UKRI
Science and Technology Facilities Council
Natural Environment Research Council

JASMIN

JASMIN: Infrastructure to support a diverse range of scientific use cases

ISENES3 Virtual workshop on requirements for a fast and scalable evaluation workflow, May 2021

Ag Stephens, Philip Kershaw, Matt Pritchard and the STFC-CEDA and STFC-SCD Teams

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

Scientific Computing
Science & Technology Facilities Council

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

RAL Space

Overview

- CEDA and JASMIN
- Scientific use cases
- Topics:
    - Storage
    - Compute
    - Data transfer/migration
    - Cloud
- Bringing it all together

Science and Technology Facilities Council

Natural Environment Research Council

CEDA: Overview

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

- Centre for Environmental Data Analysis
- Mission: to provide data and information services for environmental science
- ~30 staff
- CEDA:
  - >20Pb of environmental data
  - Catalogue and data access services
  - ~67,000 users (~20k active users)
- JASMIN

UKRI
Science and Technology Facilities Council
Natural Environment Research Council

https://www.ceda.ac.uk/

JASMIN: Purpose

Supports data analysis for (NERC) environmental science community

- Large scale, data-intensive science

Designed for performance

- Tailored to needs of academic community

Compute co-located with the data

- CEDA Archive data (curated)
- Group Workspaces (self-managed, not curated)

Flexible compute capabilities

- Interactive and batch compute
- JASMIN Cloud provides autonomy, scalability

Science and Technology Facilities Council

Natural Environment Research Council

JASMIN: numbers

- **>40 Petabytes** high performance storage
- **>13,000** computing **cores**
- **High-performance network** design
- **~50 Private cloud 'tenants'**, to enable flexible usage
- Dedicated **high memory** and **data transfer** machines
- GPU cluster to be installed

User needs for "big-data" platform

- Access to any software packages...*that we might need*
- A stable unchanging software environment...*for the duration of my project (and when we come back and re-run later)*
- Access to unlimited processing capability...*at the exact time we are ready to run*
- Access to unlimited storage...*in case we need it*
- Tools to manage all kinds of workflows *across such a platform*

Science and
Technology
Facilities Council

Natural
Environment
Research Council

JASMIN: Scientific use cases

1. Interactive login (small)
2. Notebook (small)
3. Notebook on CaaS (medium)
4. Batch via login (medium/large)
5. Batch and multi-step workflows (large)
6. External Cloud interactions (including Object Store) (medium/large)

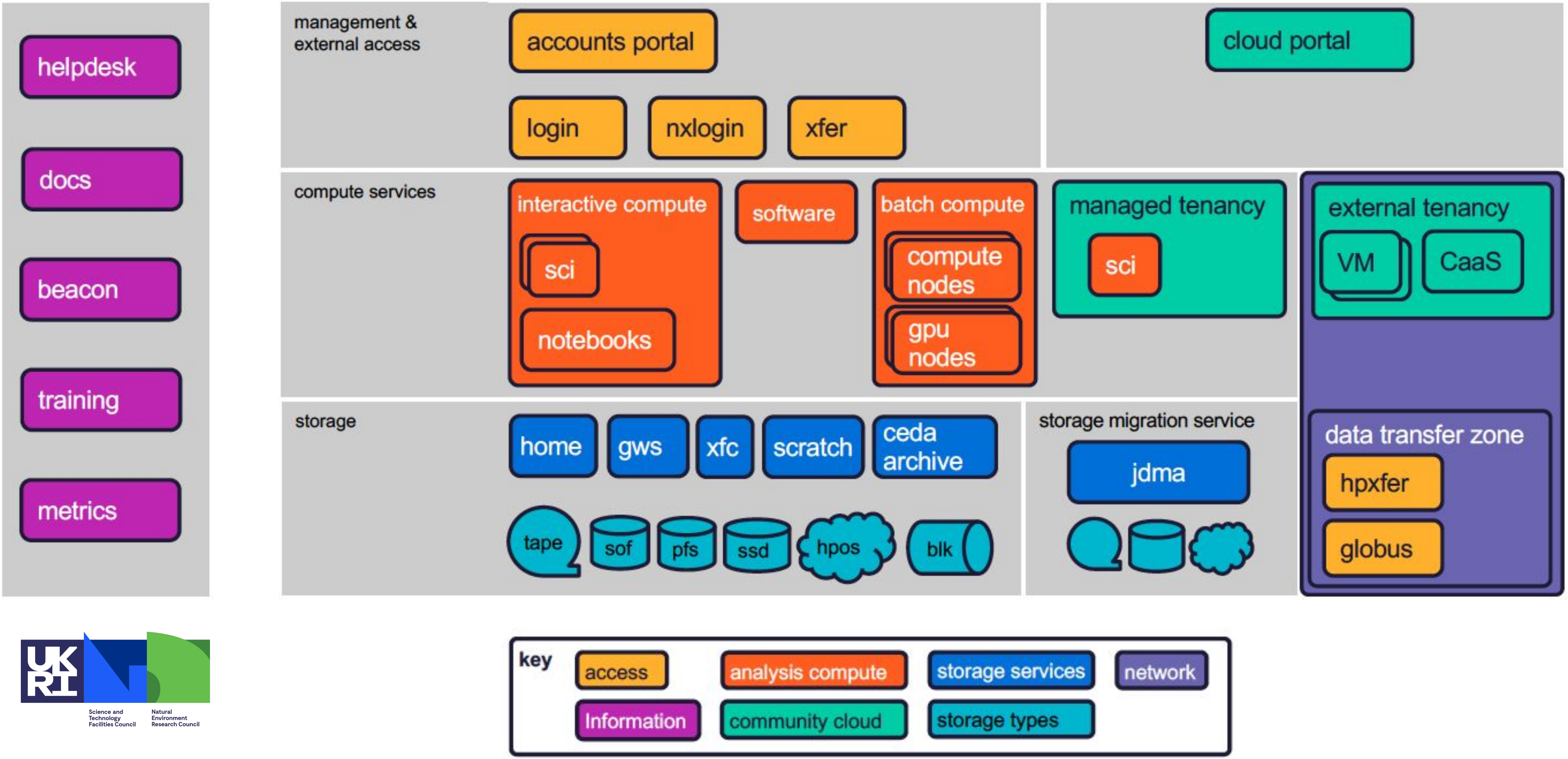
Scientific Computing
Science & Technology Facilities Council

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

UKRI
Science and Technology Facilities Council
Natural Environment Research Council

RAL Space

JASMIN Services



helpdesk

docs

beacon

training

metrics

management & external access

accounts portal

login

nxlogin

xfer

cloud portal

compute services

interactive compute

sci

notebooks

software

batch compute

compute nodes

gpu nodes

managed tenancy

sci

external tenancy

VM

CaaS

storage

home

gws

xfc

scratch

ceda archive

tape

sof

pfs

ssd

hpos

blk

storage migration service

jdma

data transfer zone

hpxfer

globus

key

access

analysis compute

storage services

network

Information

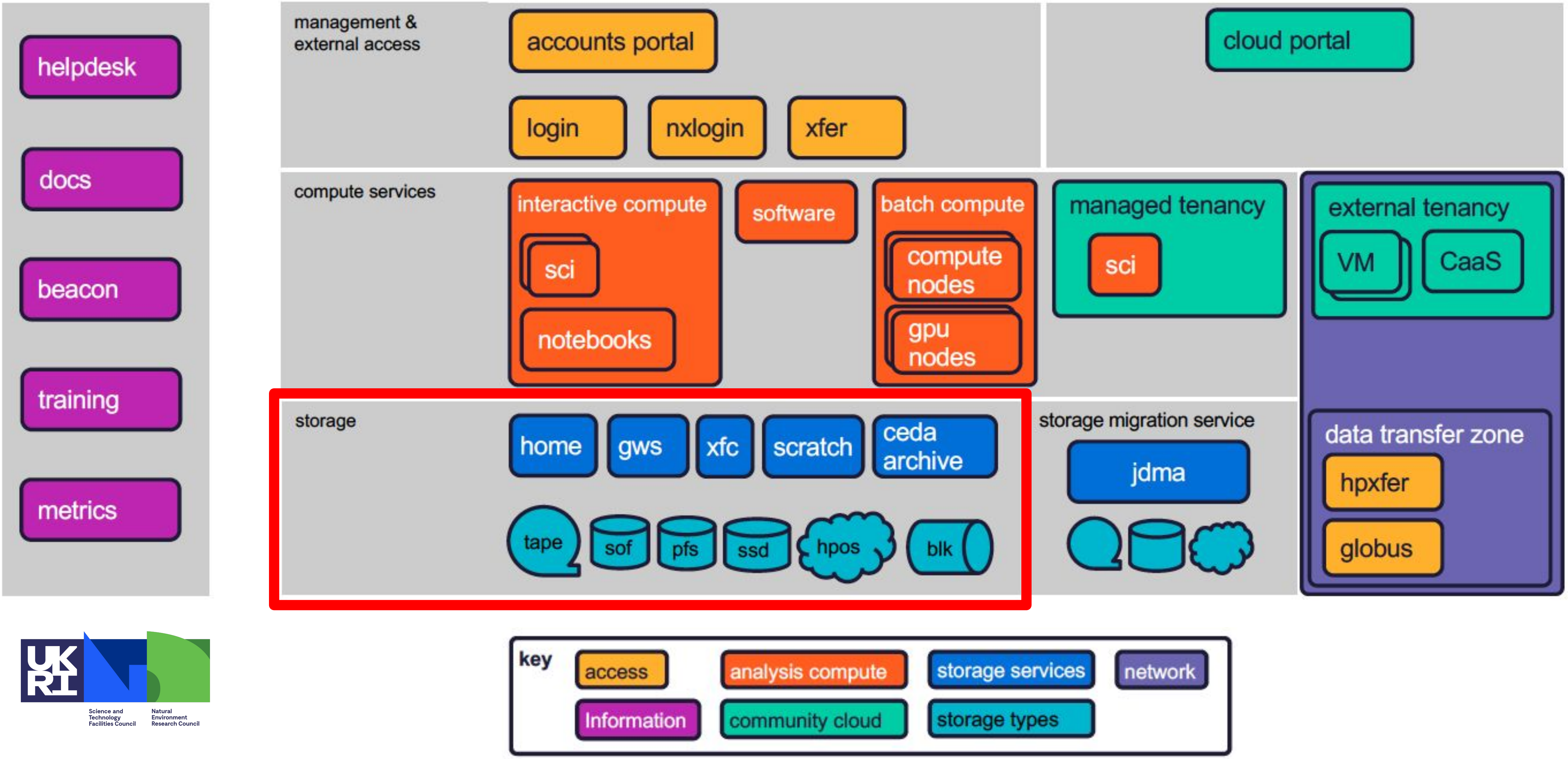
community cloud

storage types

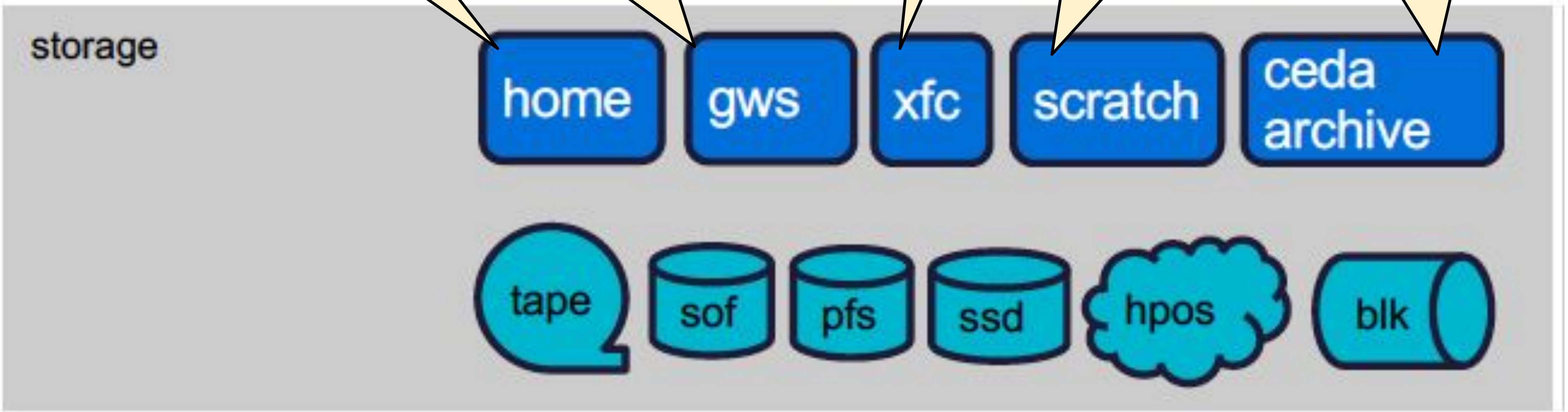
Science and Technology Facilities Council

Natural Environment Research Council

JASMIN Services



JASMIN Storage



SSD 100GB/user

Project disk, up to 250Tb, read/write

Transfer cache: temporary

100s of Tbs, temporary

Read access to CEDA data

Issues related to storage

- File-systems mounted across entire platform
- Heterogeneous file-system storage:
  - SSD, SOF, PFS - different properties/strengths/weaknesses
- Limitations on parallel writes
- Small-file (<64Kb) support: e.g. software environments
- Optimising use of storage media for large workflows:
  - Very significant impacts on efficiency
- When to use tape and object store?

Science and Technology Facilities Council

Natural Environment Research Council

The case for object storage

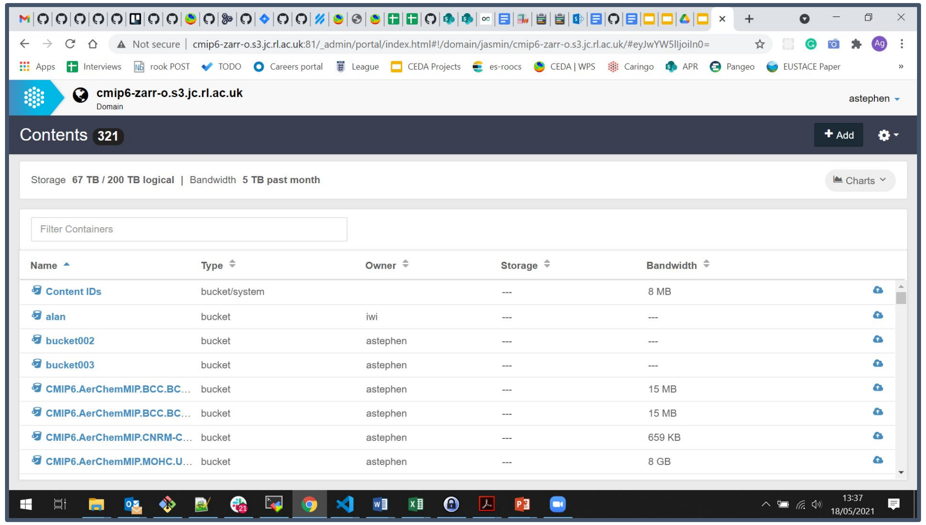
Traditional File Systems (POSIX)

- Access model is limiting – user management fixed to operating system – if you don't have a JASMIN id, you can't access it
- Strain on this model scaling with large file systems

Object Store

- Access is by HTTP so data can potentially be accessed from anywhere
- Provides a way to share data between the External Cloud and other parts of JASMIN
- Naturally scales

CMIP6 in the JASMIN Object Store



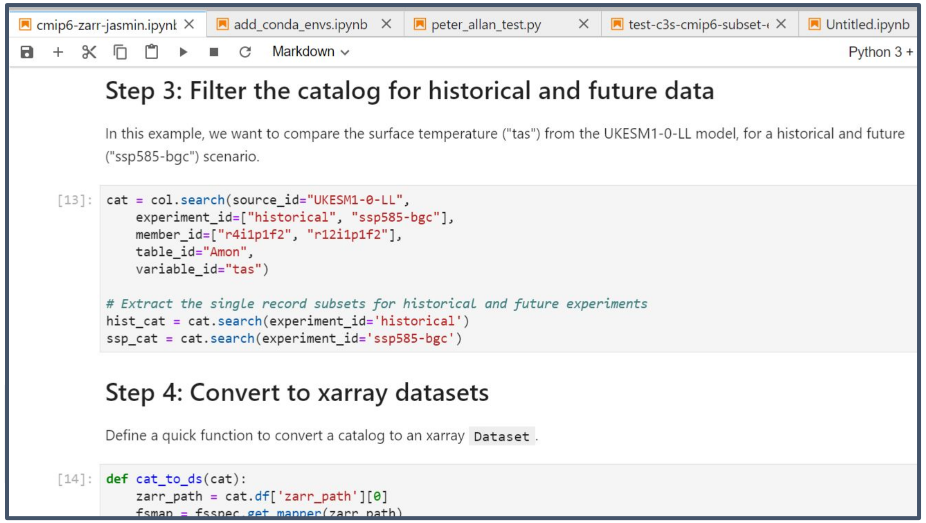
cmip6-zarr-o.s3.jc.rl.ac.uk
Domain

Contents 321

Storage **67 TB / 200 TB logical** | Bandwidth **5 TB past month**

Filter Containers

Name	Type	Owner	Storage	Bandwidth
Content IDs	bucket/system		---	8 MB
alan	bucket	iwi	---	---
bucket002	bucket	astephen	---	---
bucket003	bucket	astephen	---	---
CMIP6.AerChemMIP.BCC.BC…	bucket	astephen	---	15 MB
CMIP6.AerChemMIP.BCC.BC…	bucket	astephen	---	15 MB
CMIP6.AerChemMIP.CNRM-C…	bucket	astephen	---	659 KB
CMIP6.AerChemMIP.MOHC.U…	bucket	astephen	---	8 GB



Step 3: Filter the catalog for historical and future data

In this example, we want to compare the surface temperature ("tas") from the UKESM1-0-LL model, for a historical and future ("ssp585-bgc") scenario.

```
[13]: cat = col.search(source_id="UKESM1-0-LL",
          experiment_id=["historical", "ssp585-bgc"],
          member_id=["r4i1p1f2", "r12i1p1f2"],
          table_id="Amon",
          variable_id="tas")

      # Extract the single record subsets for historical and future experiments
      hist_cat = cat.search(experiment_id='historical')
      ssp_cat = cat.search(experiment_id='ssp585-bgc')
```

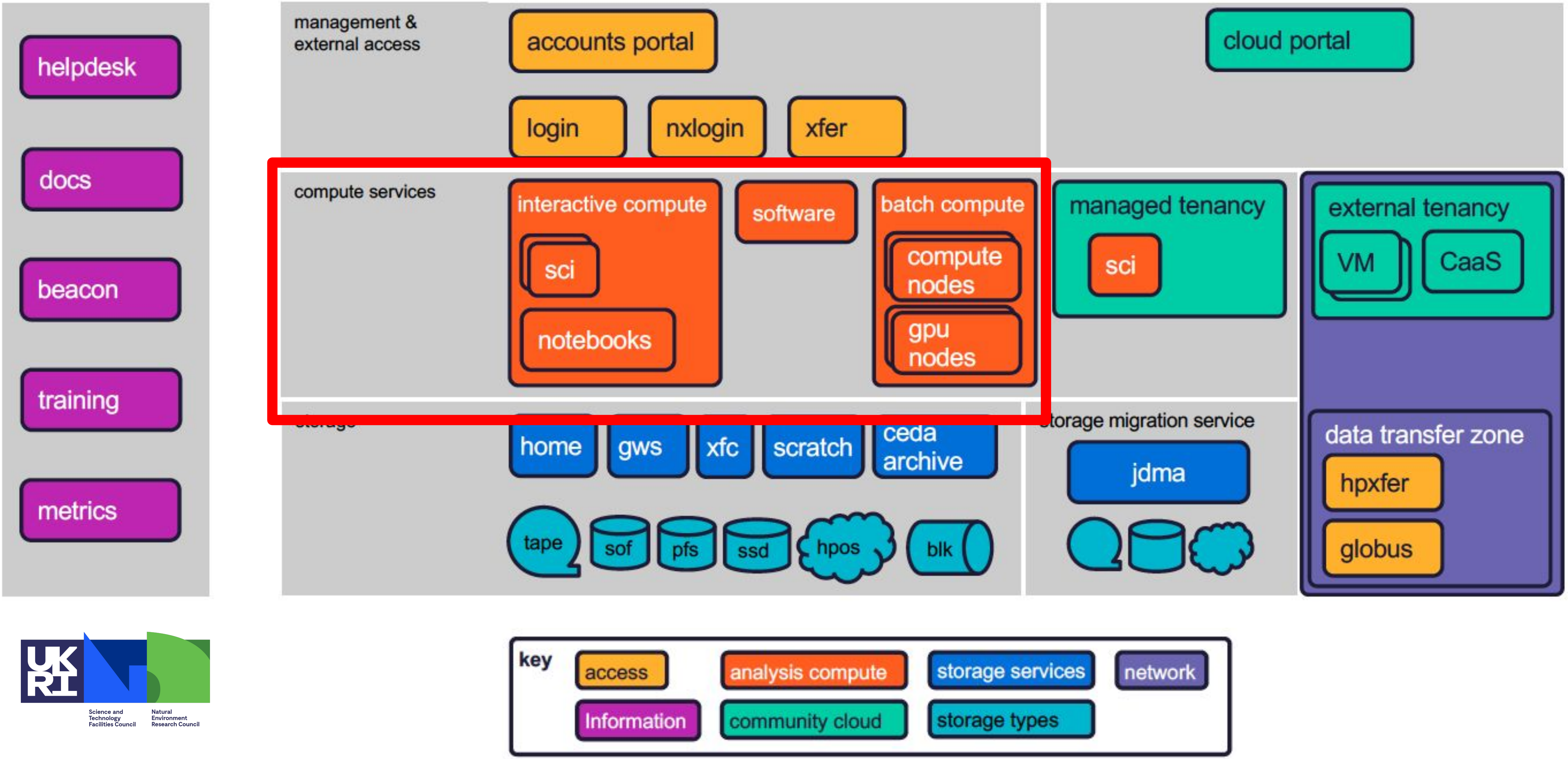
Step 4: Convert to xarray datasets

Define a quick function to convert a catalog to an xarray `Dataset`.

```
[14]: def cat_to_ds(cat):
          zarr_path = cat.df['zarr_path'][0]
          fsmap = fsspec.get_mapper(zarr_path)
```

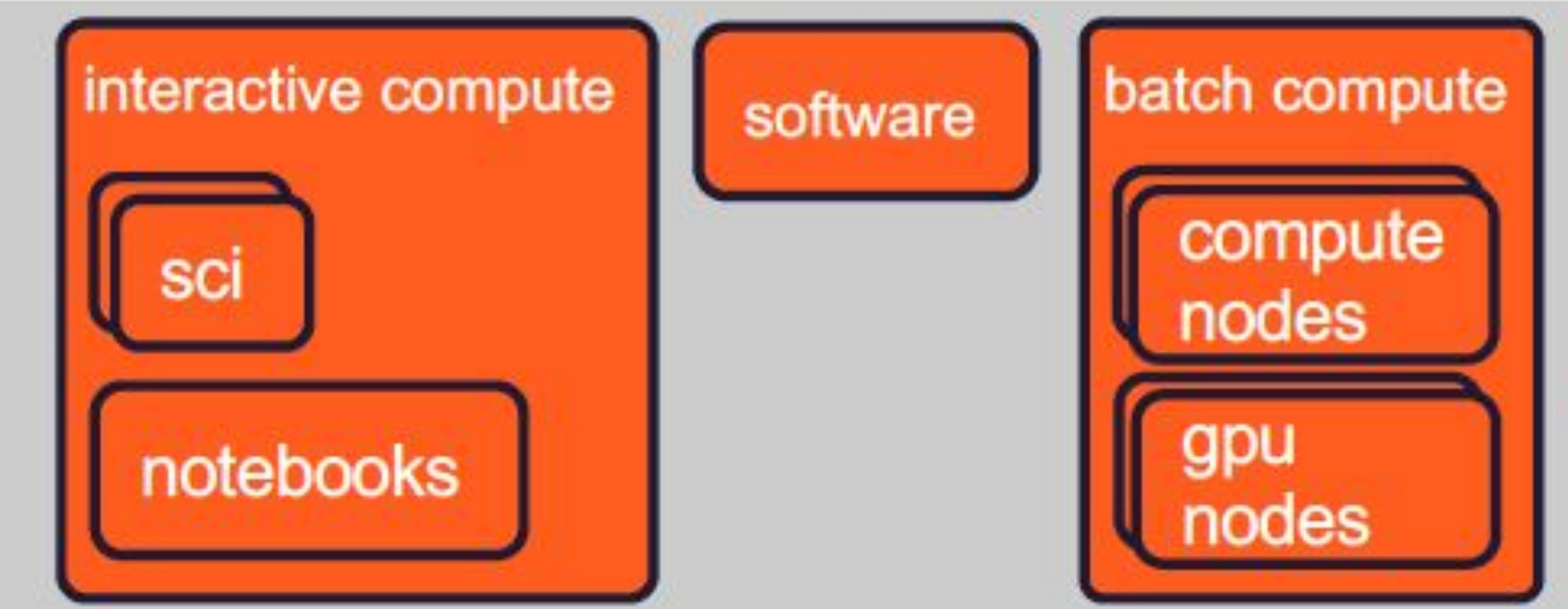
- We have developed a tool for converting our CMIP6 holdings to Zarr format written in the JASMIN Object Store.
- Processed as each ESGF Dataset:
  - One variable, model, expt, ensemble
  - All files in a time series
- Zarr files save "chunks" (i.e. sub-arrays) as individual objects.
- Notebook interface uses Intake-ESM catalog, Xarray, Matplotlib (PANGEO stack)
- Available to JASMIN users
- ~70Tb loaded so far
- Prototype stage at present

JASMIN Services



JASMIN Compute

compute services


interactive compute

sci

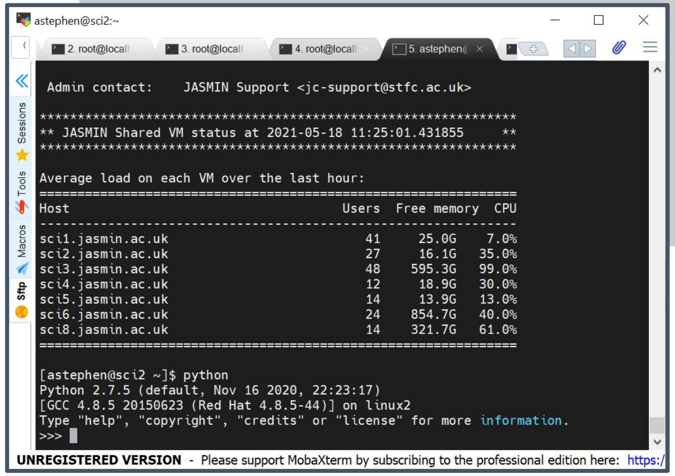
notebooks

software

batch compute

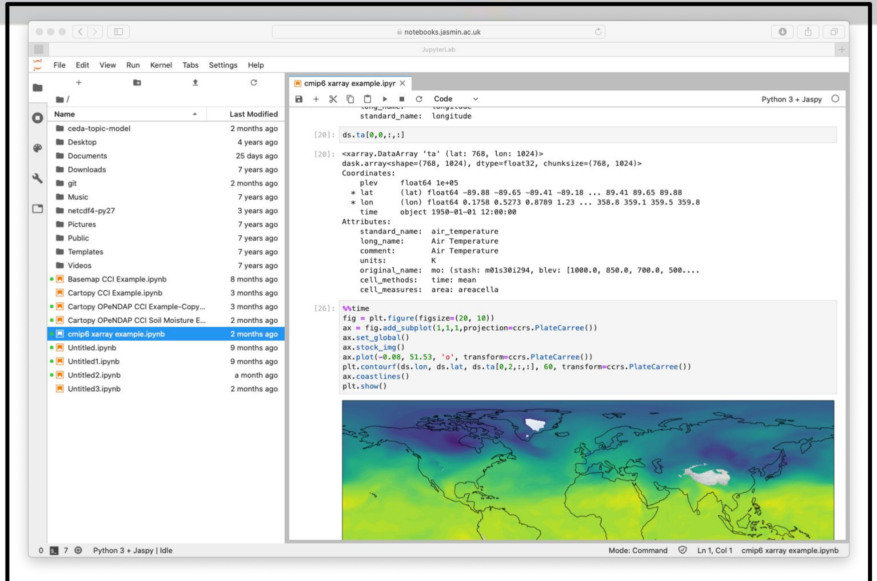
compute nodes

gpu nodes



```
 Admin contact:    JASMIN Support <jc-support@stfc.ac.uk>

***************************************************************
** JASMIN Shared VM status at 2021-05-18 11:25:01.431855     **
***************************************************************

Average load on each VM over the last hour:
===============================================================
Host                                     Users  Free memory  CPU
---------------------------------------------------------------
sci1.jasmin.ac.uk                           41      25.0G    7.0%
sci2.jasmin.ac.uk                           27      16.1G   35.0%
sci3.jasmin.ac.uk                           48     595.3G   99.0%
sci4.jasmin.ac.uk                           12      18.9G   30.0%
sci5.jasmin.ac.uk                           14      13.9G   13.0%
sci6.jasmin.ac.uk                           24     854.7G   40.0%
sci8.jasmin.ac.uk                           14     321.7G   61.0%
===============================================================

[astephen@sci2 ~]$ python
Python 2.7.5 (default, Nov 16 2020, 22:23:17)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https:/



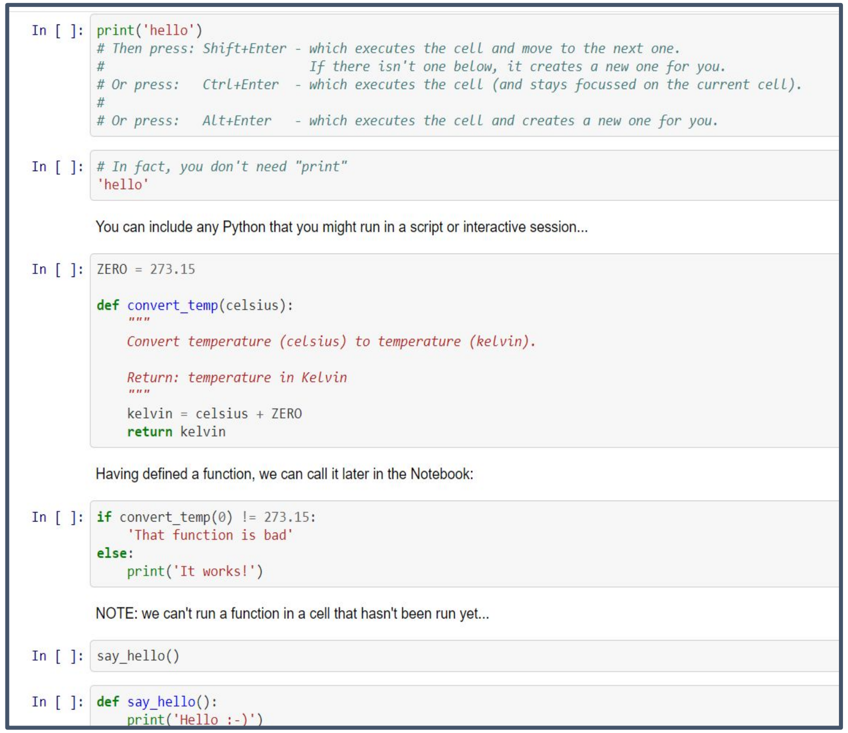
Name	Last Modified
ceda-topic-model	2 months ago
Desktop	4 years ago
Documents	25 days ago
Downloads	7 years ago
git	2 months ago
Music	7 years ago
netcdf4-py27	3 years ago
Pictures	7 years ago
Public	7 years ago
Templates	7 years ago
Videos	7 years ago
Basemap CCI Example.ipynb	8 months ago
Cartopy CCI Example.ipynb	3 months ago
Cartopy OPeNDAP CCI Example-Copy...	3 months ago
Cartopy OPeNDAP CCI Soil Moisture E...	2 months ago
cmip6 xarray example.ipynb	2 months ago
Untitled.ipynb	9 months ago
Untitled1.ipynb	9 months ago
Untitled2.ipynb	a month ago
Untitled3.ipynb	2 months ago

```
    standard_name:  longitude

[20]: ds.ta[0,0,:,:]

[20]: <xarray.DataArray 'ta' (lat: 768, lon: 1024)>
      dask.array<shape=(768, 1024), dtype=float32, chunksize=(768, 1024)>
      Coordinates:
          plev     float64 1e+05
        * lat      (lat) float64 -89.88 -89.65 -89.41 -89.18 ... 89.41 89.65 89.88
        * lon      (lon) float64 0.1758 0.5273 0.8789 1.23 ... 358.8 359.1 359.5 359.8
          time     object 1950-01-01 12:00:00
      Attributes:
          standard_name:  air_temperature
          long_name:      Air Temperature
          comment:        Air Temperature
          units:          K
          original_name:  mo: (stash: m01s30i294, blev: [1000.0, 850.0, 700.0, 500....
          cell_methods:   time: mean
          cell_measures:  area: areacella

[26]: %%time
      fig = plt.figure(figsize=(20, 10))
      ax = fig.add_subplot(1,1,1,projection=ccrs.PlateCarree())
      ax.set_global()
      ax.stock_img()
      ax.plot(-0.08, 51.53, 'o', transform=ccrs.PlateCarree())
      plt.contourf(ds.lon, ds.lat, ds.ta[0,2,:,:], 60, transform=ccrs.PlateCarree())
      ax.coastlines()
      plt.show()
```

Science and Technology Facilities Council

Natural Environment Research Council

JASMIN Notebook Service



```
In [ ]: print('hello')
        # Then press: Shift+Enter - which executes the cell and move to the next one.
        #                           If there isn't one below, it creates a new one for you.
        # Or press:   Ctrl+Enter  - which executes the cell (and stays focussed on the current cell).
        #
        # Or press:   Alt+Enter   - which executes the cell and creates a new one for you.

In [ ]: # In fact, you don't need "print"
        'hello'
```

You can include any Python that you might run in a script or interactive session...

```
In [ ]: ZERO = 273.15

        def convert_temp(celsius):
            """
            Convert temperature (celsius) to temperature (kelvin).

            Return: temperature in Kelvin
            """
            kelvin = celsius + ZERO
            return kelvin
```

Having defined a function, we can call it later in the Notebook:

```
In [ ]: if convert_temp(0) != 273.15:
            'That function is bad'
        else:
            print('It works!')
```

NOTE: we can't run a function in a cell that hasn't been run yet...

```
In [ ]: say_hello()

In [ ]: def say_hello():
            print('Hello :-)')
```

A Jupyter Notebook is an interactive programming environment that runs in a web browser.

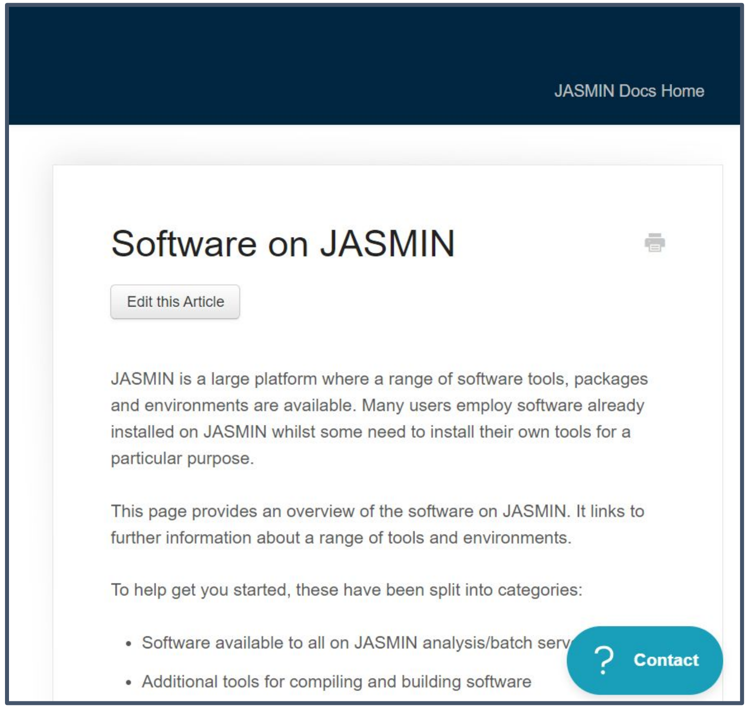
The JASMIN Notebook Service that allows you to:

- Define, edit and run code (in Python)
- Access a common (Jaspy) software environment
- Access data in the CEDA Archive and in your Group Workspaces
- Access data in the JASMIN Object Store

Science and Technology Facilities Council

Natural Environment Research Council

Software on JASMIN



Software on JASMIN

JASMIN is a large platform where a range of software tools, packages and environments are available. Many users employ software already installed on JASMIN whilst some need to install their own tools for a particular purpose.

This page provides an overview of the software on JASMIN. It links to further information about a range of tools and environments.

To help get you started, these have been split into categories:

- Software available to all on JASMIN analysis/batch serv
- Additional tools for compiling and building software

- Software is provided on analysis/batch/notebooks
- Compile / build / install software
- Restricted and server-specific software
- Data movement software

Jaspy environments

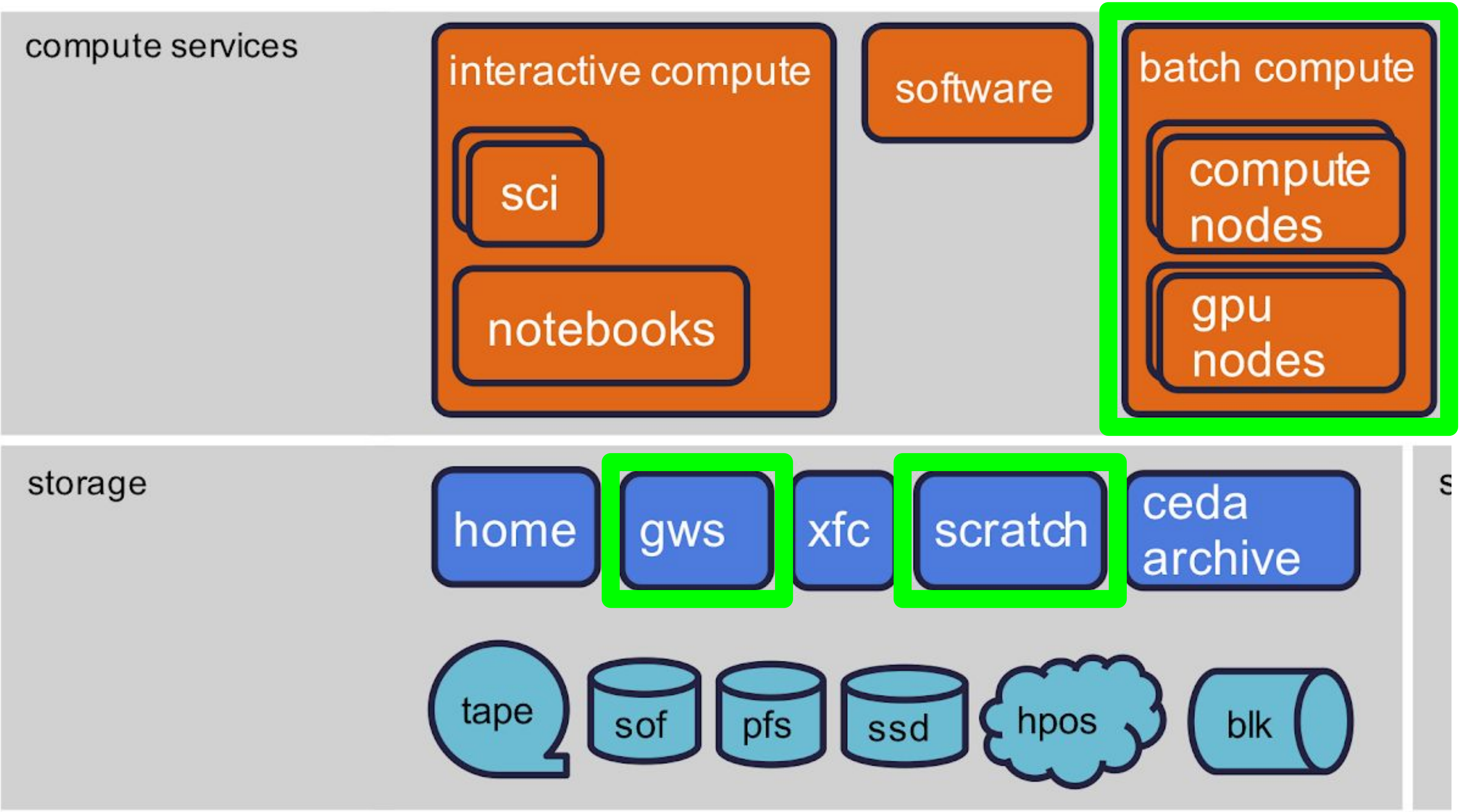
Python2.7 & others
Python3.7 & others

https://help.jasmin.ac.uk/article/273-software-on-jasmin

Issues related to batch compute

- Provide a heterogeneous batch environment:
  - Standard nodes: 16 core, 128Gb RAM
  - 1Tb nodes: 48 core, 1024Gb RAM
  - 2Tb nodes: 48 core, 2048Gb RAM
  - And others…
- Scheduler (SLURM) needs tuning to enable:
  - Prioritisation for major (funded/high-priority) projects
  - Fair-share for general use
- NOTE: users will look for ways of subverting the scheduler rules

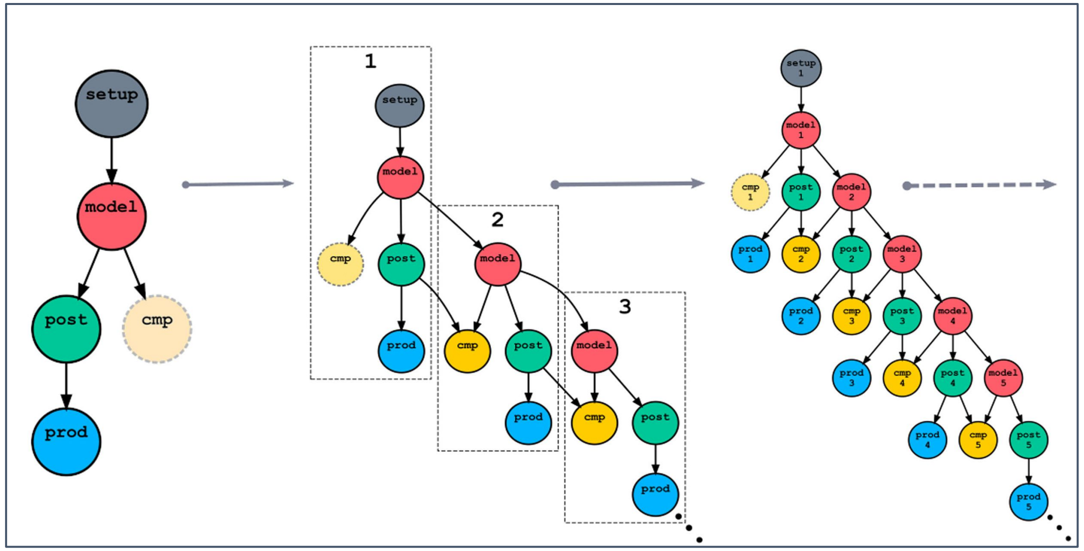
User workflows



Workflow tools - Rose/Cylc

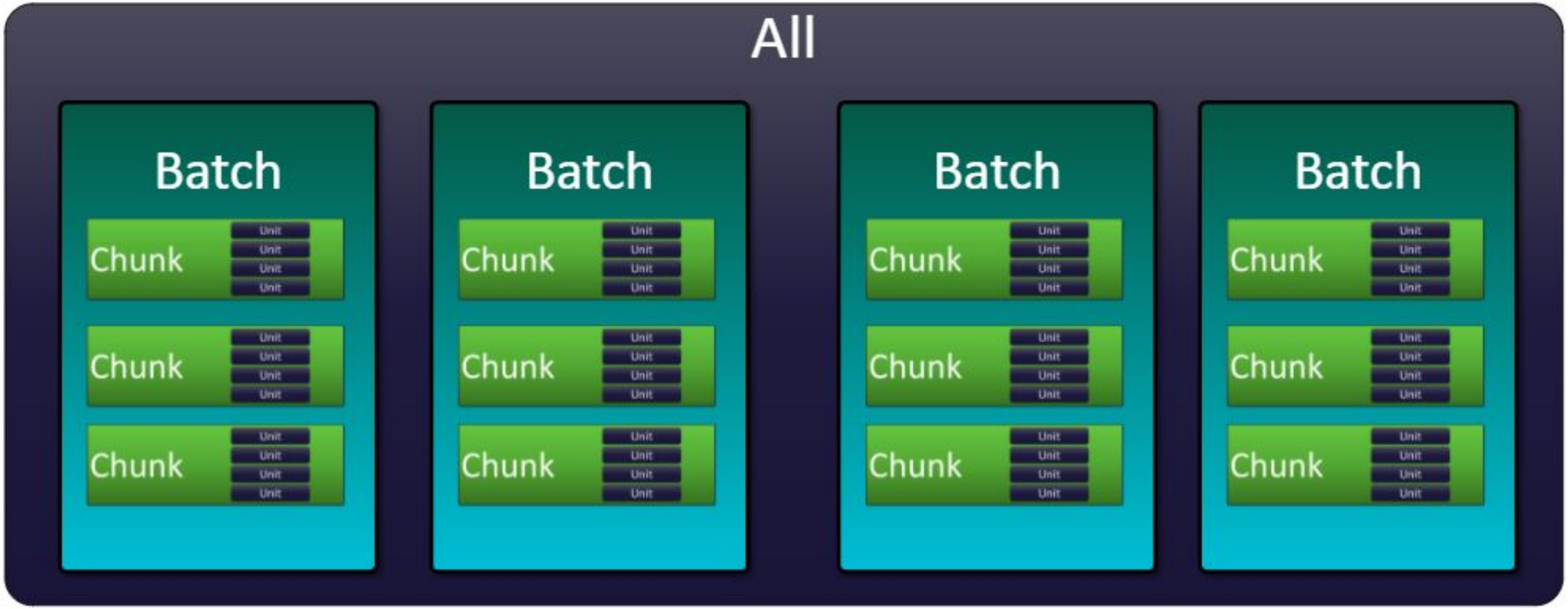
On JASMIN, we have installed **Rose/Cylc** (Met Office/NIWA):

- very good for multi-step workflows
- Includes a graphical interface
- talks to LOTUS
- sophisticated workflow management (retries etc)
- dedicated server



Science and Technology Facilities Council
Natural Environment Research Council

Encouraging good practice

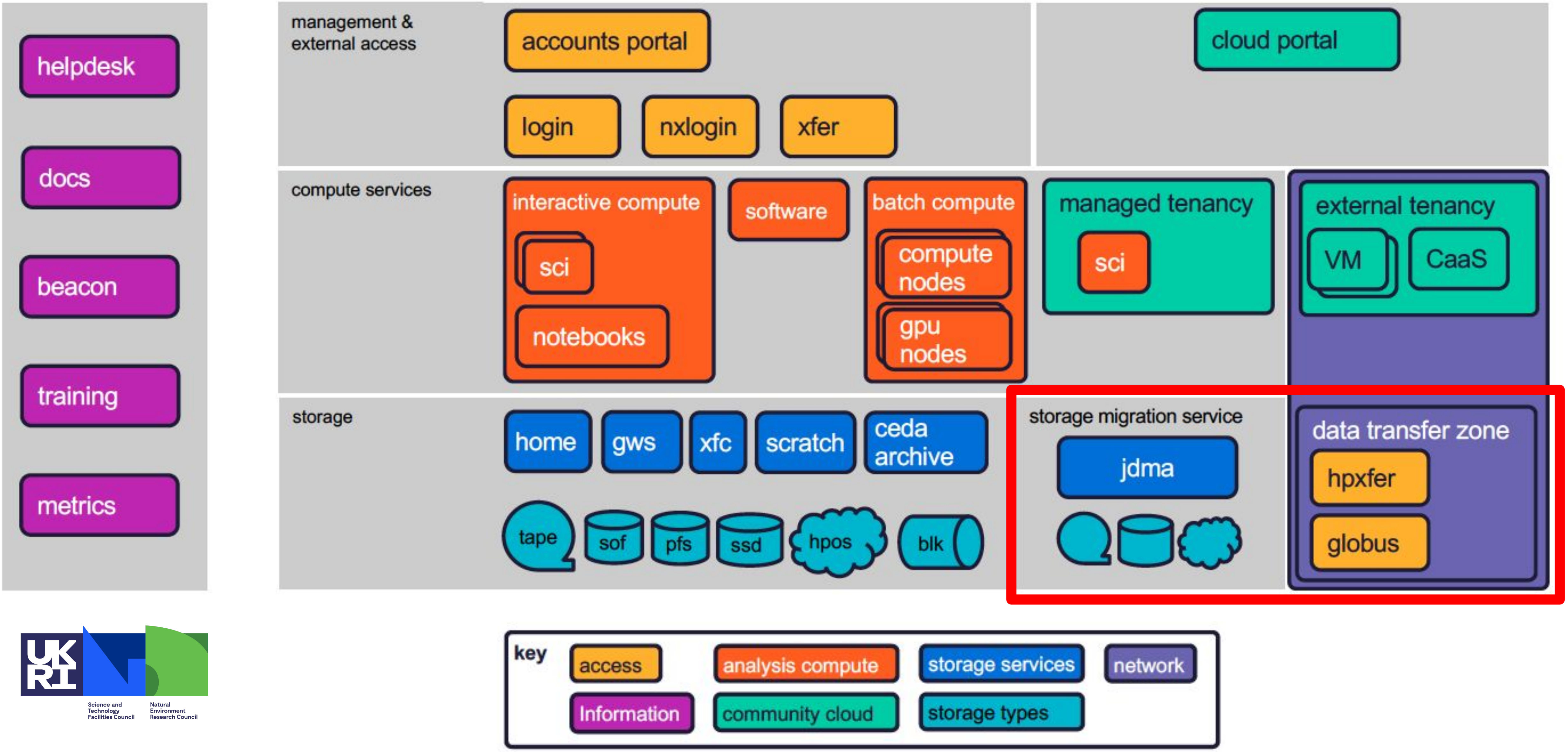


https://github.com/cedadev/abcunit-cmip5-stats

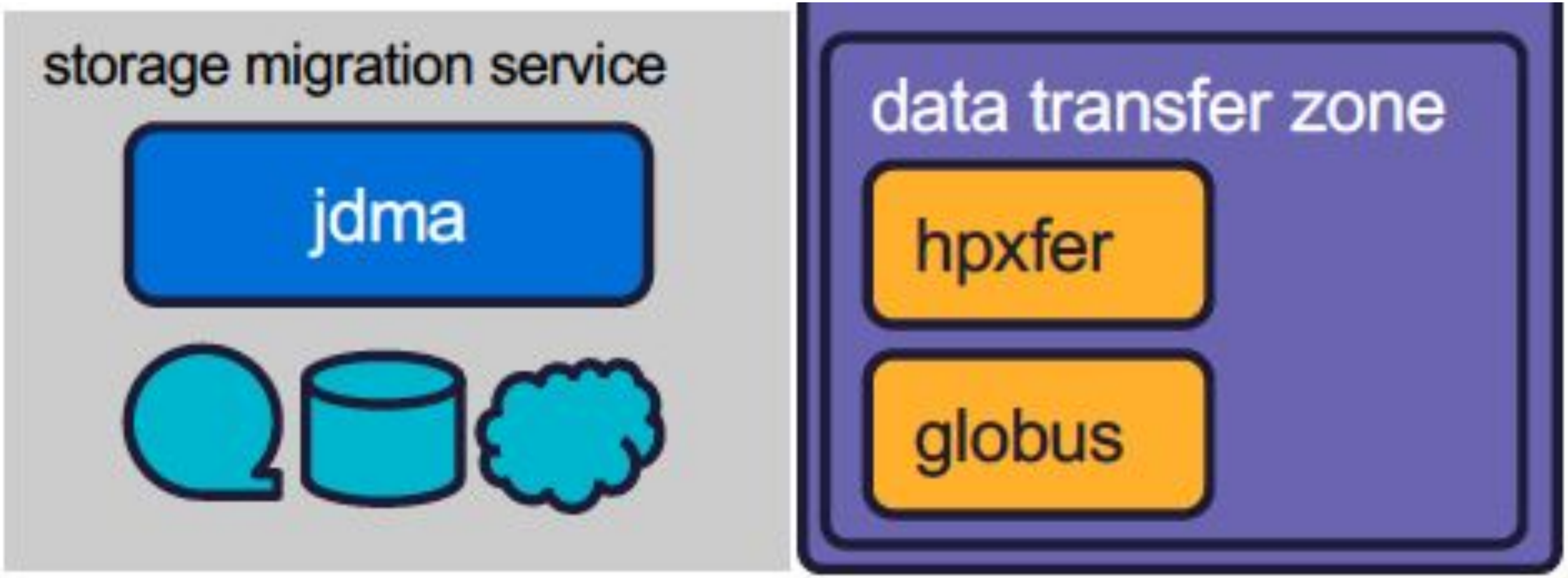
Science and Technology Facilities Council

Natural Environment Research Council

JASMIN Services



JASMIN: data transfer and migration



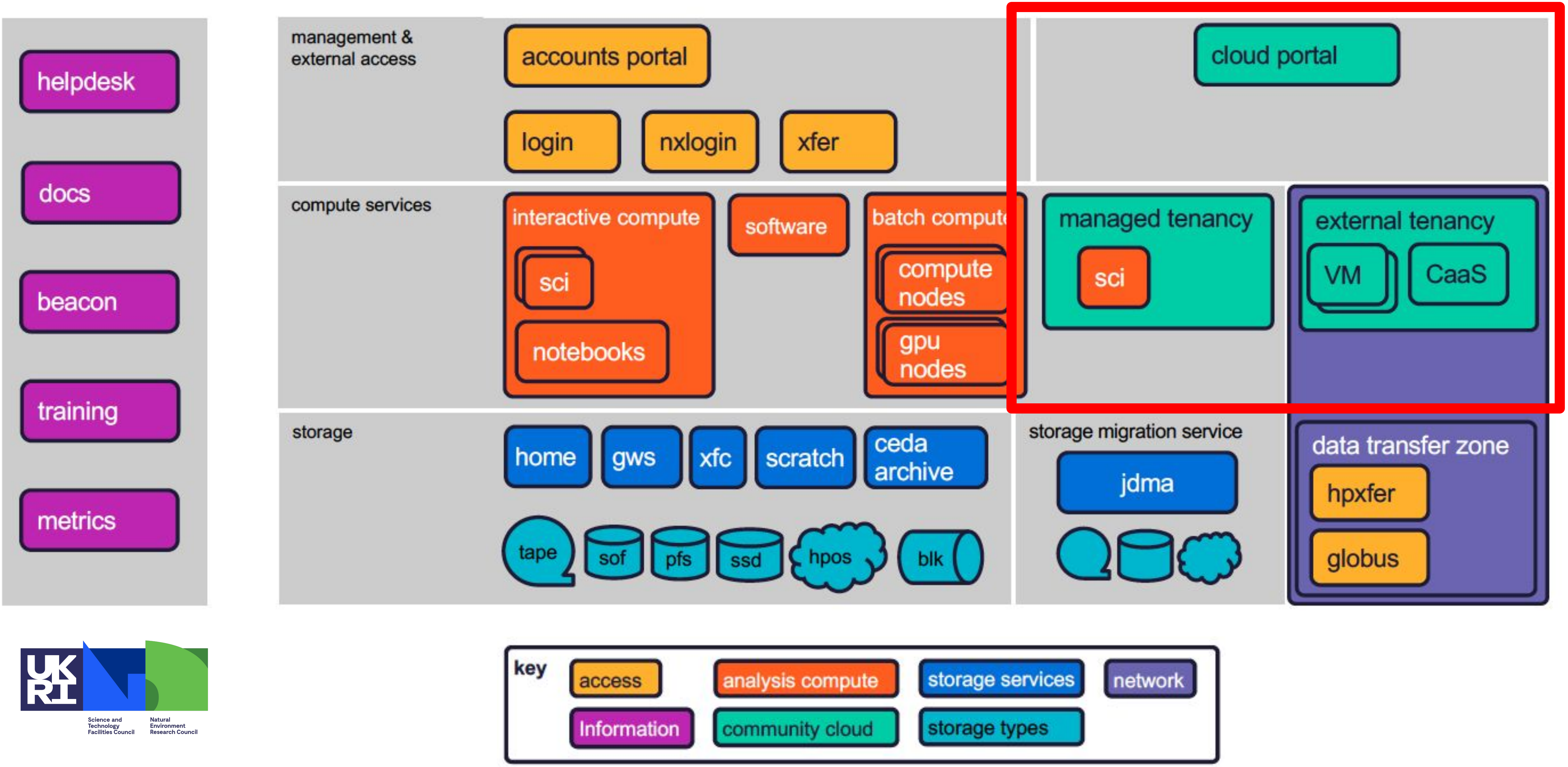
Issues related to data transfer and migration

- Use the best tool for the job
- Use the best transfer protocol
- Use the best server/service
- Incremental backup is required:
  - But hard to provide
  - Requires walking of the file-system (expensive because of parallel disk management of file-system metadata)
- Important question: "Do you need to move the data at all?"
- Can we reduce volumes using WPS subsetting, opendap etc?

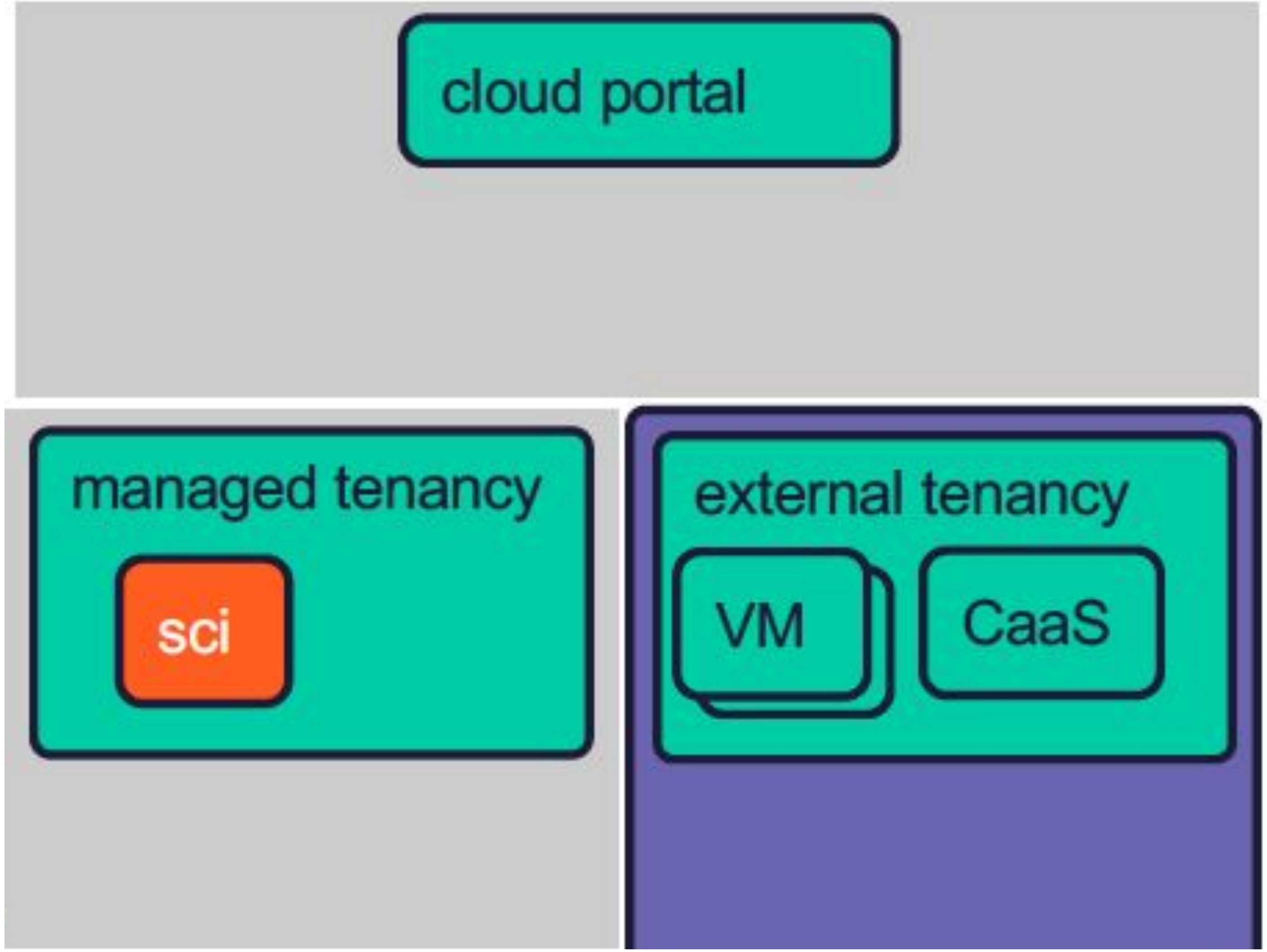
Science and Technology Facilities Council

Natural Environment Research Council

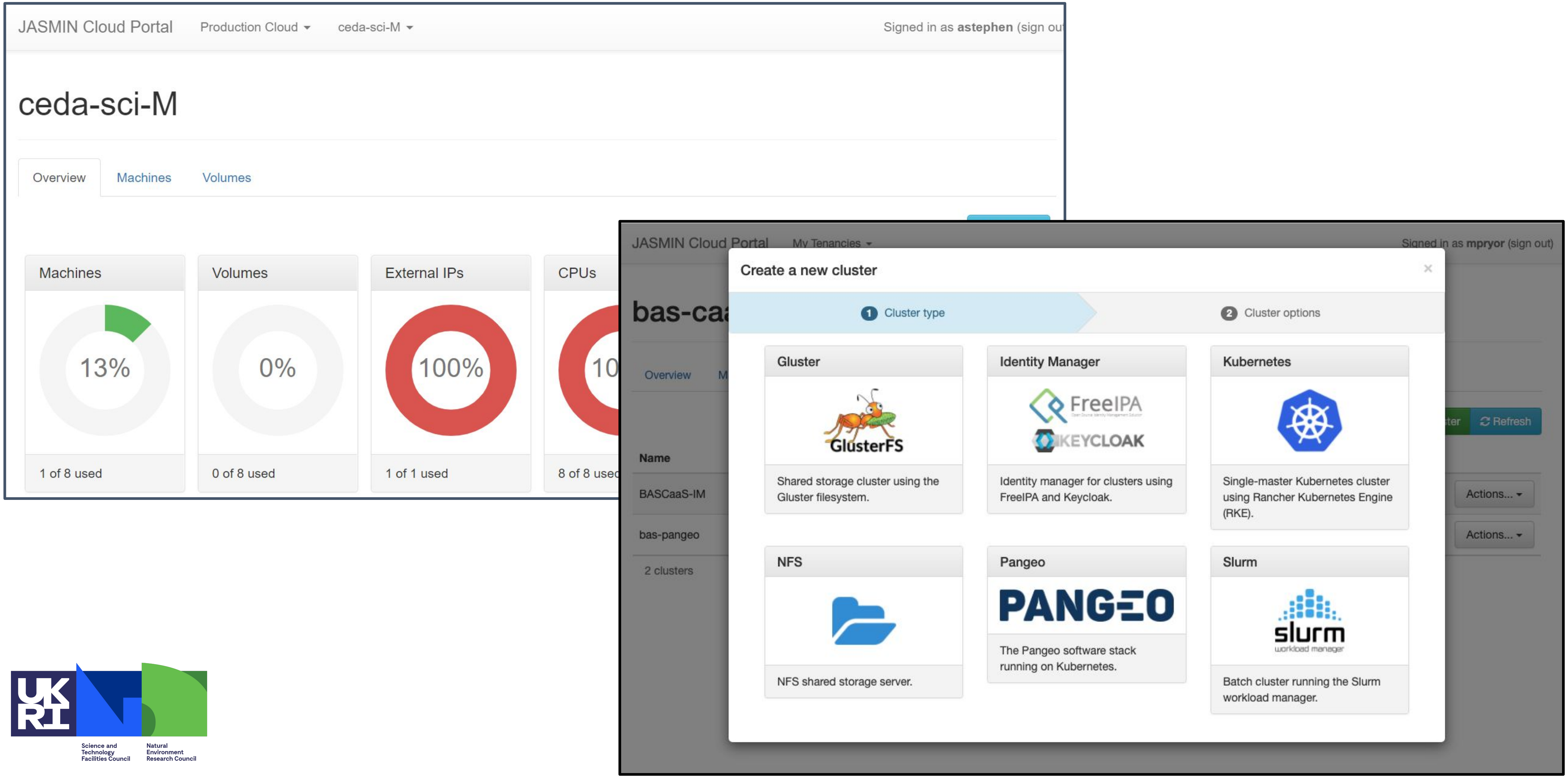
JASMIN Services



JASMIN: cloud



JASMIN: cloud



JASMIN Cloud Portal | Production Cloud ▾ | ceda-sci-M ▾ | Signed in as **astephen** (sign ou

ceda-sci-M

Overview | Machines | Volumes

Machines	Volumes	External IPs	CPUs
13%	0%	100%	10
1 of 8 used	0 of 8 used	1 of 1 used	8 of 8 used

JASMIN Cloud Portal | My Tenancies ▾ | Signed in as **mpryor** (sign out)

bas-ca

Overview | M

Name	
BASCaaS-IM	Actions... ▾
bas-pangeo	Actions... ▾

2 clusters

ter | Refresh

Create a new cluster ×

1 Cluster type | 2 Cluster options

Gluster
Shared storage cluster using the Gluster filesystem.

Identity Manager
Identity manager for clusters using FreeIPA and Keycloak.

Kubernetes
Single-master Kubernetes cluster using Rancher Kubernetes Engine (RKE).

NFS
NFS shared storage server.

Pangeo
The Pangeo software stack running on Kubernetes.

Slurm
Batch cluster running the Slurm workload manager.

Benefits of CaaS

- Dedicated clusters for your project
  - No competing for job slots with other users
- Clustering software like Kubernetes and Slurm is difficult to configure
  - Let CaaS do it for you!
- Cluster manager still gets root access
  - Apply customisations on top of CaaS-managed clusters
- Your users do not have to be JASMIN users
  - CaaS provides an identity management portal for your tenancy
  - Identities are integrated across all clusters in a tenancy
- Supports additional functionality
  - Use Kubernetes to build services for end users
  - Run a Jupyter notebook platform for your project

Issues related to cloud compute

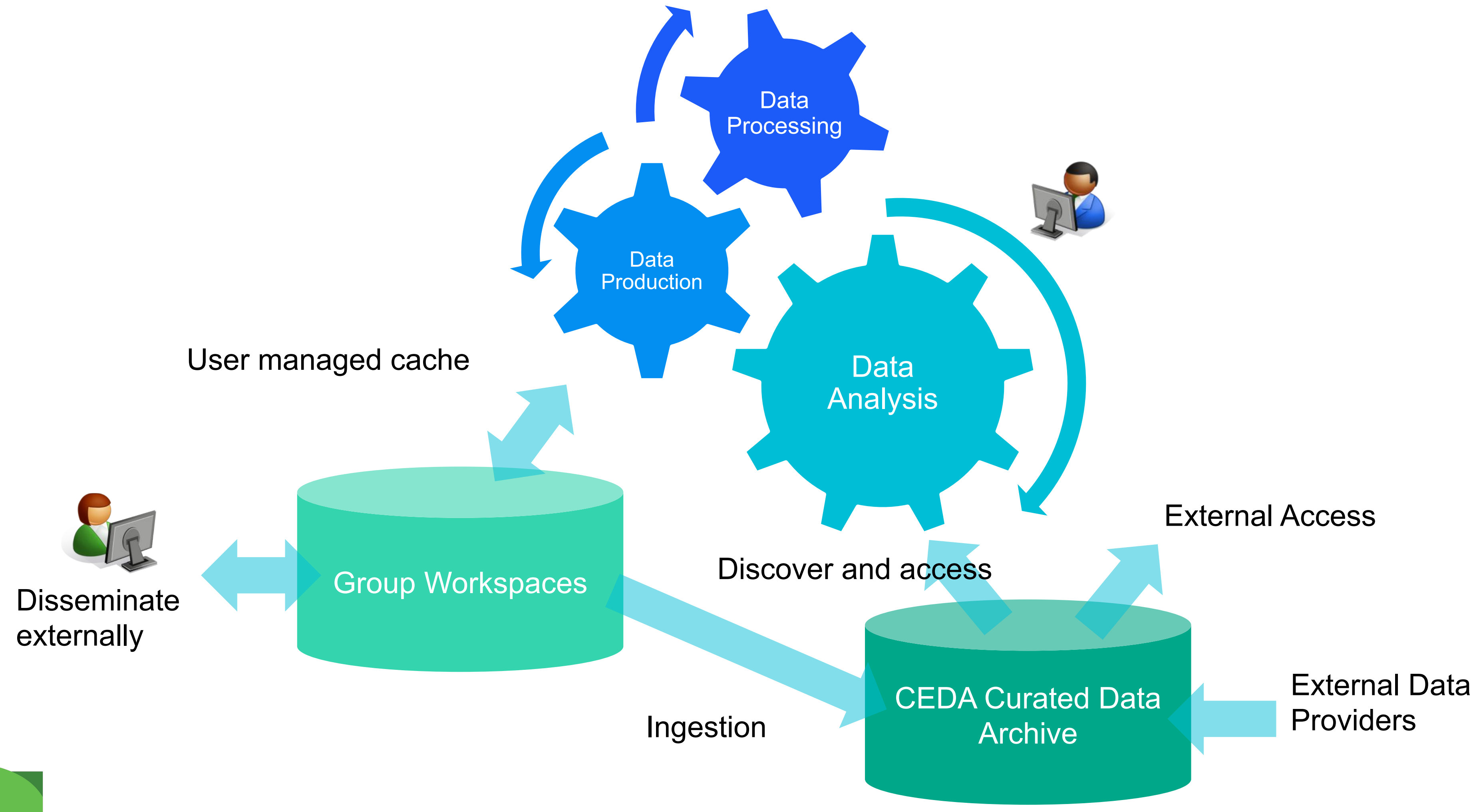
Background:

- Compute-as-a-Service (CaaS) environments in JASMIN Cloud
- Users running PANGEO and other Notebook environments
- Local storage and access to the JASMIN Object Store

Requirements:

- Provide scalable clusters for processing - e.g. enabling Dask for high-performance parallel data access and processing
    - In reality: resource implications to meet a range of different usage modes

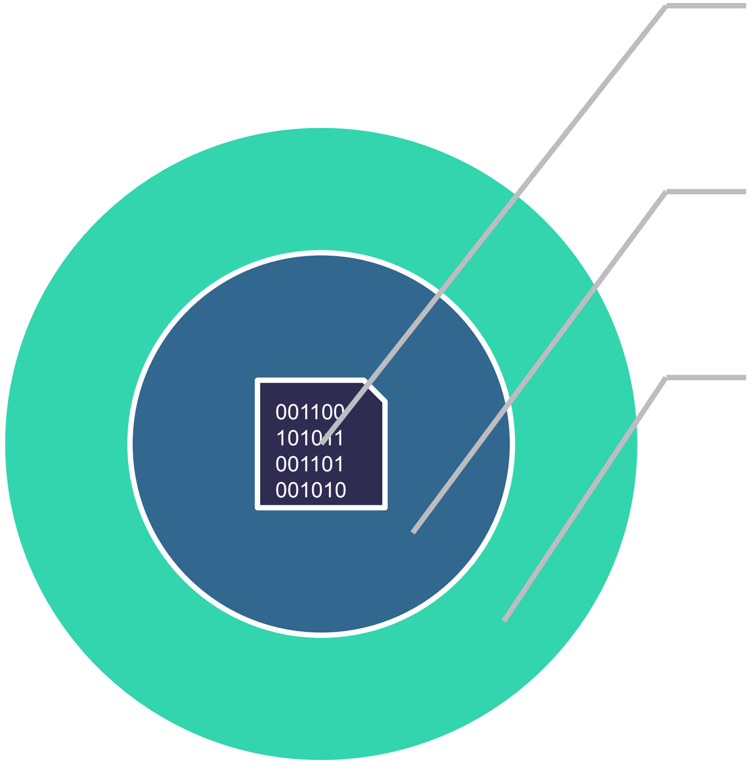
Common use case: Data Analysis and Sharing



Science and Technology Facilities Council

Natural Environment Research Council

Customisation and Environments

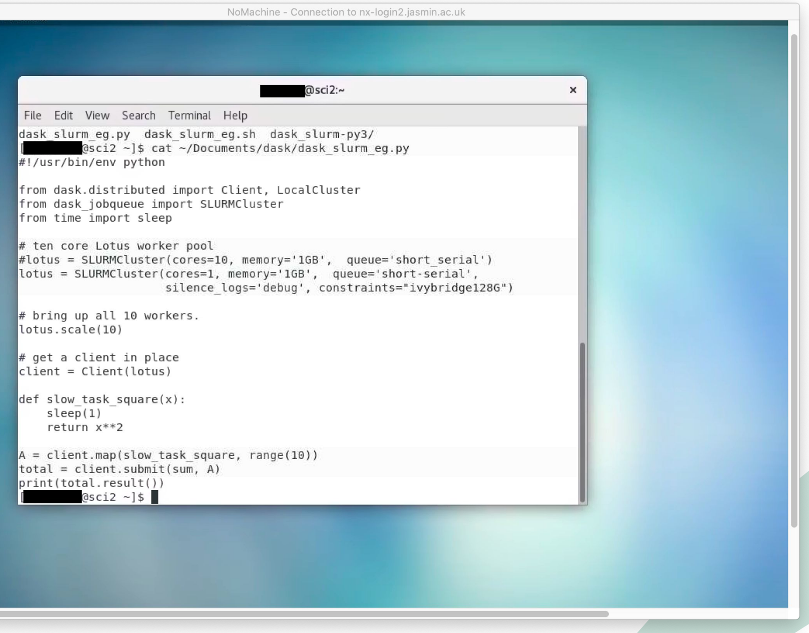


My code - I can control it ☺

Operating System - I can access a limited set of functionality from

JASMIN Infrastructure – fixed set of services providing resources that I can use

Customisation Evolution


NoMachine - Connection to nx-login2.jasmin.ac.uk
[redacted]@sci2:~
File Edit View Search Terminal Help

```
dask_slurm_eg.py  dask_slurm_eg.sh  dask_slurm-py3/
[[redacted]@sci2 ~]$ cat ~/Documents/dask/dask_slurm_eg.py
#!/usr/bin/env python

from dask.distributed import Client, LocalCluster
from dask_jobqueue import SLURMCluster
from time import sleep

# ten core Lotus worker pool
#lotus = SLURMCluster(cores=10, memory='1GB',  queue='short_serial')
lotus = SLURMCluster(cores=1, memory='1GB',  queue='short-serial',
                     silence_logs='debug', constraints="ivybridge128G")

# bring up all 10 workers.
lotus.scale(10)

# get a client in place
client = Client(lotus)

def slow_task_square(x):
    sleep(1)
    return x**2

A = client.map(slow_task_square, range(10))
total = client.submit(sum, A)
print(total.result())
[[redacted]@sci2 ~]$
```

Sci Analysis VMs

- static resources

Manage-your-own Sci Analysis VMs

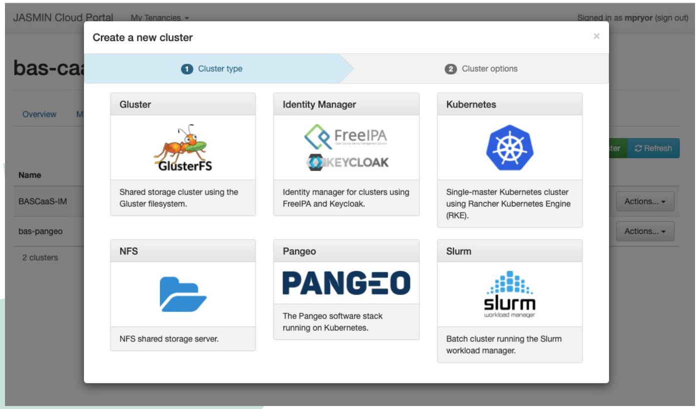
- Dynamic but fixed configuration
- Uses JASMIN Cloud

Jupyter Notebook Service

- Per user resources

Cluster-as-a-Service

- custom environments from building blocks
- On top of JASMIN Cloud



Create a new cluster

Name
BASCaaS-IM
bas-pangeo

2 clusters

Gluster – Shared storage cluster using the Gluster filesystem.

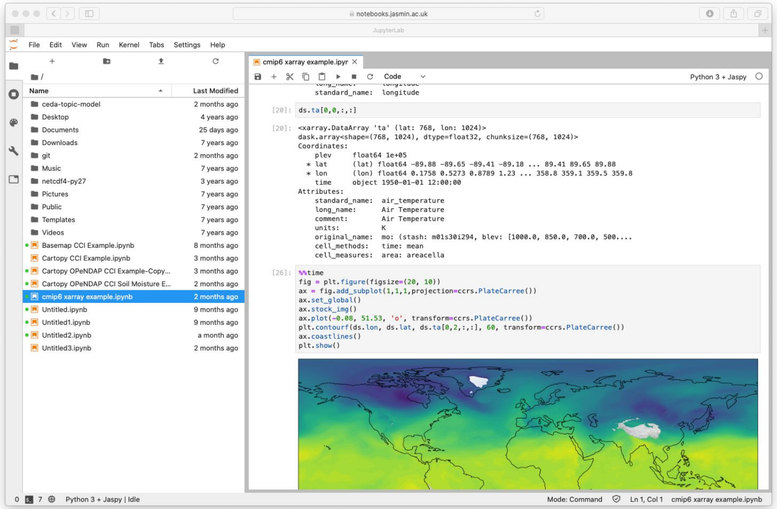
Identity Manager – Identity manager for clusters using FreeIPA and Keycloak.

Kubernetes – Single-master Kubernetes cluster using Rancher Kubernetes Engine (RKE).

NFS – NFS shared storage server.

Pangeo – The Pangeo software stack running on Kubernetes.

Slurm – Batch cluster running the Slurm workload manager.

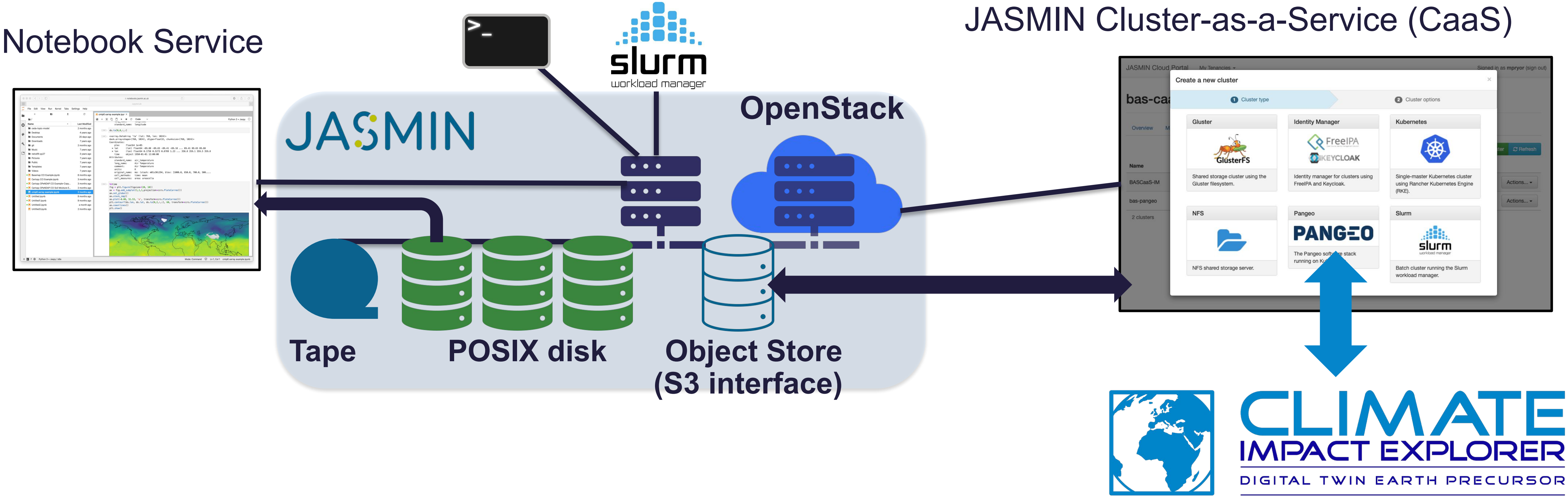

notebooks.jasmin.ac.uk
JupyterLab
File Edit View Run Kernel Tabs Settings Help
cmip6 xarray example.ipyr
Code
Python 3 + Jaspy
0 7 Python 3 + Jaspy | Idle
Mode: Command
Ln 1, Col 1
cmip6 xarray example.ipynb

Name	Last Modified
ceda-topic-model	2 months ago
Desktop	4 years ago
Documents	25 days ago
Downloads	7 years ago
git	2 months ago
Music	7 years ago
netcdf4-py27	3 years ago
Pictures	7 years ago
Public	7 years ago
Templates	7 years ago
Videos	7 years ago
Basemap CCI Example.ipynb	8 months ago
Cartopy CCI Example.ipynb	3 months ago
Cartopy OPeNDAP CCI Example-Copy...	3 months ago
Cartopy OPeNDAP CCI Soil Moisture E...	2 months ago
cmip6 xarray example.ipynb	2 months ago
Untitled.ipynb	9 months ago
Untitled1.ipynb	9 months ago
Untitled2.ipynb	a month ago
Untitled3.ipynb	2 months ago

```
    standard_name:  longitude

[20]: ds.ta[0,0,:,:]

[20]: <xarray.DataArray 'ta' (lat: 768, lon: 1024)>
      dask.array<shape=(768, 1024), dtype=float32, chunksize=(768, 1024)>
      Coordinates:
          plev     float64 1e+05
        * lat      (lat) float64 -89.88 -89.65 -89.41 -89.18 ... 89.41 89.65 89.88
        * lon      (lon) float64 0.1758 0.5273 0.8789 1.23 ... 358.8 359.1 359.5 359.8
          time     object 1950-01-01 12:00:00
      Attributes:
          standard_name:  air_temperature
          long_name:      Air Temperature
          comment:        Air Temperature
          units:          K
          original_name:  mo: (stash: m01s30i294, blev: [1000.0, 850.0, 700.0, 500....
          cell_methods:   time: mean
          cell_measures:  area: areacella

[26]: %%time
      fig = plt.figure(figsize=(20, 10))
      ax = fig.add_subplot(1,1,1,projection=ccrs.PlateCarree())
      ax.set_global()
      ax.stock_img()
      ax.plot(-0.08, 51.53, 'o', transform=ccrs.PlateCarree())
      plt.contourf(ds.lon, ds.lat, ds.ta[0,2,:,:], 60, transform=ccrs.PlateCarree())
      ax.coastlines()
      plt.show()
```

Science and Technology Facilities Council
Natural Environment Research Council

Bringing it all together:
DTEP Climate Impact Explorer



JASMIN Cloud Portal My Tenancies ▾ Signed in as **mpryor** (sign out)

bas-ca

Overview M

Name

BASCaaS-IM Actions... ▾

bas-pangeo Actions... ▾

2 clusters

ter ⟳ Refresh

Create a new cluster ×

1 Cluster type 2 Cluster options

Gluster
GlusterFS
Shared storage cluster using the Gluster filesystem.

Identity Manager
FreeIPA KEYCLOAK
Identity manager for clusters using FreeIPA and Keycloak.

Kubernetes
Single-master Kubernetes cluster using Rancher Kubernetes Engine (RKE).

NFS
NFS shared storage server.

Pangeo
PANGEO
The Pangeo soft re stack running on Ku

Slurm
slurm workload manager
Batch cluster running the Slurm workload manager.

Summary

- In the last decade, JASMIN has gone from an idea to a major platform in the UK scientific computing landscape
- Co-location of data and code is key
- Providing a variety of storage and processing solutions:
  - meets the needs of many users
  - requires significant effort (and therefore funding) to maintain/develop
- Tensions when supporting scientific users:
  - Must be flexible/scalable: we want to facilitate cutting-edge science
  - Must be pragmatic: users will generate requirements + find loopholes
- The future:
  - Mix of POSIX, object store and tape
  - Mix of traditional batch and new scalable processing (cloud)
  - More solutions that don't need to access the files until required: improved search and lazy evaluation

Science and
Technology
Facilities Council

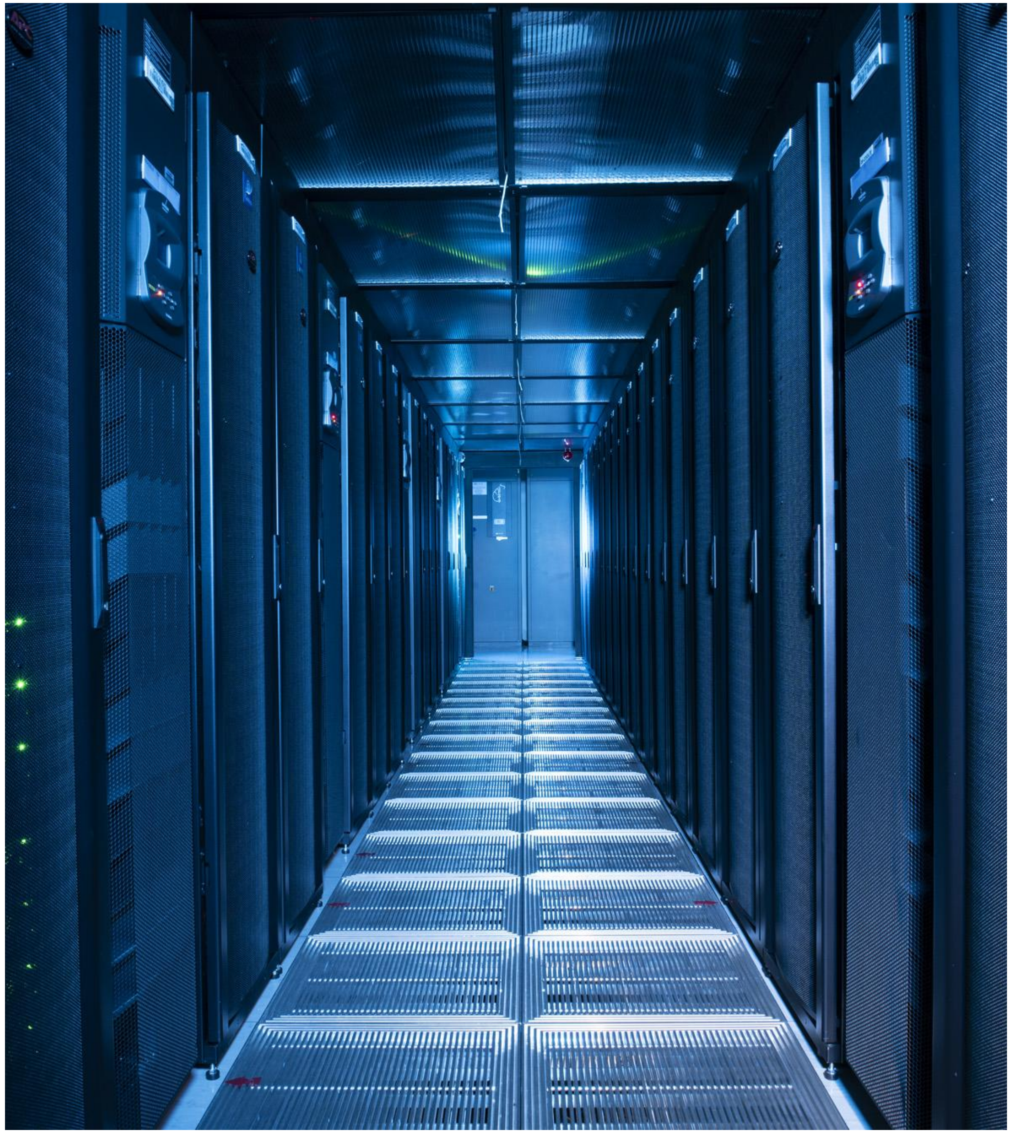
Natural
Environment
Research Council

Science and Technology Facilities Council

Natural Environment Research Council

JASMIN

Thank you!



Email - ag.stephens@stfc.ac.uk

GitHub - @agstephens

Websites:

- www.ceda.ac.uk
- www.jasmin.ac.uk

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

Scientific Computing
Science & Technology Facilities Council

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

RAL Space