PaNOSC WP4 workshop European XFEL 25 to 27 June, Schenefeld, Germany

Useful links:

* This document: <https://docs.google.com/document/d/15tp-qCzhzwOA_BITVgnojZyE0bKoqWmzfZ1qgwC0XDQ/edit> or <https://tinyurl.com/euxfel-wp4>
* INDICO page: <https://indico.desy.de/indico/event/23379/>
* Slack: <https://panosc.slack.com> (request access from [Jamie.hall@ill.eu](mailto:Jamie.hall@ill.eu) or [Hans.fangohr@xfel.eu](mailto:Hans.fangohr@xfel.eu) for access)
* Social media: Twitter @panosc\_eu, #PaNOSC
* Present slides by
  + Opening Chrome or Firefox
  + Open <https://appear.in/fangohr>
  + Mute your microphone (and speakers)
  + Share your screen

Note: only 4 clients are allowed to connect (One is Jerome, one is the displaying machine, one is the speaker.).

# 25 June 2019

Minutes (please all contribute ideas, concerns, questions, possible designs, URLs, etc)

XTOB 1.108 has a kitchen & glasses

Emergency cooling (ice cream) available ~13.50

9:35 Andy Goetz

* WP4 is biggest work package
* 15% of project is over
* Invited to present use cases at a future (name?) workshop
  + Use cases not documented in the EC proposal
  + Some use cases at <https://github.com/panosc-eu/panosc/tree/master/Work%20Packages/WP4%20Data%20analysis%20services/resources>

10:00 Hans Fangohr

* STFC 1 PB space - how to access
* Amsterdam EGI meeting
* Keep track of where data is moved to - more convenient locations - can this be done in data catalog?
* Broad goals:
  + Identify use cases for each institute
  + Agree on the broad strokes for the infrastructure and solutions to use
  + Propose and fix some more fine-grained internal milestones
  + Ensure better co-operation between facilities
  + Form small inter-facility teams amongst ourselves

10:20 CERIC-ERIC Update - Carlos

* Marco from CERIC ERIC also here
* Use cases:
  + VUO + h5nuvola (web based hdf5 viewer <https://github.com/ElettraSciComp/h5nuvola> )
  + H5nuvola on binder:
    - Uses app mode (hiding cells, only shows widgets)
    - Does things that hdfview does
    - Demo: <http://tinyurl.com/y3422mg8>
  + VUO
    - Management of experiment lifecycle
    - Metada for experiment
    - Remote access (via web)
  + PyMCA (jupyter+remote desktop, Fitting x-ray powder diffraction (XRD), input from hd5, CLI and GUI) - <http://pymca.sourceforge.net/>
  + PtyPy (jupyter, GPU enabled, io is hdf5) - <https://ptycho.github.io/ptypy/>
* JupyterHub deployment under consideration
  + Have a single notebook server for TwinMic beamline (X-ray microscopy)
    - Kernels: Python2, Python 3, PtyPy + GPU
* Users largely rely on the notebooks and services for the last steps, just the visualisation, not full data analysis/reduction/creation

10:40 ELI Update - Jakub

* Recently hired extra person for PaNOSC
* Interviewing for new WP3 position
* Mariana, Eduard, Robert Racz
* Notebook that creates XDMF file that can be presented by Paraview
* Notebook to create files for visualisation, then use (hard-ware accelerated) visualisation software (Binary buffers direct to GPU for fast 3D visualisations)
* Notebook use cases:
  + Analytical modelling
  + Simulation analysis (runs notebook on HPC)
* First users in June 2019
* 200 to 400 TB estimated for 2019, multiple PB projected over next years

10:48 ESRF Update - Thomas

* Shut down for year -
* Old cluster for jupyter, DAAS
* JupyterHubs
  + Prototypes in use, one beamline does all data reduction and processing in Jupyter Hub
  + Not visible from outside ESRF; will be fixed when new hardware and Kubernetes arrives (people use ssh port-forwarding for now)
* Use cases:
  + Goals: based on open source, have sample data
  + 1. PyNX
    - Command line or Jupyter Notebook
    - Need tools to visualise the data (3d or 2d cut planes)
    - No 3d visualisation in notebook for now. [Hans: Try ipyvolume or k3d?]
  + 2. Cryspy
    - Simulation of spectoscropy (Marius Retegan)
    - Desktop application -> VISA
  + PyFAI
    - GUI for calibration, QT
    - Aim to move selection into notebook
  + Comment on Hardware: PyNX and PyFAI are meant to run on GPU. Problem, no GPU where the notebook is running, but notebook can spawn jobs on nodes with GPUs

ESS Update - Lottie

* DMSC data centre
  + Work on workflows
  + Scipp (pronouce: skippy)
  + Consider using DASK
  + Sasview/ sasmodels
  + McStas
    - Neutron ray training simulation package
    - Python generates meta file, translated to instrument, then run on cluster -> use Jupyter to create metalanguage file
  + Qens: binderHub example
* Status
  + Jupyterhub instance running in dev network
  + Kubernetes and LDAP
  + BinderHub installation ongoing, issues with proxies
* Staff: split between Kareem Galal , Lottie and New staff member starting 1 August.
* Comment: Use rancher for kubernetes

11:44 Report from EuXFEL (Robert)

* Notebook applications
  + Karabo-data-interactive
  + Geo-assembler -> link to KaraboFAI
  + Warppipes for complex jobs
  + Can track on SLURM if they don’t know how to do batch jobs
  + Run data calibration pipeline via notebook (Steffen’s talk)
* Tutorials for users (in notebook)
  + General introduction
  + Specific for a particular experiment
    - Mix of tutorial and template
* JupyterHUB
  + Dev version allows picking template notebook (such as Python-Tutorial)

11:54 Report from ILL (Jamie)

- JupyterHub

- in pilot phase

- BinderHub

- not convinced people will use it

- question: What barriers could be removed to make adoption more attractive? Proposal from the audience: Default container/requirements.txt; integration of Git(lab) with Jupyter (no need to learn about Git)

- 4 people across WP3 and WP4

- developing new RDP protocol, converting x server to WebGL + textures (not public yet)

- working group of scientists

12:02 Report from WP5 (Carsten Fortmann-Grote)

* Developing an API layer for the users. Access via Jupyter Notebooks, Desktop applications, or other workflow managers (Knime, …?)
* Defining interfaces between different simulation steps
  + [Should we use the same interfaces for data analysis? Ideally yes.]
* Connection of simulation work in WP5 to WP4
  + Produce data to feed into software
  + Produce data to benchmark analysis algorithms
  + Combined (iterative) data analysis schemes, using forward-reverse data analysis schemes
* Data standards
  + openPMD for physics
    - [www.openpmd.org](http://www.openpmd.org)
    - Standard format for simulations of particles and meshes
    - Does parallel data
  + NEXUS for detector data
* Model based on ASE - Atomic Simulation Environment
* Can we use simulation to provide data for analysis for our early prototypes (while real data cannot be moved)?

12:15 Lunch break

13:36 Report from WP3 (Gareth)

* *Seamless* data catalog integration with EOSC
* Link to use cases on slides
* Data movement
* Need some authentication/separate search to deal with embargoed data
* Consult with scientists to find out what metadata needs to be tracked
* PaNOSC API Draft - link on slides
* Question from WP3 for WP4: what metadata do you need to be searchable?
* Authentication is still being dealt with by WP6
* EOSC is the portal, WP3 just provides the API to EOSC for the search
* Catalogue search demonstrator portal is a deliverable for WP3
  + Include links to fire up appropriate analysis
  + Something to use as a prototype work between WP3 and WP4 at ESS?

13:57 Report from WP6 (Jamie)

* SAML 2.0, ABFAB (Moonshot)
* <https://wiki.moonshot.ja.net/>
* EOSC Authentication , using Umbrella ID currently, will move to EDUTeams (Géant)
* Ongoing evaluation for data transfer standard
* EGI notebooks
* Globus, rucio, FTS3 for data transfer
* OneData
* <https://fts.web.cern.ch/>
* <https://rucio.cern.ch/>
* <https://onedata.org/#/home>
* <https://www.egi.eu/services/notebooks/>

14:00 Report EGI (Giuseppe)

* Jupyter Notebooks ‘as a service’ in the EGI Cloud: <https://notebooks.egi.eu/hub/login>
* Builds on top of JupyterHub
* ONEDATA
* B2DROP
* Reproduce results:
  + Notebook, mounts data set through ONEDATA
* Can generate PID
* Install OneProvider
* Caching available, but not by default
* Location aware scheduling is not currently a feature. (is it on the roadmap?)
* Pre stage the data? Sure, ask for it ahead of time
* Make data available
* Webdav as FlexVolume in k8s
* Next steps:
  + Move to production
  + Deploy binder instance
  + Test AAI check in

14:15 Calipso Plus, VISA, towards a common Portal (Jamie, Aidan)

* Discussions
* [Udocker](https://github.com/indigo-dc/udocker) - run docker containers (no fancy networking, not k8s compatible) without root permissions (M Crusoe)
* ILL has screencasts explaining how data for an instrument is typically analysed
* Common API
* Discussion: avoid re-inventing the wheel
  + M. Crusoe invites you all to join the wheel building and maintenance communities as you discover them :-)

15:07 Break

15:30 Discussions:

[break out rooms available: XHQ 1.007 & XHQ E1.150 & XHQ E2.037]

17:00 Summary

17:30 Close

# Wednesday 26 June 2019

9:15 Welcome (Hans)

* Upload presentations to INDICO (<https://indico.desy.de/indico/event/23379/>)

9:15 Michael Crusoe: Common Workflow Language (CWL)

<https://tinyurl.com/PaNOSC-CWL>

* C for Commonly used and Commonly implemented
* Standard for describing
  + Command line tools
  + Workflows
* May not have shared filesystem
* Use on laptop and remote server
* There are bad standards out there!
* Discussion/examples:
  + Do not pass around paths, it’s not portable
  + Containers are supported but don’t have to be used
* Workflow approach vs notebook approach
* Provenance! Show where data comes from - verify results
* YAML since you can have comments, not in JSON!
* CWL model for tools
  + Turns POSIX command-line tools into functions
  + Typed input and output
* Hardware description available (#cores, memory, temporary and output storage estimations)
  + -> can use for PaNOSC to extend Binder specification to support hardware requirements?
  + GPUs requirements not included.
* Data locality with CWL
  + URI/IRI
* Community based standards development
* ResearchObject.org - helps with provenance
  + Intro
    - EOSC all about re-use of tools
    - Customise for the customers relevant
  + Just a json file …
  + Raw data is never fair
    - Where is identifier (?)
    - Where is provenance
    - Where is attribution?
  + Recommends to checkout openarchive (?)
* How much can you trust results from this process?
* Biocompute objects
* Are you just writing scripts or have standalone tools?
* Are you packaging and containerising the tools?
* Users rarely reuse workflows without making changes; different tools, parameters
* Where do manifests get persisted?
* It’s your choice
* <http://www.researchobject.org/ro2019/>
  + Tuesday 24 September 2019 (2019-09-24)
  + 2019-07-05 RO2019 submissions due: articles
  + 2019-07-15 RO2019 submissions due: abstracts for oral presentation
* Workshop on Research Objects (RO2019) At IEEE eScience 2019, San Diego, CA

10:00 Anton Barty: ExPaNDS Project

* 750 TB in 5 days
* |Don’t want to mess around
* Science is not in computing is in the results
* Don’t want to be told there is a new API and rewrite software
* Shouldn’t have to repeat experiment unless you want to
* Archive beamline scientists for every experiment for full reproducibility
* Kickoff meeting 11 and 12 September
* Electronic logbook
* Google sheets as electronic logbook
* Confluence as electronic logbook
* Need for better online logbooks
* User education
* Google spreadsheet is more compelling
* Snap a photo on your tablet and upload it
* ESRF is preparing electronic logbook

10:20 Michael Schuh (DESY IT)

* Function as a service here means container as a function
* Kafka, Apache Openwhisk to manage scaling
* If no-one uses this function it doesn’t block memory
* Chain the containers
* Upload a lot of files to dCache it automatically sends message to Kafka queue and open whisk produces a visualisation for each file
* “Git-ops”: put everything into git, you just push, everything else is built (and deployed?)
* Keeps everything stateless, so it is portable (not mounting file systems, but use webdav)

11:50 Robert Rosca - Review of computational portal (workflow tools?)

* CodeOcean
  + <https://codeocean.com/>
  + Jupyterlab
  + Costs money - but good template
* DAGsHUB
* DVC - Data Version Control
* Wholetale
* Hubzero.org
* …
* Michael Crusoe mentions: [galaxy platform](https://galaxyproject.org/) that combines it all, is open source, has been around for 10 years, and has an annual conference . Mostly bio, but very 'customizable for other communities.
  + Video demo of interactive environments <https://www.youtube.com/watch?v=OOrCUYj3Pok>

Next WP4 meeting

Should be every 4-6 months

Hackathons

Working meeting

ILL can host next meeting

ESS following meeting

Jamie has licence for confluence

Confluence.panosc.eu

Rooms available 26 June

* E1.041 (11:30 to 15:30)
* R2.042 (11:30 to 14:00)
* E1.150 (after 12:00)
* E2.148 (after 15:00)
* E2.037 (all day)
* XTOB1.01 (after 11:00)

13:30 Notebooks in use (Scientists from EuXFEL, Steffen Hauf)

* Offline calibration at XFEL
* Remove calibration detector peculiarities
* Corrected and calibrated data is the main product
* Produce correction parameters
* Manage correction parameters
* Optimise corrections
* Python - fast development cycles - good data analysis
* pyCalibrate
* Calibration web service
* Configuration via yaml files
* Defined in notebook - automatically propagate
* CLI
* Produce pdf reports
* Expect Usage
* Monitoring
* Anything in notebook will be rendered into pdf
* AGIPD LPD DSSC
* Complex reports, over 100 pages
* 24 production notebooks
* 30k SLURM jobs
* 15 minutes
* Heatmap plots don\t convert well from SVG to pdf/png
* Tabular output doesn’t render well in pdf
* RST and HTML same
* 1.5 years development
* Use case: “I see something on detector I don’t understand”
* “Interact with notebook to determine cause”
* nbparametrise()
* Why use notebook over normal python?
  + Good for ongoing development
  + Allows data-experts to contribute even if they are not confident programmers
  + From interactively developed notebook to a scriptable notebook is often just some tidying up
* <https://github.com/jupyter/nbdime>
* Nbdiff notebooks; no known gitlab integration

13:45 Notebooks in use (Scientists from EuXFEL, MID, Mario)

* Software deployment
  + Install into proposal folder a conda distribution (tailored for the experiment)
  + Also provide same examples (in notebooks) how to read and display and analysise data
  + Also provide sample data
* Why using Jupyter Notebooks?
  + 1. Persistent and interactive notebooks
    - Better than Matlab: figure not inline in Matlab
  + 2. Code, Figures and Markdown
    - Headings
    - Explanations
  + 3. Extensions
    - Table of contents
    - Heading / code folding
    - Snippets, …
  + 4. Running in a browser and easy to use (with Jupyter Hub)
* Use cases at MID
  + Tutorial, with table of contents, with folding of (numbered) headings
    - Initialisation cells to create required variables and functions
  + As an e-log
    - [Long notebook]
  + Tried online visualisation in the last experiment (update from data stream automatically)
* Issues
  + Performance issues with large notebooks
    - Slow to scroll, table of contents helps
  + Notebooks can only be used by one user at a time
  + Connection issues and output issues
  + Interactive figures can be slow and provide only basic functionality
    - Cannot click on line to change colour
    - Cannot click on data in the figure
  + Notebooks are lacking file system control and browsing features
* Questions
  + 500 MB notebook limit for jupyterhub

14:41 Notebooks in use (Scientists from EuXFEL, SCS, Laurent)

* Main advantage: flexibility to diagnose problems in realtime, based on existing (non-trivial) analysis notebooks
* Snippet
* Import snippets
* Tedious to configure snippets
* Have to add in json file

15:45 Discussion: -> Break out groups Wednesday

19:00 Workshop dinner

# 

# Thursday

9:00 Breakfast coffee

9:30 Visualisation tools and hdf5 (Thomas Vincent)

* Nexus
  + ESRF uses NeXus through the entire data file - raw data, calibrated, processed, all conform to the NeXus standard
  + nexusformat.org
  + Use at Sardana as well as ESRF
  + NXxas as an application definition, some flawed design choices with instrument naming conventions - <http://download.nexusformat.org/doc/html/classes/applications/NXxas.html>
  + Application definitions should come from developers or communities otherwise useless ...
  + NeXus also stores the program which was used and its version, and it can store the workflow and configuration so that you can use the NeXus metadata to reproduce the processing used to generate the data
  + Can use NxNote for config storage
* ESRF uses ICAT for their metadata catalogue
  + To map NEXUS application type
  + to categories in ICAT
* <http://www.esrf.eu/computing/bliss/doc/bliss/> - BeamLine Instrumentation Support Software
  + REDIS + memcached
  + Storing online data until memory is full?
  + Provenance done via NXprocess
* SILX view application understands the NeXus format
* Capable of generating some plots based on the type of the data and the metadata
* Attributes (metadata) and the name of the dataset/group is used to specify what the data is, can cause problems with naming conflicts/changes in the future
* Requirements:
  + Faster pan/zoom interaction
  + Users changing colour maps, normalization, ranges, etc…
  + Masking
  + ROI and profile selection
  + Better interactive plotting for large images (30k x 30k, 12G pixels)
  + Better 3D plotting

10:00 Virtual hdf5 data sets, h5glance and h5py (Thomas Kluyver)

* H5glance
* Started with h5ls -rv - wall of text
* H5view slow with x-forwarding
* H5glance
* Nicer way to look at files
* Pasteable link into jupyter notebook
* Low level h5py close to C API
* Virtual datasets
* Composed of pointers to other datasets
* Exciting!
* Use case - big detectors - each writes to separate file
* Sequence files - cuts off sequence and starts a new one!
* 64 files in a run from one detector
* Can be exposed as one massive array
* Cxi 12 MB pointers to 190,000 images
* Convenient interface
* Q: Does h5copy copy actual data from virtual datasets?
* Q: EuXFEL specific: when data for a module for a train is missing, how does the virtual data set (or the tool that creates the virtual data set) deal with this?

<https://indico.esrf.fr/indico/event/33/overview>

10:30 Deployment of computational environments (Robert Rosca)

* Robert describes script-based creation of computational environments
* <https://spack.io/>
* Manages versions of e.g. hdf5 and dependencies versions
* Flatpack?
  + ESRF:
    - use debian packages, provide nightly builds
    - Python virtual environments
    - Use miniconda to install applications
    - Using ‘module’ as well
  + Thomas mentions: Binderhub as another option
  + Michael C mentions: GUIX
    - Some facilities/communities make Debian packages installed natively or access via software containers
  + <https://www.softwareheritage.org/>
  + ESS: for C++ use Conan <https://conan.io/>, for node npm, and python pip
  + ESS config with Ansible <https://www.ansible.com/> and Puppet <https://puppet.com/>, Docker for integration testing
  + Mirrors (archives):
    - Software heritage
  + DESY: on the cloud
    - Puppet modules
  + Soleil: using Maven

11:30 JupyterHub & BinderHub experience (Frank Schluenzen)

* HPC
* Choose node on scheduler (SLURM, <https://slurm.schedmd.com/documentation.html>)
* Choose GPU
* Choose number of nodes
* Pick notebook/jupyter lab/binder repository
* Spawn
* Build conda environment in memory and setup notebook, if only one notebook launch automatically else see multiple notebooks in file manager
* /dev/shm is where it is actually taking place (not persistent, modify but won’t change, need to export)
* $ scontrol show job $SLURM\_ID
* Can see details of job and number of cores
* If i want a different repository - always go back to same page
* 16 nodes, infiniband, FhGFS ([https://www.beegfs.io](https://www.beegfs.io/))
* 25k cores
* 200+ GPU
* EOSC cloud
* JupyterHub on Maxwell
  + 3 AMB nodes for jupyter jobs, 32(x2) cores per node, 256GB RAM, allow 32 concurrent jobs per node
  + ~180 unique users in May 2019, ~15% of total number of users
  + Comparable with Matlab usage now
* Performance issues
  + Notebooks don’t need much CPU time
  + But lots of RAM
  + Dev version available in <https://max-jhub003.desy.de>
* Packer (<https://packer.io/>) for Openstack
* Loadbalancer as a service
* Hashcorp Terraform (<https://www.terraform.io/>)
* Kubespray (terraform + ansible)
* Cinder storage class

13:30 Discussion and planning

* Rooms available:
  + E2.037: Robert: Use case discussion and gathering. Desired outcome:
    - Create list of use cases (notebook and other) [in this file or on github/../wp4]
    - Some useful items of information/attributes:
      * Title of use case
      * Which facility
      * Where is the analysis script?
      * Data small(ish)?
      * Data public?
      * Scientist who cares about this (and can provide expert advice)
      * Hardware requirements and approximate execution time
  + To do:
    - Change structure on git page
      * Folder for scripts, (small) data(?), notebooks
      * Screenshots of desktop applications
      * List of scientists
      * Special requirements like gpus, matlab, etc…
      * Share tutorials or templates given to users
      * Look at facility documentation to gather uses?
    - **Use the shared data hosting for the example data STFC**
    - Facility use cases:
      * CERIC-ERIC
        + VUO, h5nuvola, PtyPy, PyMCA tutorial
        + H5nuvola demo: <http://tinyurl.com/y3422mg8>
        + Request facility data (to be added)
        + <http://www.silx.org/doc/PyMca/dev/tutorials.html>
      * ELI
        + TBA
        + Analytical modelling
        + Simulation analysis (runs notebook on HPC)
        + See Jakub presentation
      * ESRF
        + [Already on git page](https://github.com/panosc-eu/panosc/blob/master/Work%20Packages/WP4%20Data%20analysis%20services/use_cases/ESRF-use-cases.rst)
        + <http://ftp.esrf.fr/pub/scisoft/PyNX/example_notebooks/CDI-Siemens%20star.html>
        + Needs GPU!
        + pyFAI best with GPU
        + Simulations
        + <https://gitlab.esrf.fr/paleo/sidi/blob/master/examples/distributed_fbp.ipynb>
        + Notebook submits to cluster, need OAR/SLURM
      * ESS
        + Mantid/scipp - python script + desktop app
        + SasView is desktop app
        + https://www.sasview.org/downloads/OldTutorial.pdf
        + QENS - notebook
        + <https://github.com/QENSlibrary/QENSmodels/tree/master/examples-binder>
      * ILL
        + VISA (TBA)

* + - * XFEL - see presentations
        + <https://in.xfel.eu/readthedocs/>
        + <https://in.xfel.eu/readthedocs/docs/data-analysis-user-documentation/en/latest/>
    - <https://github.com/panosc-eu/panosc/tree/master/Work%20Packages/WP4%20Data%20analysis%20services/use_cases>
  + XHQ: E2.062 Hans/Jamie Overall design: Amsterdam plan / BinderHub / role of portal plan, milestones
  + XHQ: E2.148?
* Improving Jupyter ecosystem [Robert Rosca]
  + Reproducibility, plotting, dashboards, interactivity

15:00 Summary

15:30 Coffee and close

16:00-17:00 XFEL tour

# Break out groups

# Tuesday

* XHQ E1.150: Hans/Jamie Overall design: Amsterdam plan / BinderHub / role of portal plan, milestones
* XHQ E1.007: Aidan JupyterHub Deployment - technical aspects
  + Different spawners - batch, kube, sudo
  + Binderhub, repo2docker, zero2jupyterhub
  + <https://github.com/jupyterhub/batchspawner/issues/138#issuecomment-501790720>
  + <https://docs.google.com/document/d/1_oeCBMvo6zW5DMeFHCmjQy-Eq2bQO9Q1DekT4laY27A>
* XHQ E2.037: Thomas Vincent Visualisation (and Hdf5 visualisation) -> can we converge / re-use?

# Wednesday

Break out groups at break + 30 minutes (~15:30)

* XHQ E1.150: Hans/Jamie Overall design: Amsterdam plan / BinderHub / role of portal plan, milestones
* XHQ E2.148: Detailed Q&A and hands-on session: Notebooks on HPC clusters? [Michael Schuh, Johannes Reppin]
* XHQ E2.037: Detailed Q&A and hands-on session: Workflows - CWL/tools/dependencies/…? [Michael Crusoe]
  + <https://www.commonwl.org/user_guide/>
  + <https://toil.readthedocs.io/en/latest/running/hpcEnvironments.html?highlight=slurm>
  + <https://github.com/common-workflow-language/cwltool#leveraging-softwarerequirements-beta> How to use “module load” or conda instead of Docker containers
  + Robert Rosca got toil-cwl-runner to submit SLURM jobs to Desy's Maxwell cluster.
  + virtualenv -p python3 env3
  + . env3/bin/activate
  + pip install toil[cwl]
  + TMPDIR=$HOME TOIL\_SLURM\_ARGS='--partition=exfel' toil-cwl-runner <https://github.com/common-workflow-language/user_guide/raw/gh-pages/_includes/cwl/02-1st-example/1st-tool.cwl> <https://github.com/common-workflow-language/user_guide/raw/gh-pages/_includes/cwl/02-1st-example/echo-job.yml>
  + M. Crusoe proposes a project to expose a Jupyter Notebook to a CWL runner seamlessly
* ?

Further ideas for break out groups:

* Use case discussions and finalisation
* Improving Jupyter ecosystem [Robert Rosca]
  + Reproducibility, plotting, dashboards, interactivity

Ideas(?) for tasks

* Put our software into pandata software catalog

# ––––––––––––––––––––––––––––

# Notes from break out groups

## Design discussion:

* Jacub Grosz, Michael Crusoe, Stuart, Giuseppe, Jamie, Gareth, Hans
* Sandstorm.io
* Other terms for portal: science gateway, virtual research environment :
  + Conference in November Science September 23 to 25 in San Diego, sciencegateways.org
  + Openscience foundation - combine data sources (osf.io)
* **Portal**
  + Only one (not WP3 and WP4 separately)
  + Select data set from proposal
    - Ideally based on experiment type (work out from NEXUS experiment type)
    - Or offer all analysis tools used at that facility, ordered by instrument (hierarchical)
  + Have suggestion of data analysis environment / templates
  + Fire up analysis environments
    - Either as independent browser tabs, or integrated into portal tab
  + Nice to have:
    - Display links to active analysis sessions
    - Chat to allow users from the same experiment to communicate
  + Jakub might produce some mock-ups
  + Milestones: ?
* **Discussion**
  + Might be a delay in accessing data if data has been archived already
    - Should communicate this to users: displayed message, sent email
  + Use microservices to avoid building monolithic application
* To be written up
  + Amsterdam model, discussed with notebooks
  + Milestone 1:
    - analysis in facility specific notebook (it is okay if this only works at the originating facility)
    - starting from some (facility-specific) data set
    - small and public would be useful
    - doing some analysis operation (python preferred) on this to
    - create a figure/table/number of interest to scientists
    - integrate scientists - we will need to use this analysis to attract them later to test and use the services
  + Milestone 2:
    - Create BinderHub instances at each facility
    - Translate the facility specific Jupyter Notebook (from Milestone 1) to execute in the BinderHub environment. This requires specification of compute environment via Binder style environment specification
    - put this specification and analysis notebook into some repository (github or local facility repo or else…)
    - Point BinderHub to that repository
    - The facility specific notebook should now execute via BinderHub at the facility
    - the computer environment is in a container
    - This software environment is facility-independent!
    - However, data access will have to use some facility specific method
  + Milestone 3:
    - find a facility independent way to describe the data source
      * analog to Binder software requirements, these are data requirements
      * it is unclear how to do that
      * if data sets are typically kept only by the facility they originate from, maybe it is okay if the data set locator specification is facility specific (such as EuXFEL/v1/p2222/r0013)?
    - Find a way to bring that data to the compute location (or vice versa):
      * Start with small example data sets (can wget them initially?)
      * Could we move (public) example data sets to the 1PB STFC (?) storage space we have for demonstrators?
      * At some point, use file transfer capabilities created in pilot in WP6.3 to access remote data
      * From discussion at meeting: ONEDATA seems a good vehicle?
        + EGI Notebooks support it
    - At this point, we should have (some) use cases that can execute on any PaNOSC facility and/or the EOSCHub, fetching data on demand
* **More discussion (tomorrow)**
* Resource management / API
* DAta analysis needing lots of RAM / hardware requirements
* Amsterdam model with VISA?

**MILESTONE: User interface design (=Portal), 3 months, Jakub + Gareth + Stuart + Jamie**

* What does the portal look like?
* Possible workflows
* Allow to start analysis part of workflow without search metadata catalog
* Offer simulation tools from WP5 as ‘virtual data sets’ that can be created by running the simulation

List of milestones:

MILESTONE: Oct 2019 Sample jupyter notebooks (Thomas+Robert)

MILESTONE: Nov 2019 User interface design (portal), (Jakub + Gareth + Stuart + Jamie)

MILESTONE: Feb 2020 Architecture selection of required technologies and specification (Stuart+Eduard+Aidan)

MILESTONE: Mar 2020 Compute resources API specification (Stuart+Jamie)

MILESTONE: Jun 2020 Sample Remote desktop with local data (Stuart + Jamie, VISA)

MILESTONE: Jun 2020 Sample Jupyter Notebooks execute in BinderEnvironment with local data (Robert+Thomas)

MILESTONE: Oct 2020 Prototype implementation for portal (Carlos + Eduard + Jamie)

* Must seek feedback from users

MILESTONE: February 2021 Sample Jupyter Notebooks execute in BinderEnvironment with remote data (Thomas+Robert+Jamie)

MILESTONE: Mar 2021 Launch Remote Desktop and Jupyter analysis from Portal (Eduard)

* Need metadata catalog to be searchable at this point -> align with WP3
* WP3 demonstrator: scheduled for March 2021

MILESTONE: Nov 2021 Demonstrator prototype completed (Eduard + Robert + Jamie + Thomas)

MILESTONE: Nov 2021 First iteration of Documentation/Training materials (Kareem + Thomas) [Python training in Jupyter, particular analysis for selected analysis types, …]

MILESTONE: May 2021, Obtained user feedback after beta testing / validation (Thomas)

MILESTONE: Nov 2022, Deliverable 4.4

Check: cross check and link with tasks/deliverables in WP3, WP6, WP5, WP8

WP4 Deliverables (from page 65 of proposal)

* Deliverable 4.1. Report on the current technical elements of data analysis at each partner site (M12 -> Nov 2019, R, PU, ILL)
* Deliverable 4.2. Prototype remote desktop and Jupyter service (M18 -> May 2020, DEM, CO, ILL)
* Deliverable 4.3. Remote desktop and Jupyter analysis service deployed at EOSC (M42 -> May 2022, DEM, CO, XFEL.EU)
* Deliverable 4.4. Publicly accessible demonstrator (M48 -> Nov 2022, DEM, PU, CERIC-ERIC)

Future ideas/features:

* Chat/sharing/collaboration microservice (?)
* Quota system microservice (to be discussed)

**Architecture**

* What microservices
* Deploy microservices through containers?
* How do we integrate binderhub/Jupyterhub
* Remote desktop websocket) microservice
* Kafka message queue

**Required APIs**

* Authentication (WP6, Task 6.4) - to obtain token; Jamie to work with WP6
* Catalog (WP3) - to retrieve embargoed proposals needs token / to search open data
* MILESTONE Compute resources API specification, complete before annual meeting, Stuart + Jamie

- to request compute resources (potentially at other sites / in the cloud), to show current use,

**Experiment classification**

* Scientific experiment type ontology to suggest appropriate analysis steps
* Develop and use this for a selected set of data (showcases)
* Allow users to choose any data analysis environment, but aim to suggest likely sensible ones where possible. Could base suggestions on beamline, or on statistics of past usage.

**Required outcomes from meeting**

* Internal milestones:
  + What do we want to achieve
  + By when
  + Who is in charge (not to do the work exclusively themselves, but to ensure it is done)

## Visualisation break out group

* Unstructured discussions
* Generic h5 viewer is needed
  + Multiple concurrent activities underway
* Could work with Jupyter Lab

## JupyterHub Deployment

* Jupyter Hub deployment
* CalipsoPlus deployment tutorial
* Creating custom notebooks
  + Created common notebook images
  + Accessing data from the notebooks
    - NFS, ?
  + Permissions
* Would be good to work together on visualisation capabilities
* Trello board?
* Function as a service, Kafka
* Collaborate on docker container recipes
* Ancible and jupyter jub

–––––––––––––––––––––––––––

### WP4 site leads:

* Thomas Vincent (ESRF), Jamie Hall (ILL), Robert Rosca (EuXFEL), Kareem [Lottie] (ESS), Carlos (CERIC-ERIC), Jakub (ELI), Giuseppe (EGI)