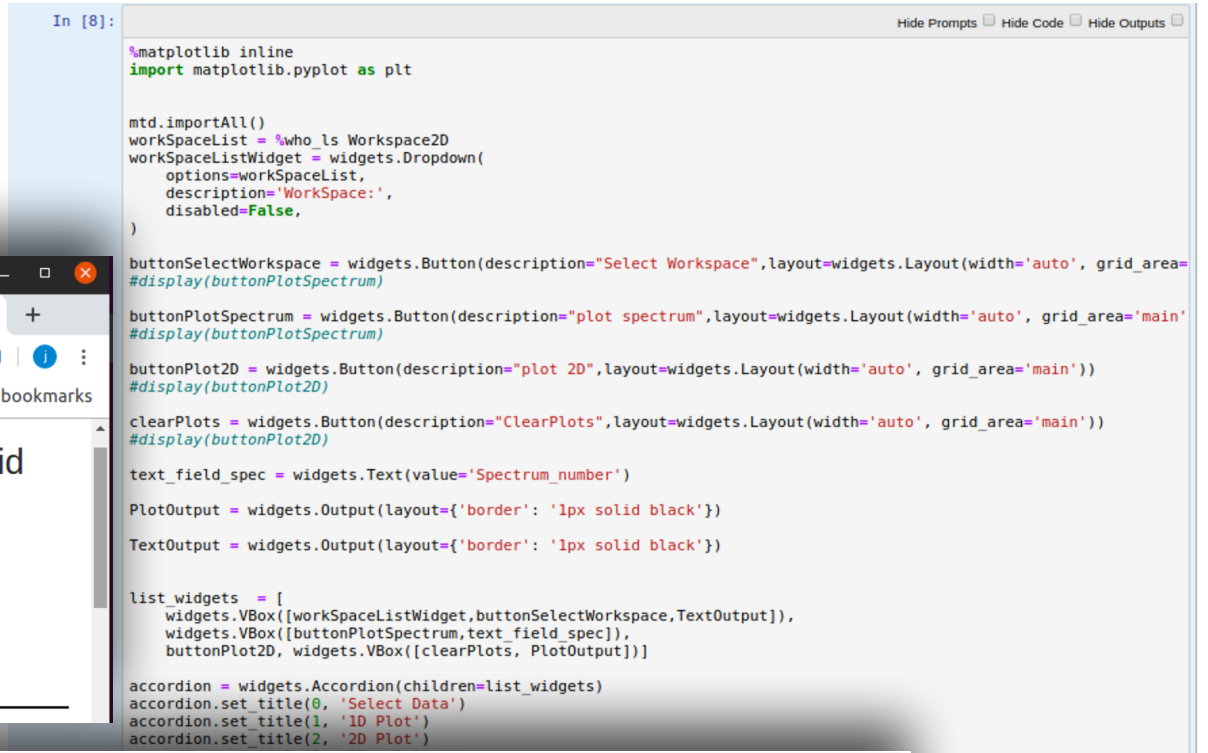
Hide Prompts ☐ Hide Code ☐ Hide Outputs ☐

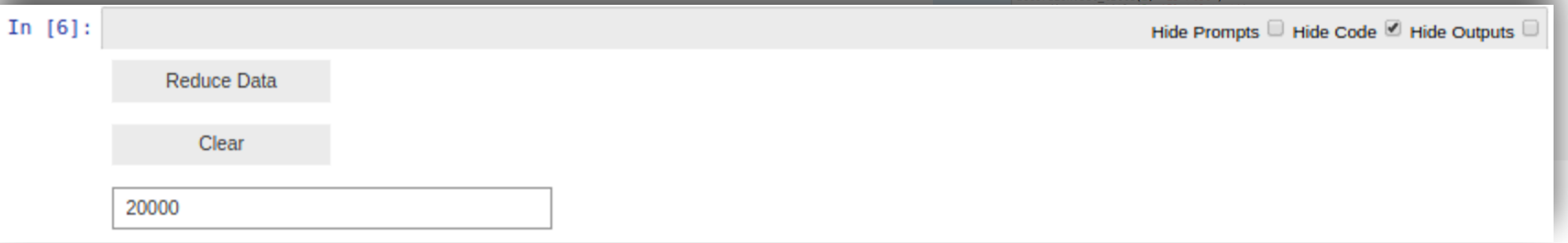
```
from Direct.DirectEnergyConversion import *
import Direct.dgreduce as dgreduce

iliad_setup=dgreduce.setup
iliad=dgreduce.arb_units
iliad_abs=dgreduce.abs_units

iliad_reducer = dgreduce.getReducer

#@out.capture()
def reduceSimple(runNumber):
    inst='mar'
    iliad_setup(inst)
    ext='.raw'
    #mapfile='mari_res2013'
    mapfile=pathToMapFile+'mari_121_cor'

    #det_cal_file must be specified if the reduction sends out put to a workpsace
    cal_file=pathToData+'MAR19717.raw'
    #hard mask file
    #mask_file='/home/mari/Users/taylor/mask.msk'
    #load vanadium file
    whitebeamfile=pathToData+'MAR19717.raw'
    LoadRaw(Filename=whitebeamfile,OutputWorkspace="wb_wksp",LoadLogFiles="0")

    runs=pathToData+'MAR'+str(runNumber)+'.raw'; ei=12; rebin_params='-10,.05,11'; sum=False
    LoadRaw(Filename=runs,OutputWorkspace="run_wksp",LoadLogFiles="0")
    w1=iliad("wb_wksp","run_wksp",ei,rebin_params,mapfile,fixei=False,det_cal_file=cal_file,norm_method='current')
```

Data processing - Standard scripts

- Decorate with widgets
- Run data reduction
- Visualise data



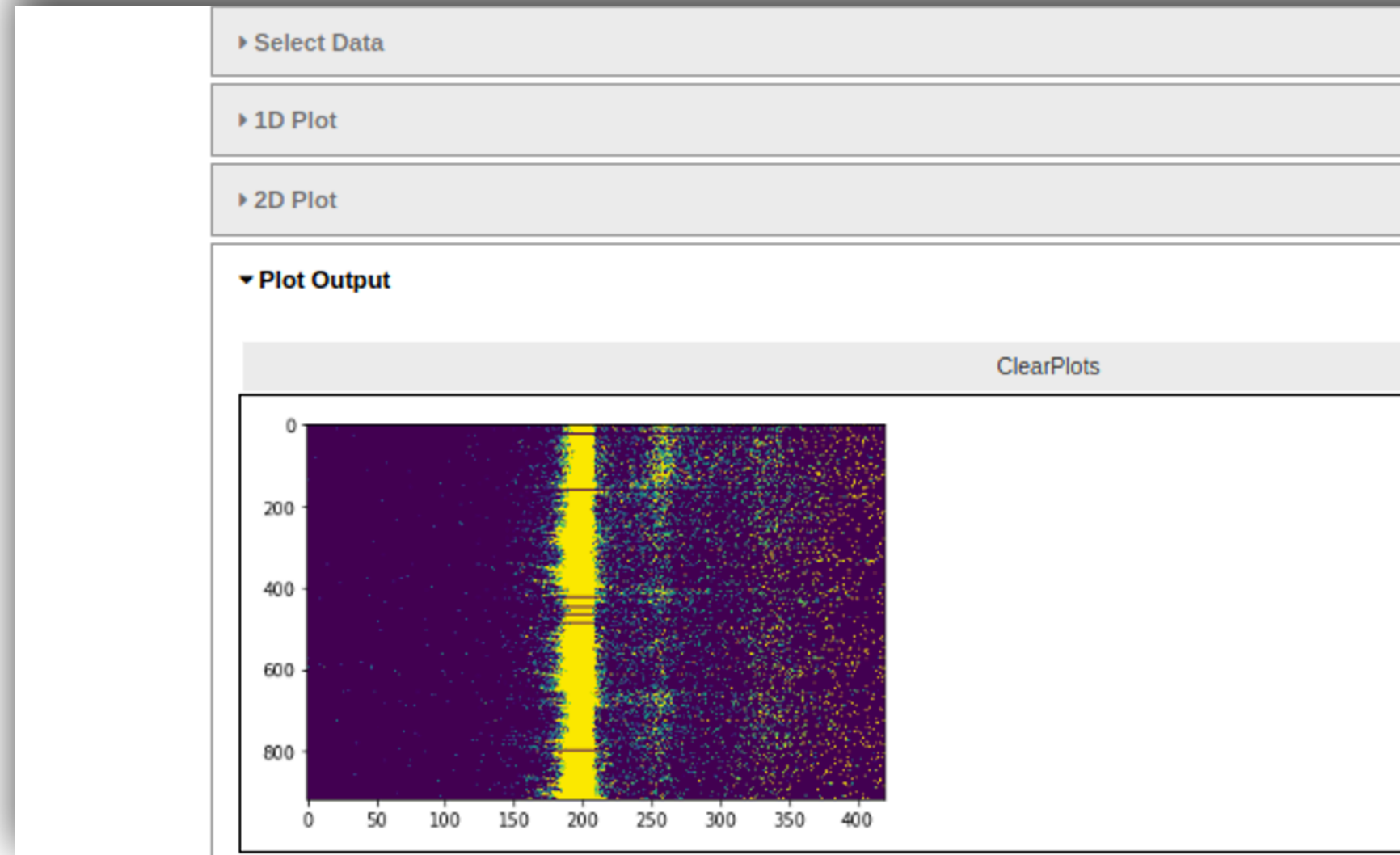




Interactive Data visualisation

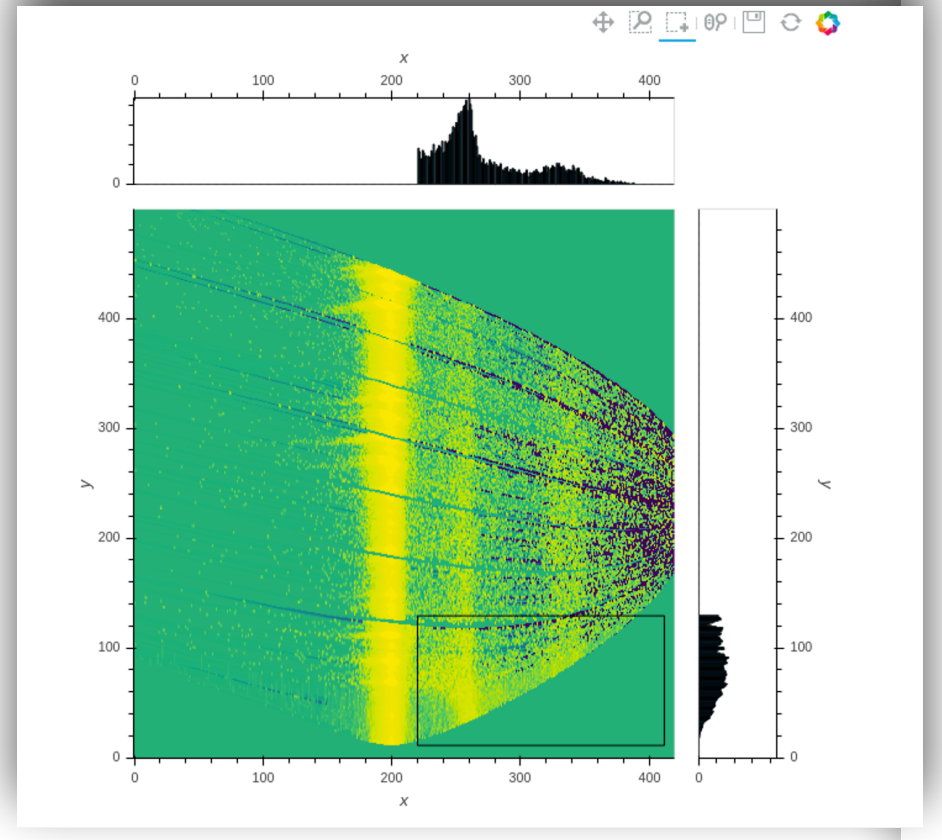
- Data Selection

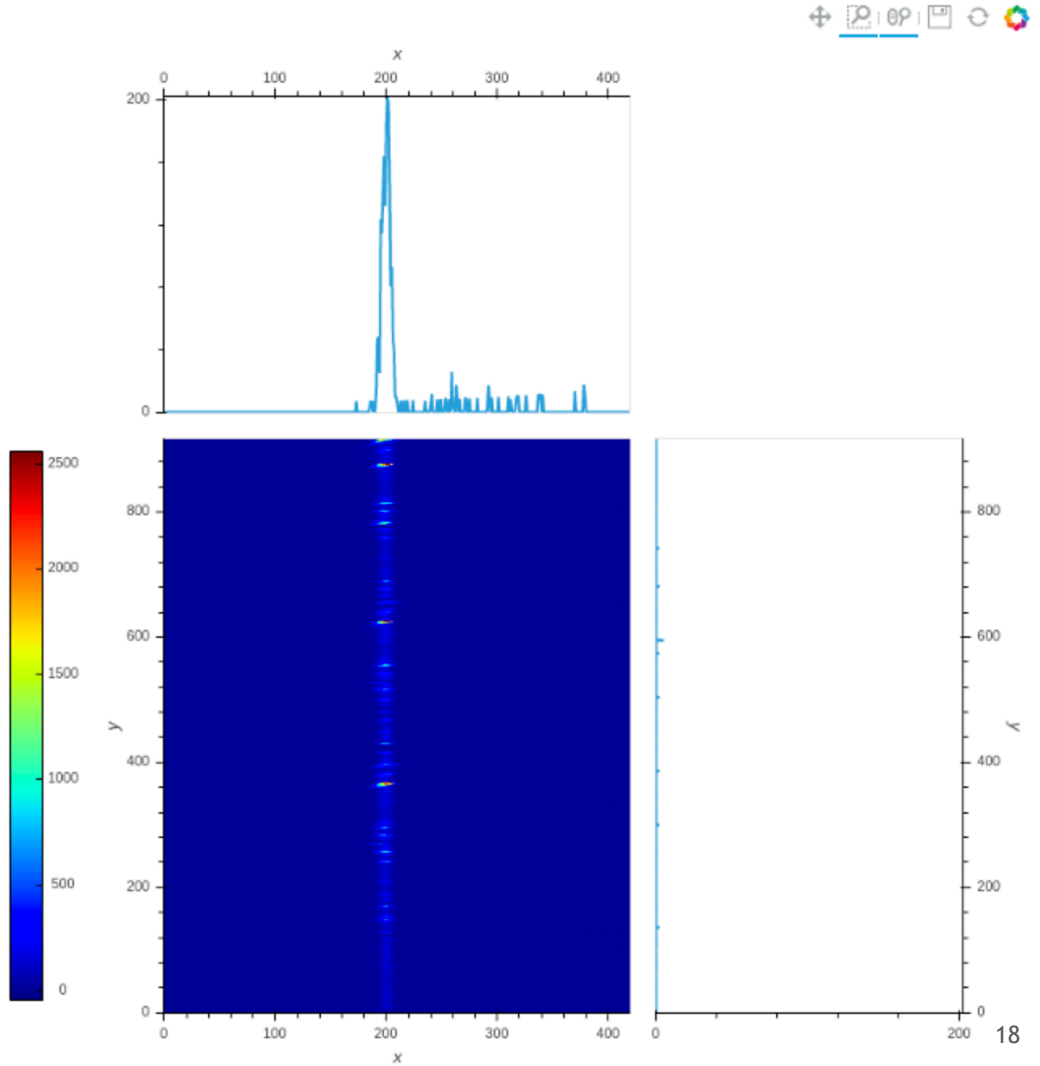
- Plot 1D spectrum
- Plot 2D Area



Holoviews multiplots

- Simple multiplot
- 2D Area + Mouse select for x, y 1 D integrations

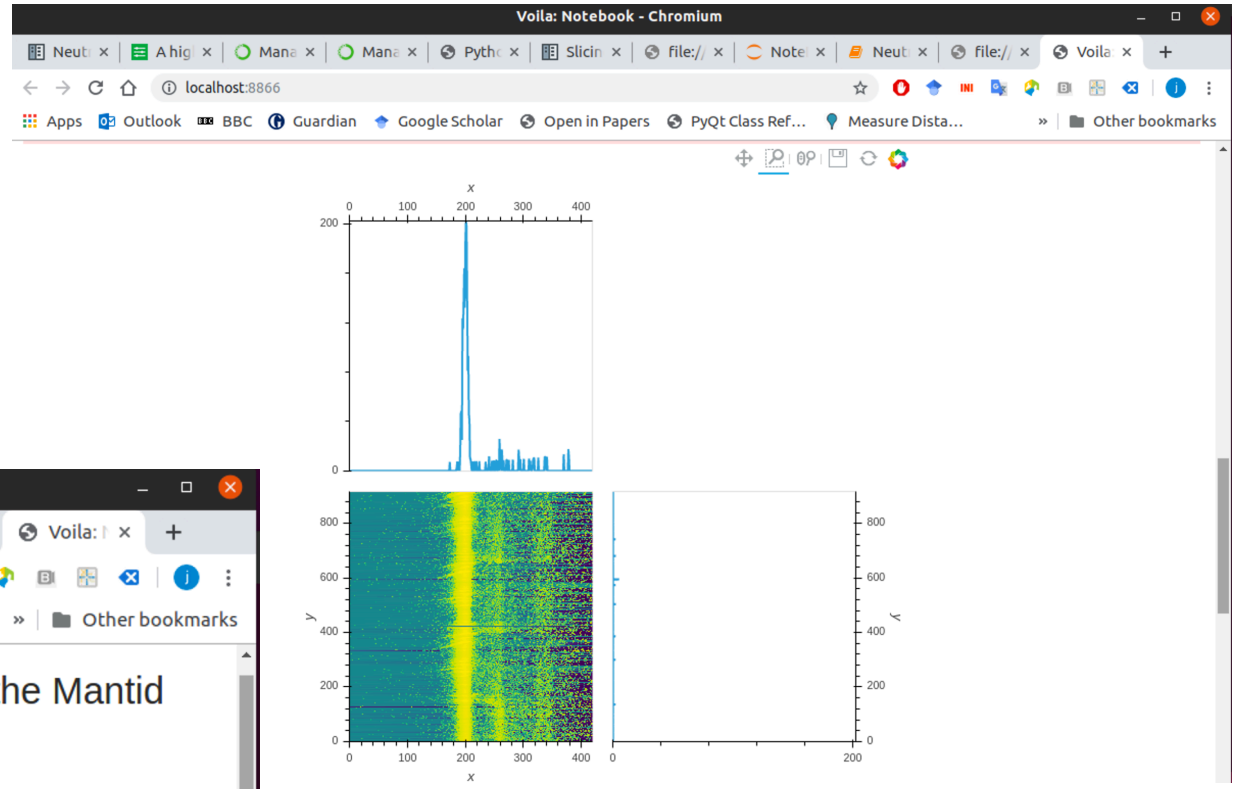


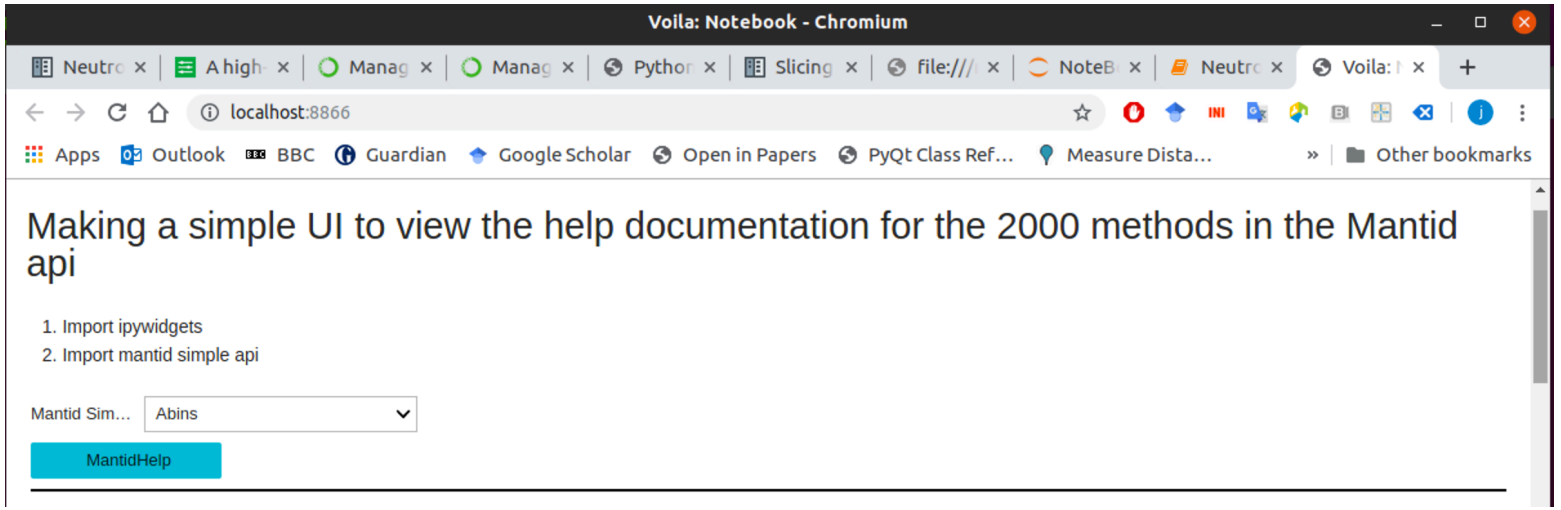


Tools deployed as interactive web apps

- Same notebook rendered as Web Page
- Deploys simple to use UIs

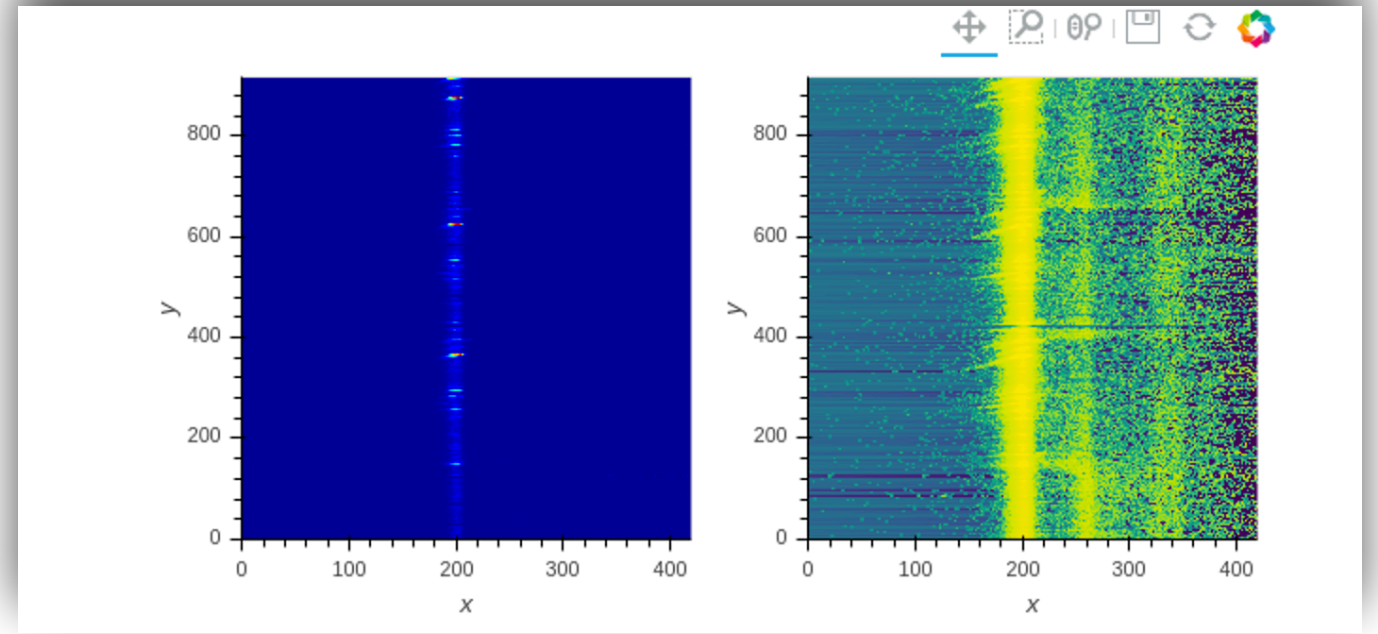
Voilà turns Jupyter notebooks into standalone web applications.





Data Visualisation

- So many really useful tools
- MPL
- Plotly
- HoloViews
- DataShader
- Users are not constrained to a single library

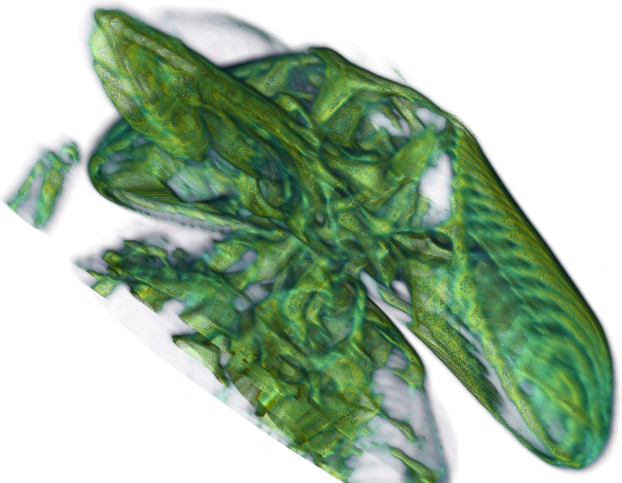


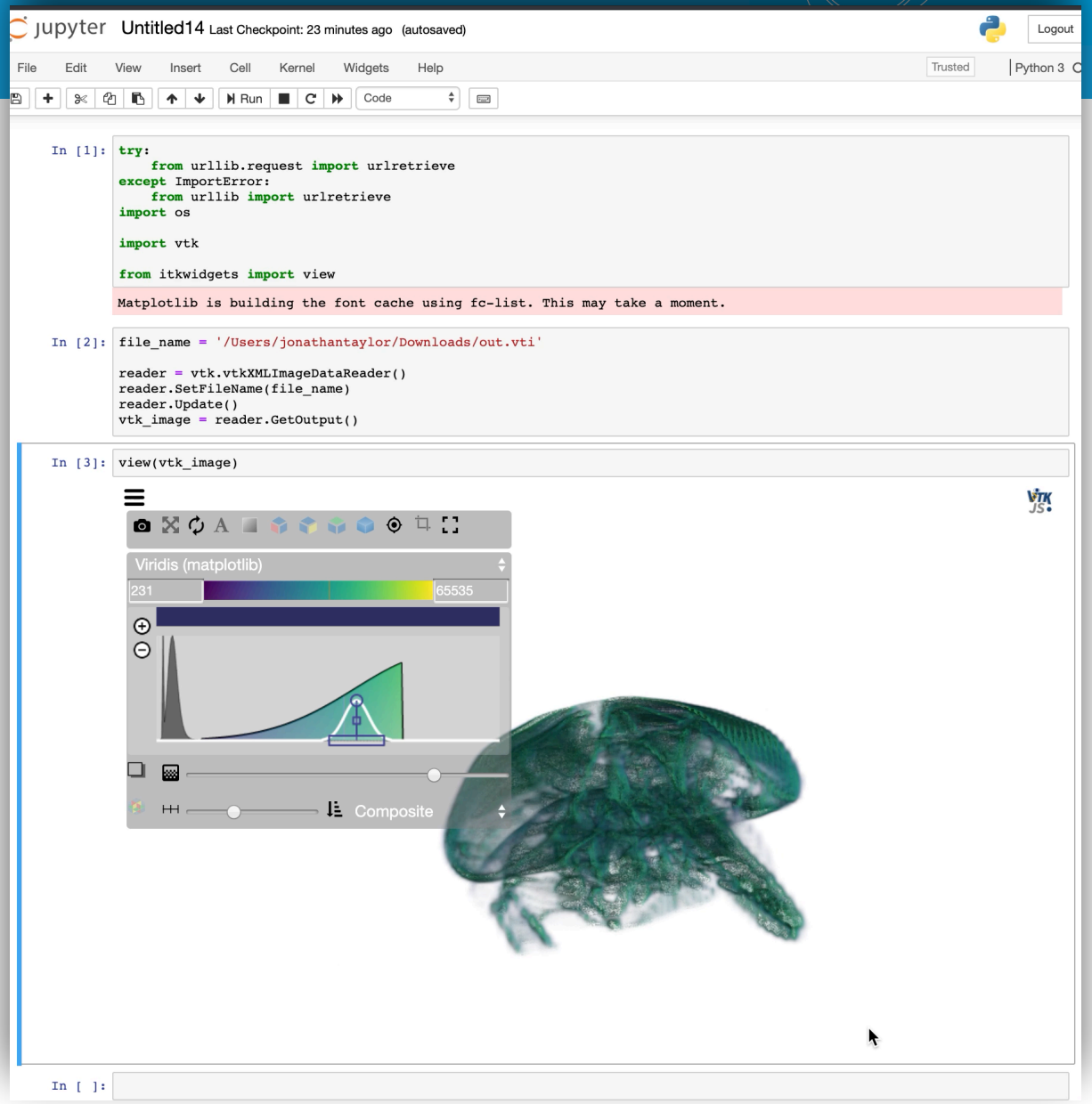
Inelastic Neutron scattering data visualised using Data Shader

Visualisation of Imaging data

- ITKWidgets & VTK

- Data From PSI imaging group
- Neutron CT of fruit fly
- Can be used for nD neutron data





Jupyter for data analysis - example

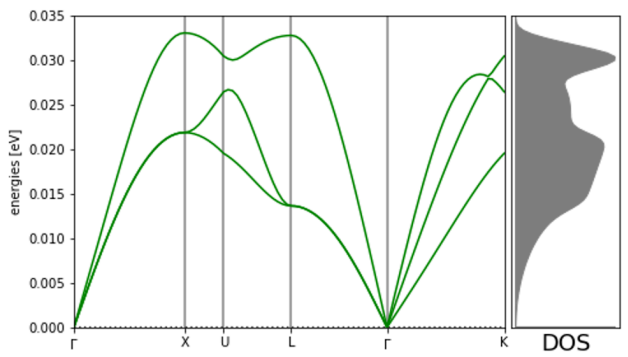
- Most analysis codes are exposed through Python
- Example :
  - Atomic simulation environment ASE
  - Simple calculation of Density of states and phonon dispersion relations for Aluminium

In [26]:

Slide Type Sub-Slide

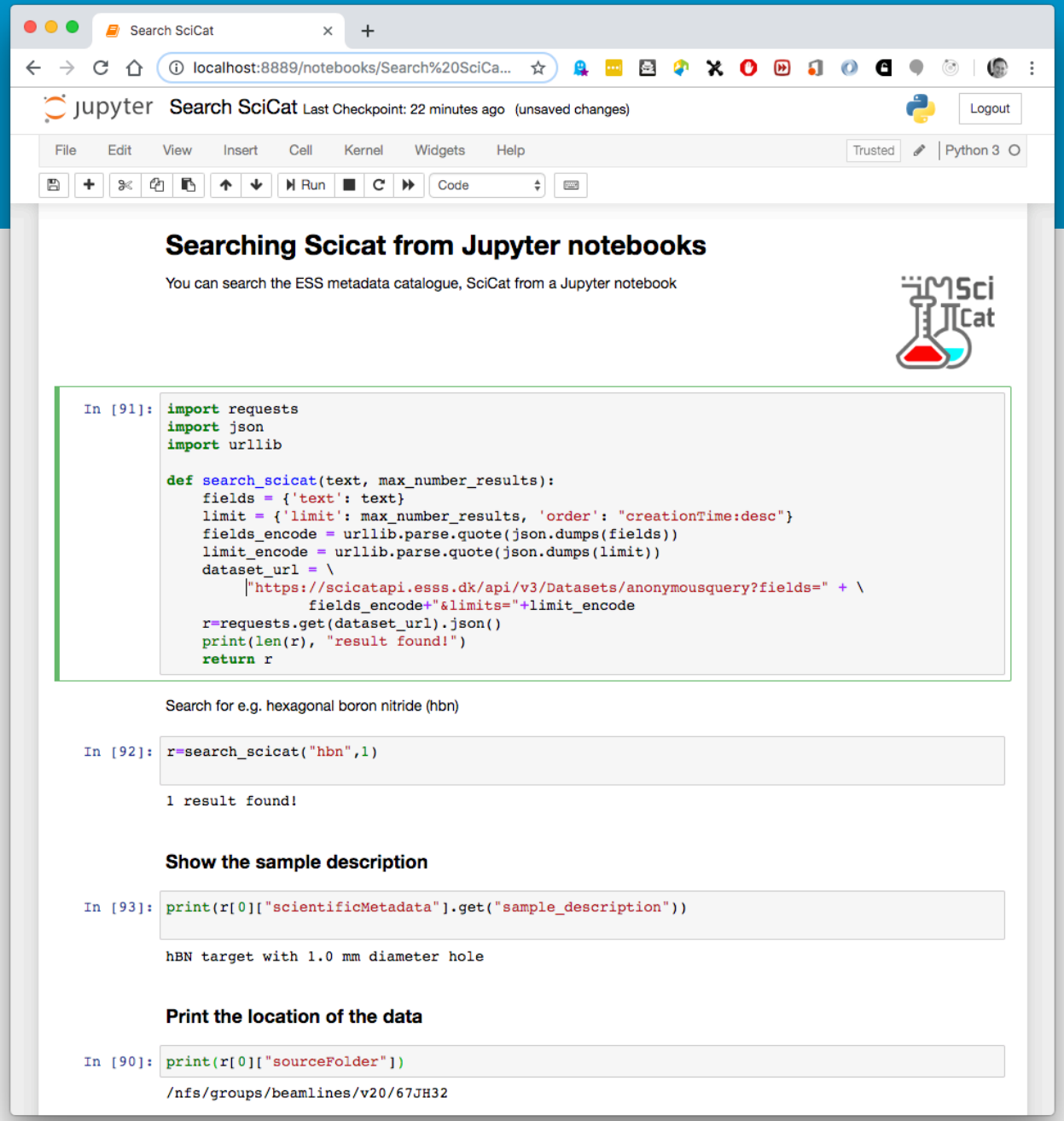
```
# Plot the band structure and DOS:
%matplotlib inline
import matplotlib.pyplot as plt


fig = plt.figure(1, figsize=(7, 4))
ax = fig.add_axes([.12, .07, .67, .85])

emax = 0.035
bs.plot(ax=ax, show=False, emin=0.0, emax=emax)
aa=bs.todict()
aa.keys
dosax = fig.add_axes([.8, .07, .17, .85])
dosax.fill_between(dos.weights[0], dos.energy, y2=0, color='grey',edgecolor='k', lw=1)

dosax.set_ylim(0, emax)
dosax.set_yticks([])
dosax.set_xticks([])
dosax.set_xlabel("DOS", fontsize=18)
```

Out[26]: Text(0.5, 0, 'DOS')



Jupyter & SciCat

- Access the facility data catalogue.
- SciCat can be accessed directly from your jupyter notebook!
- You can search for metadata about different experiments at ESS



https://git.io/fjSFG

Summary

EUROPEAN SPALLATION SOURCE

- Jupyter will be used at ESS for deployment of a broad range of scientific computing
- Widgets / JupyterLab (not mentioned) provide a flexible UX
- Development of UI is fast when based on well written python compatible frameworks