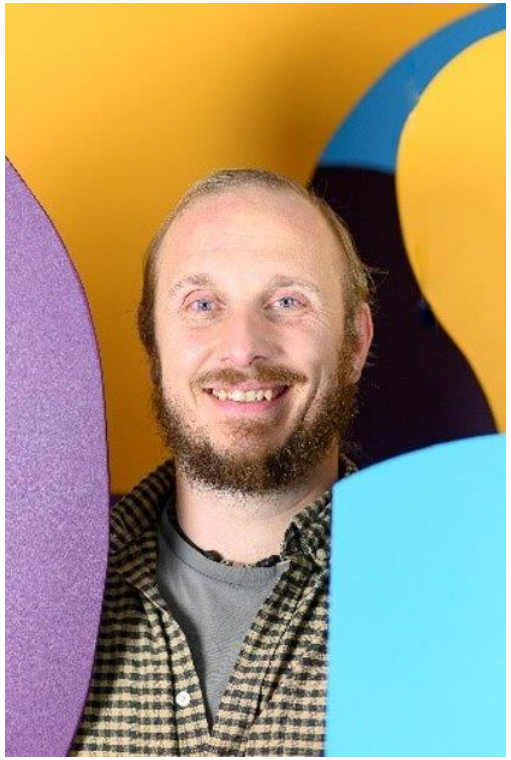
Interactive Geospatial Analysis at HPC scales: An Introduction to RS-DAT

netherlands eScience center

ITC UTwente 15-05-2023

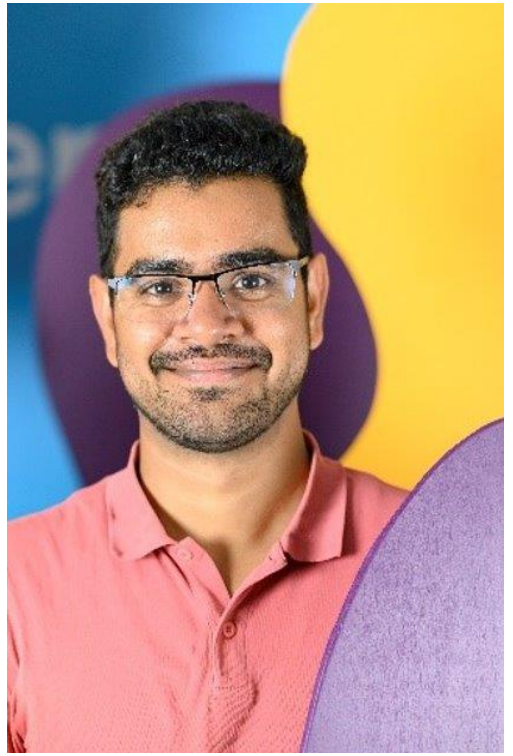
SURF

People



Meiert Grootes

Francesco Nattino



Pranav Chandramouli

netherlands eScience center

Housekeeping

https://tinyurl.com/2023-05-15-rs-dat

netherlands eScience center

Schedule

9:30 - 9:45	Welcome and icebreaker
9:45 - 10:00	SURF services for research and SPIDER (Guest speaker: Lodewijk Nauta, SURF)
10:00 - 11:00	HPC, RS-DAT, and the EO software ecosystem
11:00 - 11:15	Coffee break
11:15 - 12:30	Deployment with RS-DAT and data retrieval
12:30 - 13:30	Lunch break
13:30 - 15:00	Scaling EO workflows with HPC
15:00 - 15:15	Coffee break
15:15 - 16:30	Hands-on session
16:30 - 17:00	Wrap-up
17:00	END

netherlands eScience center

"Empowering researchers across all disciplines through advanced research software"

The Netherlands eScience Center

National centre / independent foundation since 2012 /

NWO & SURF

SURF

NWO

netherlands eScience center

MISSION

empowering researchers across all disciplines through innovative research software.

AMBITION 1
collaboratively designing software for research

AMBITION 2
building digital expertise

STRATEGIC PRIORITIES

- ✔ institutional alignment
- ✔ software sustainability

- open calls for eScience domain research
- collaborations in advanced eScience technologies
- workshops & training
- community building & networking

netherlands eScience center

Let’s stay in touch

Check for our open calls

www.eScienceCenter.nl

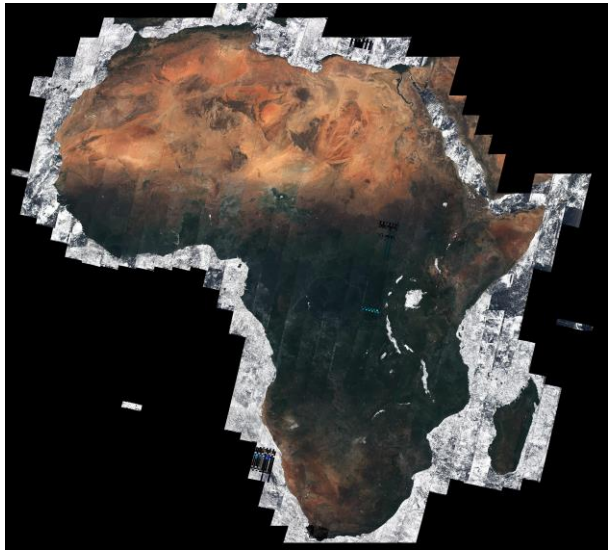
info@esciencecenter.nl

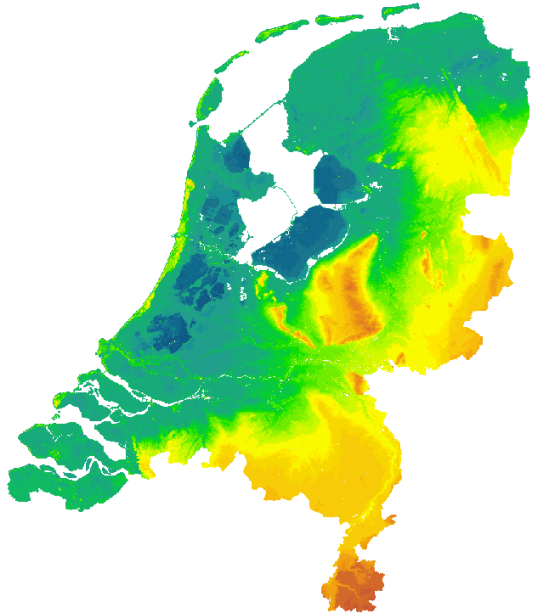
+31 20 460 4770

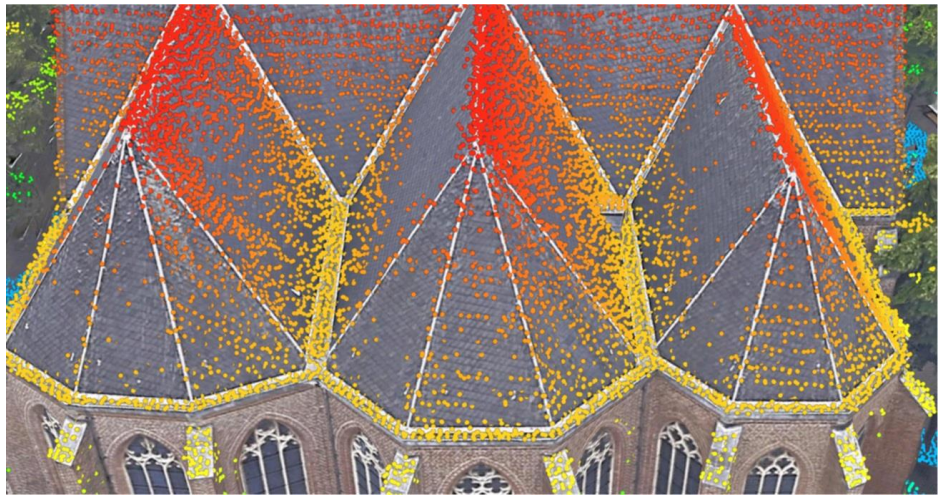
netherlands eScience center

From PC to HPC/HTC

- Data deluge across many/most scientific domains.
- Local storage and processing resources inadequate
- Institutional or national resources at larger scale needed
- ITC/UTwente well positioned with CRIB
- At (even) larger scales SURF offers combined storage and processing resources (a.o.)







SURF presentation (~10-15 min)

SURF services, specifically SPIDER

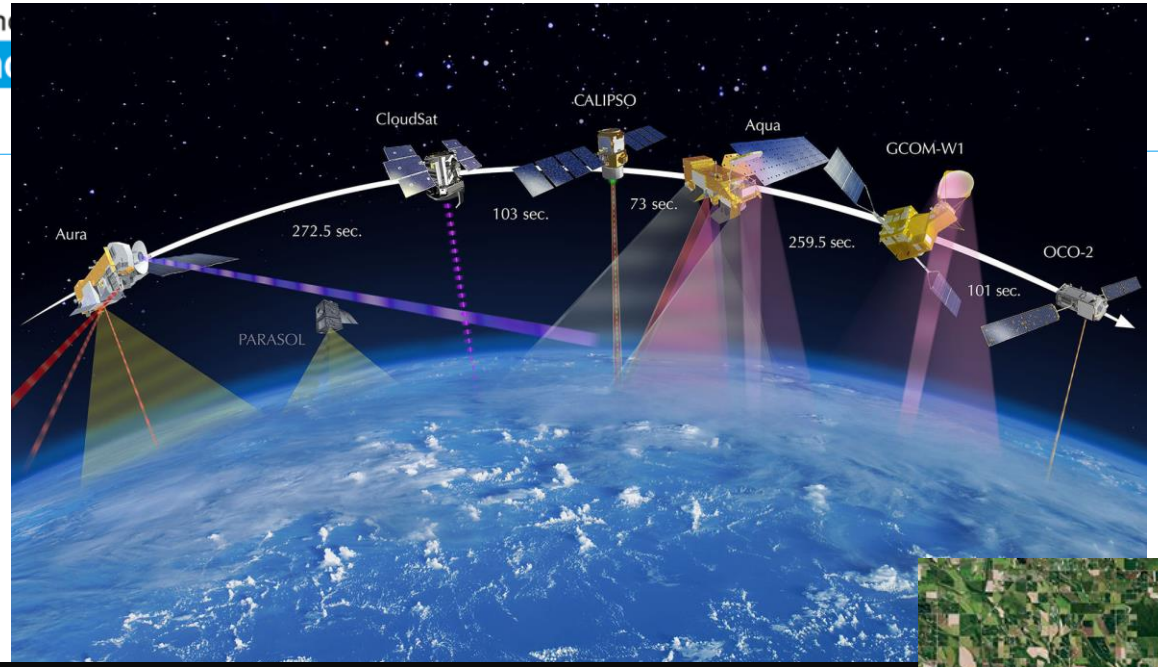
Interactive Geospatial Analysis at HPC scales: An Introduction to RS-DAT

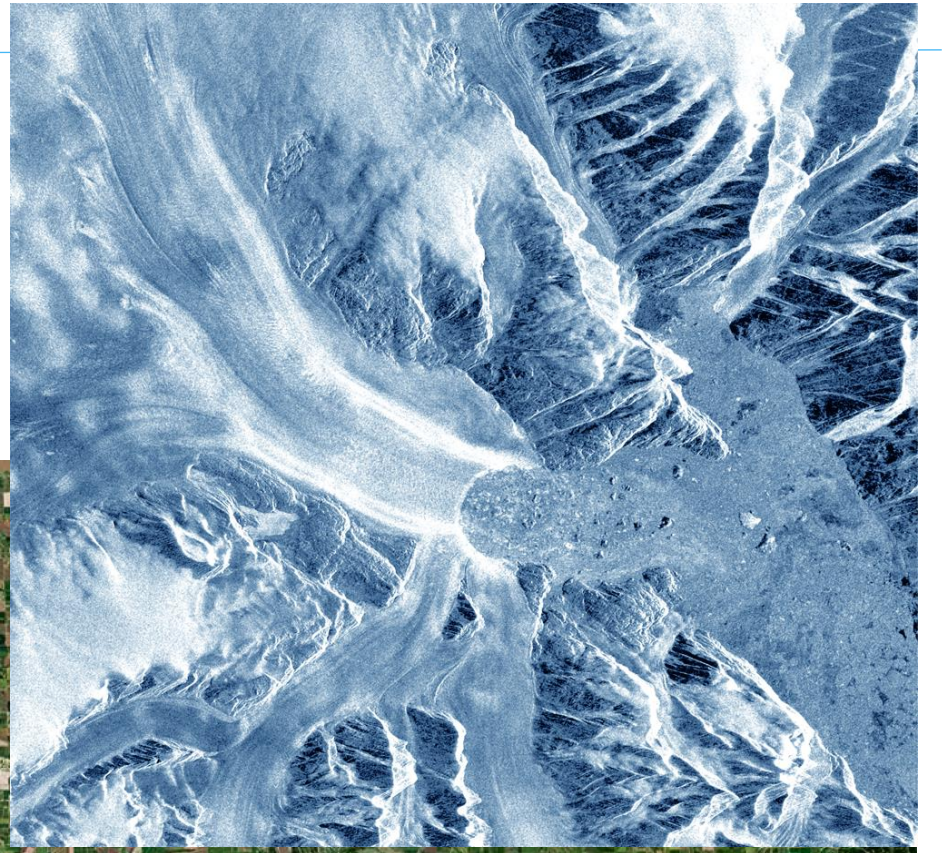
netherlands eScience center

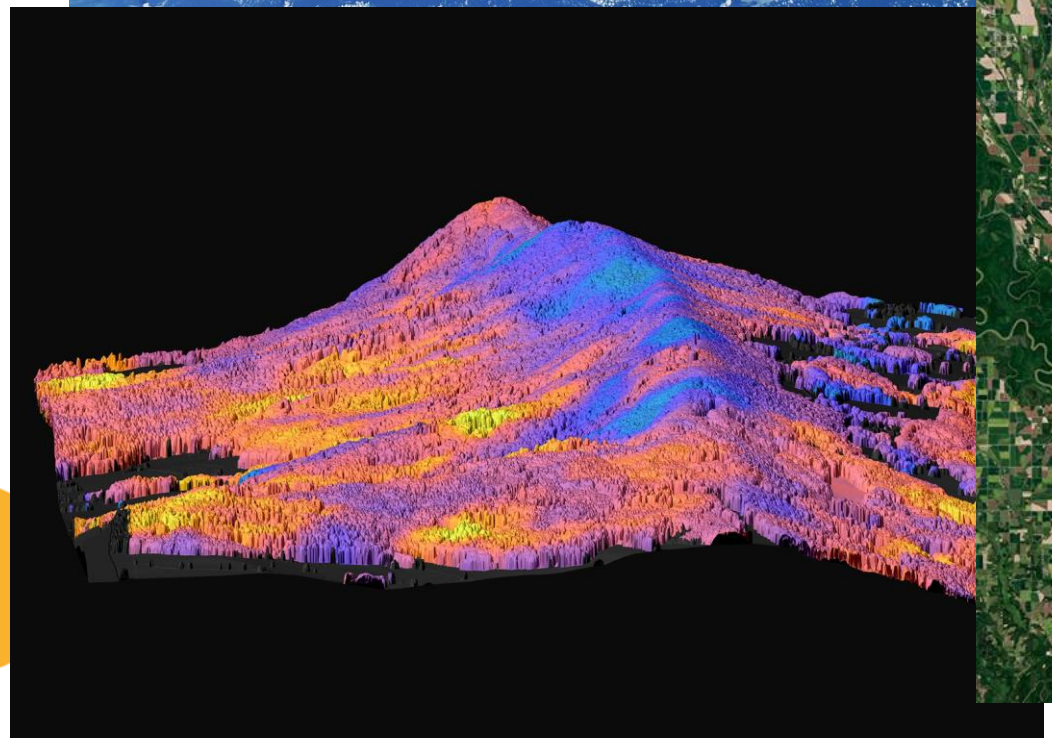
ITC UTwente 15-05-2023

SURF

EO as a scientific resource







All images Credit: ESA/NASA CC-BY

netherlands eScience center

EO as a scientific resource

Remote Sensing of Environment
Volume 101, Issue 4, 30 April 2006, Pages 447-462

Autonomous detection of cryospheric change with hyperion on-board Earth Observing-1

DOI: 10.1111/ddi.13468
RESEARCH ARTICLE
Diversity and Distributions WILEY

Better together? Assessing different remote sensing products for predicting habitat suitability of wetland birds

Zsófia Koma[1,2] | Arie C. Seijmonsbergen[1] | Meiert W. Grootes[3] | Francesco Nattino[3] | Jim Groot[1] | Henk Sierdsema[4] | Ruud P. B. Foppen[4,5] | W. Daniel Kissling[1,6]

JOURNAL OF GEOPHYSICAL RESEARCH
Atmospheres
Journal of Geophysical Research: Atmospheres homepage
Composition and Chemistry | Free Access

Global monitoring of air pollution over land from the Earth

Earth Observation Based Monitoring of Forests in Germany: A Review

by Stefanie Holzwarth [1,*], Frank Thonfeld [1,2], Sahra Abdullahi [1], Sarah Asam [1], Emmanuel Da Ponte Canova [1], Ursula Gessner [1], Juliane Huth [1], Tanja Kraus [1], Benjamin Leutner [1] and Claudia Kuenzer [1,2]

1 German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), 82234 Wessling, Germany
2 Institute of Geography and Geology, University of Wuerzburg, 97074 Wuerzburg, Germany

Open Access Editorial

Perspectives on "Earth Observation and GIScience for Agricultural Applications"

by Flavio Lupia [1,*], Jamal Jokar Arsanjani [2], Cidália C
Giuseppe Pulighe [1]

1 CREA Research Centre for Agricultural Policies and Bioeconomy, Via E
2 Geoinformatics Research Group, Department of Planning and Develop
A.C. Meyers Vænge 15, DK-2450 Copenhagen, Denmark
3 Department of Mathematics, University of Coimbra, Apartado 3008, EC
4 INESC Coimbra, DEEC, Rua Sílvio Lima, Pólo II, 3030-290 Coimbra, F
* Author to whom correspondence should be addressed.

ISPRS Int. J. Geo-Inf. **2022**, *11*(7), 372; https://doi.org/10.3390/ijgi11070

ISPRS Journal of Photogrammetry and Ren
Sensing
Volume 116, June 2016, Pages 55-72

Review Article

Optical remotely sensed time series d land cover classification: A review

Cristina Gómez [a,b], Joanne C. White [c], Michael A. Wulder [c]

Show more

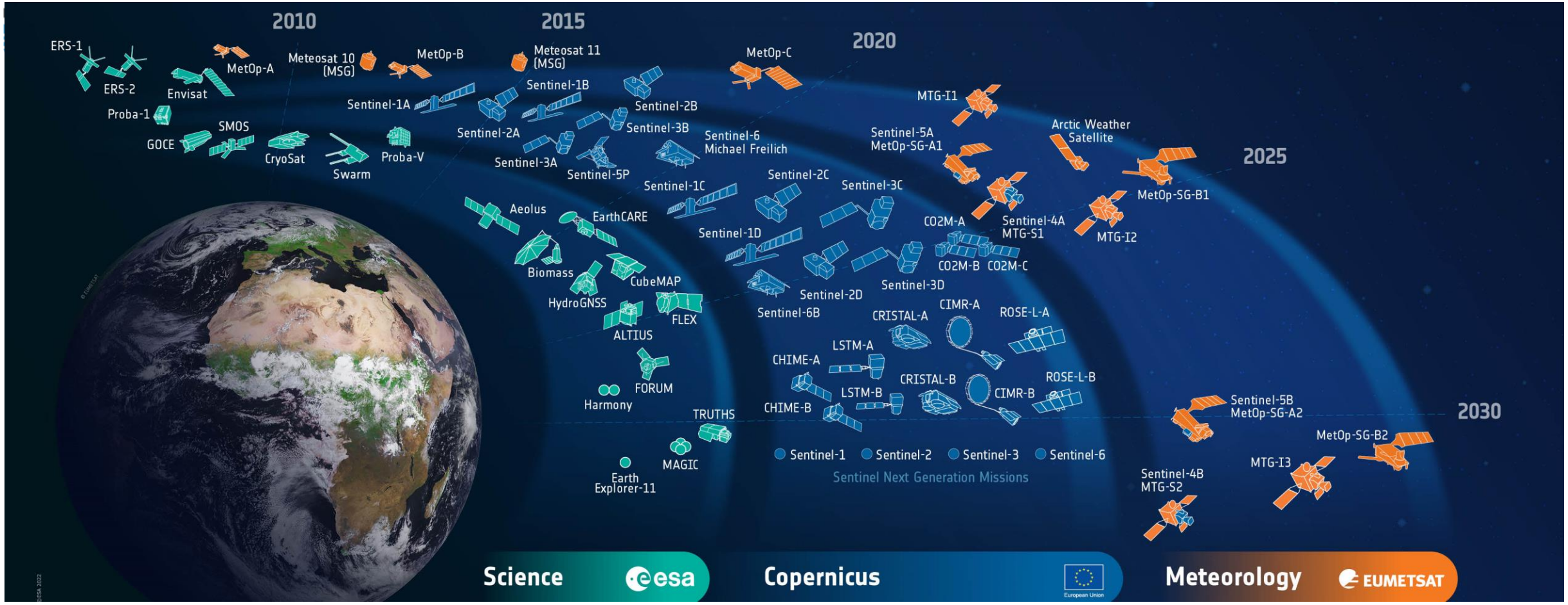
+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.isprsjprs.2016.03.008

Remote Sensing of Environment
Volume 120, 15 May 2012, Pages 91-101

Sentinels for science: Potential of Sentinel-1, -2, and -3 missions for scientific observations of ocean, cryosphere, and land

Zbyněk Malenovský [a], Helmut Rott [b,c], Josef Cihlar [d], Michael E. Schaepman [a], Glenda García-Santos [a], Richard Fernandes [e], Michael Berger [f]

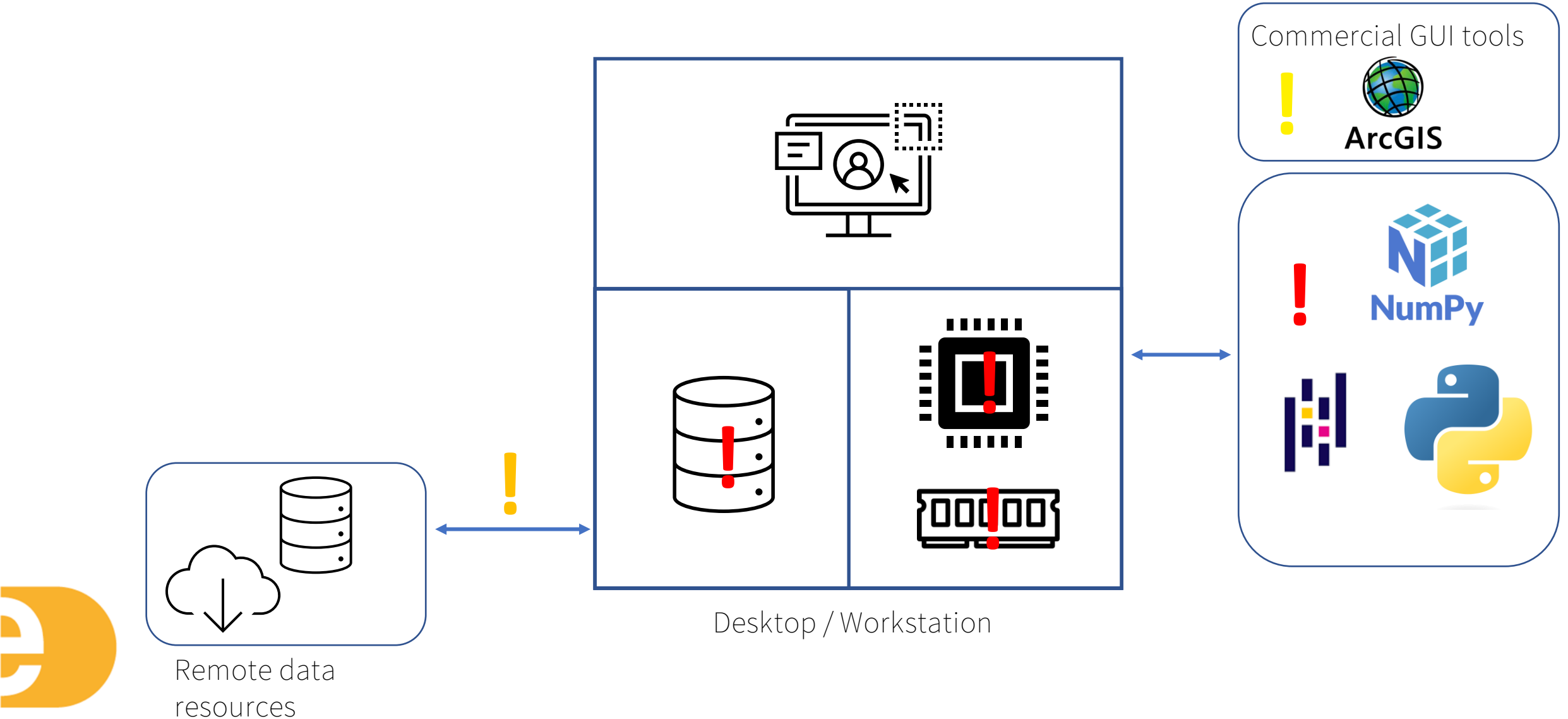


The challenge of Big GeoData

- Petabyte scale, high-resolution data - unique opportunity for scientific exploitation
- Fundamental challenge due to volume – storage & processing
- Established (local) processing solutions do not scale

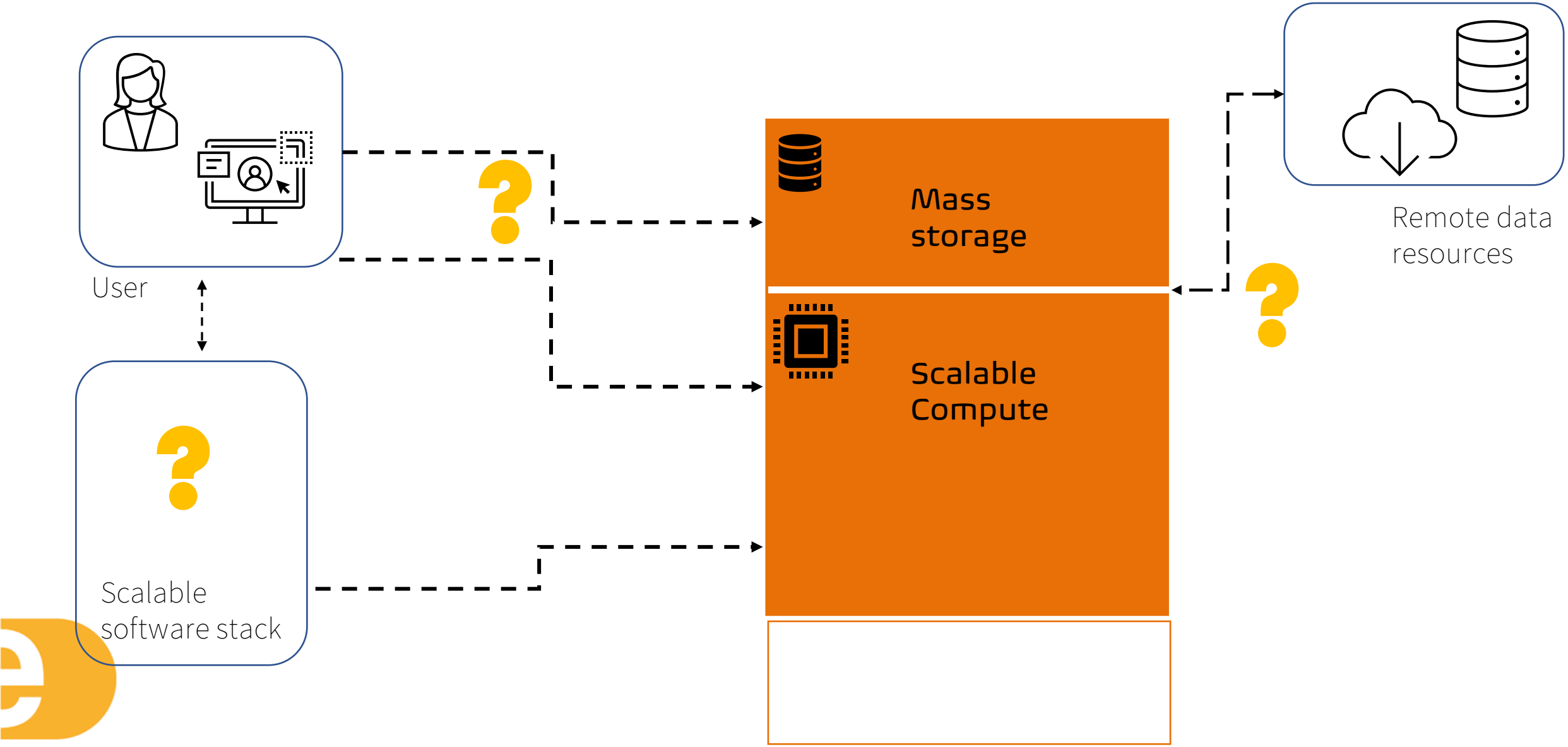
netherlands eScience center

Failure of the status quo



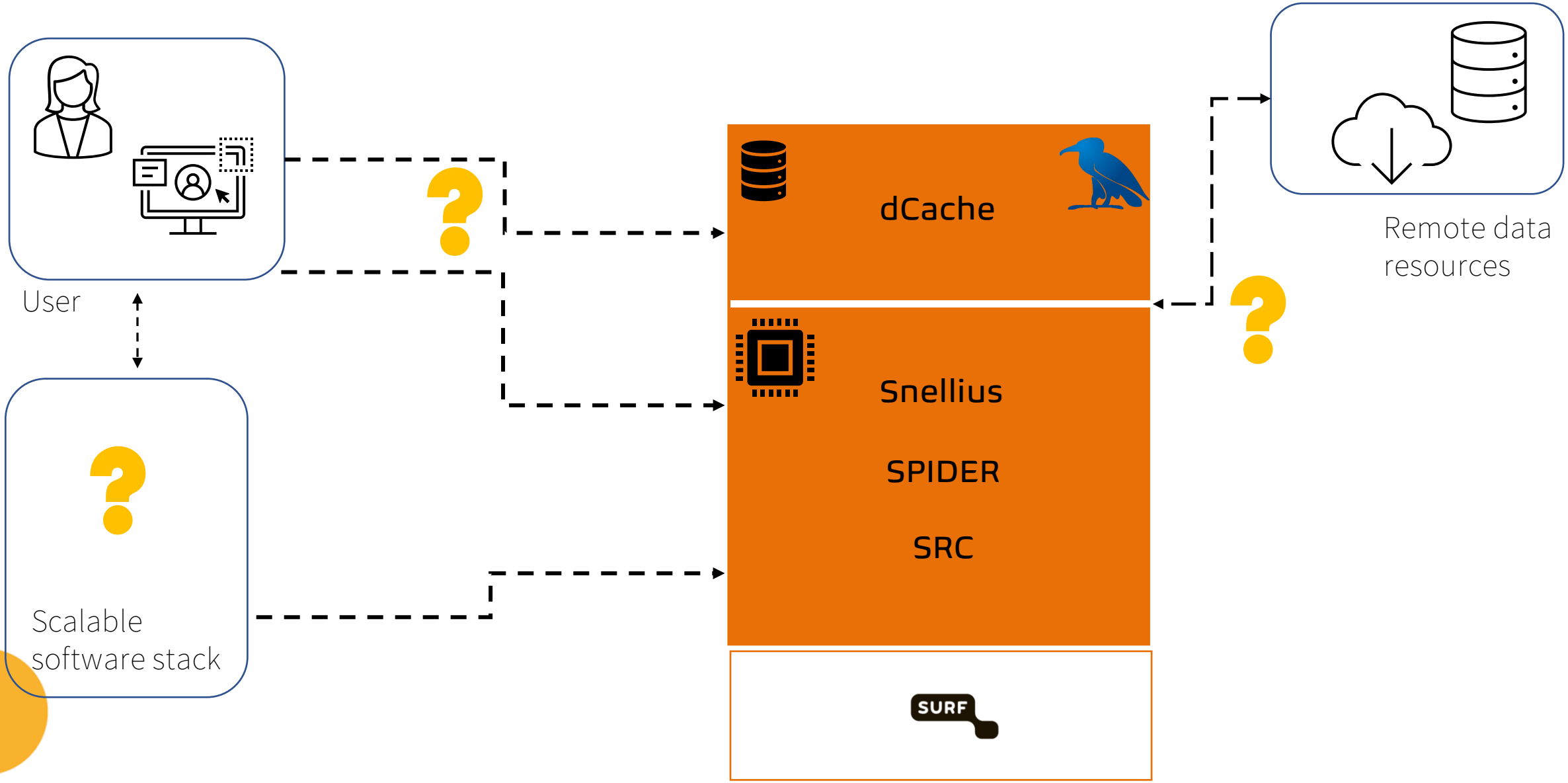
netherlands eScience center

What next?



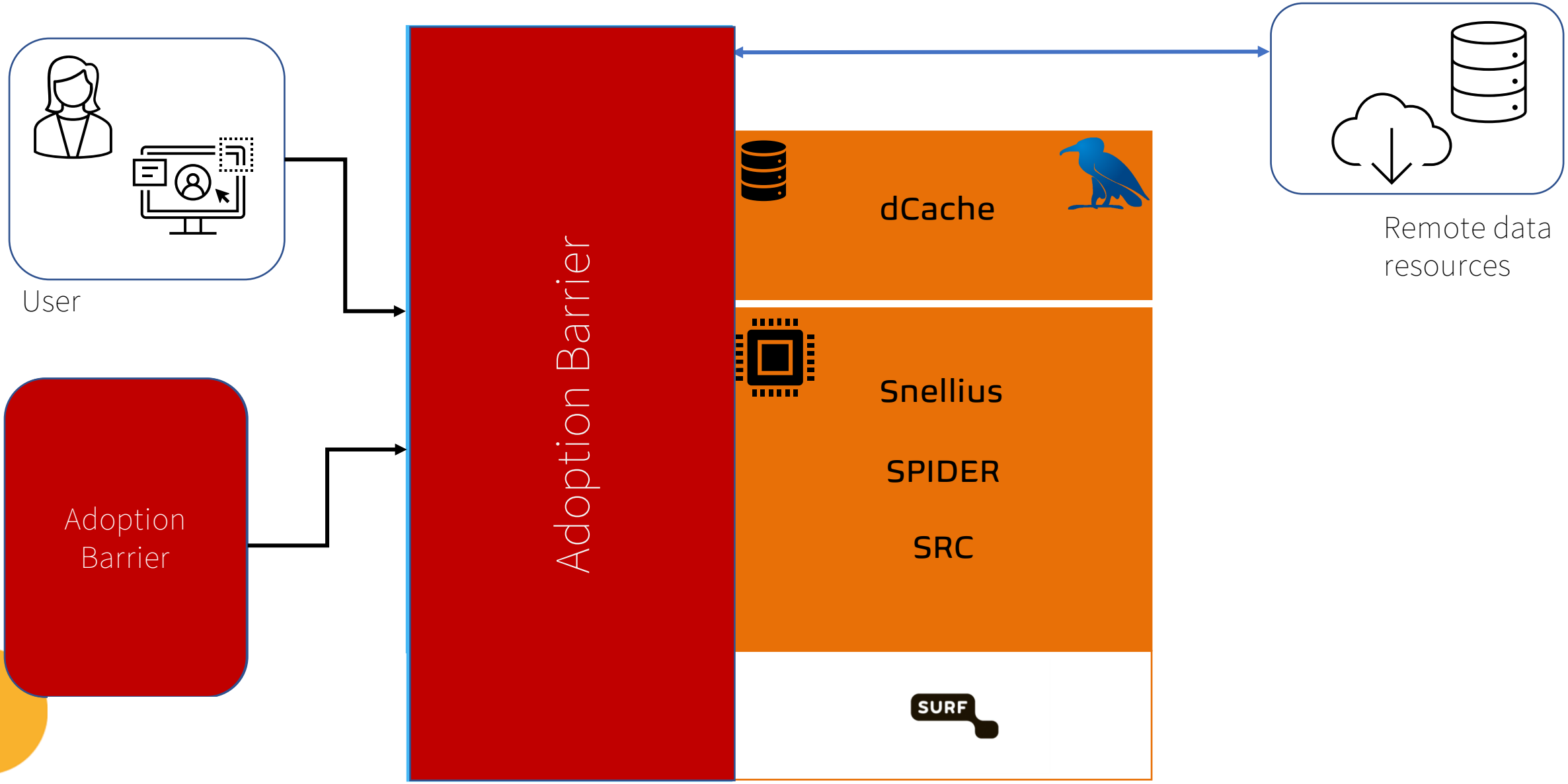
netherlands eScience center

What next?

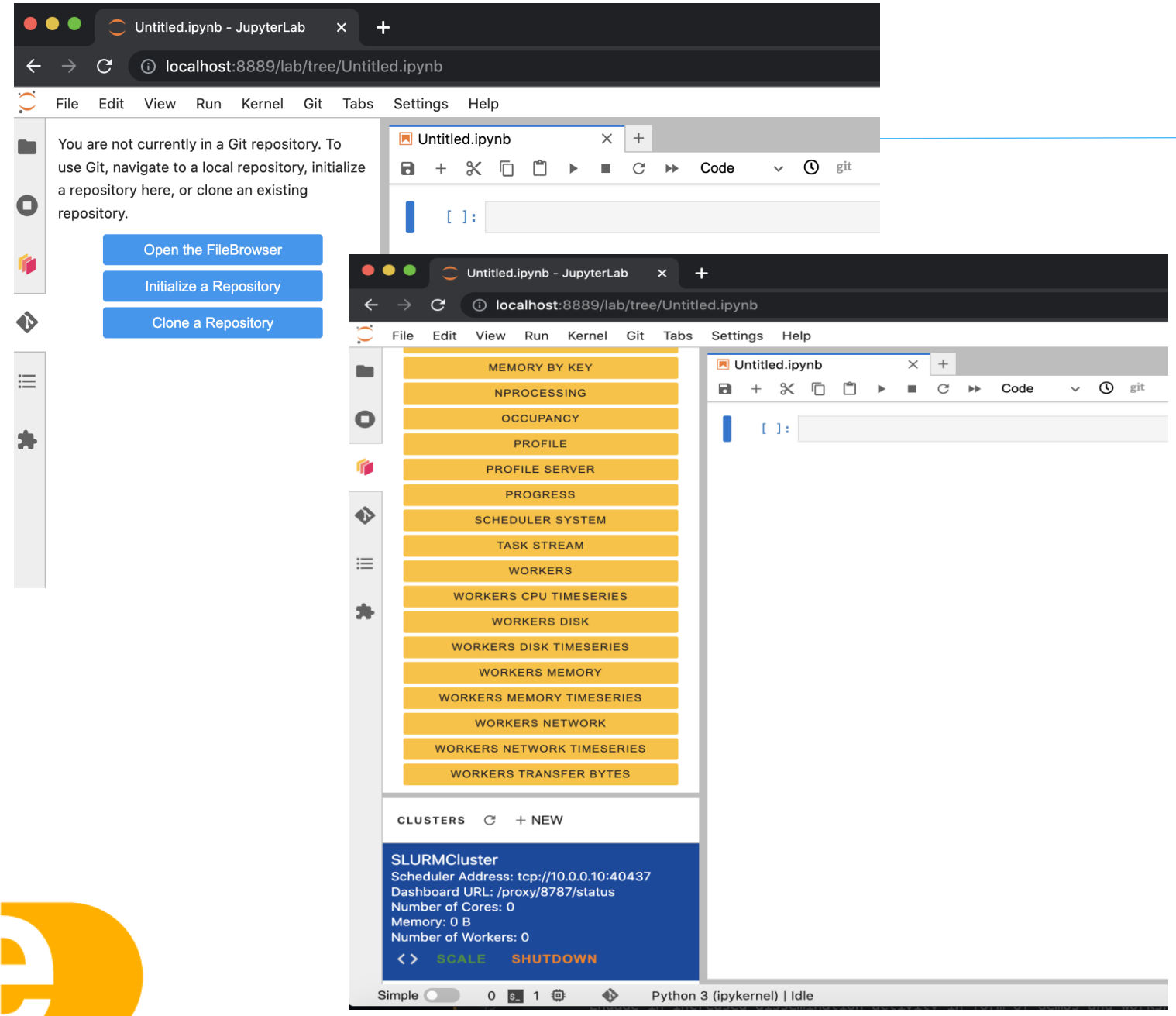


netherlands eScience center

What is RS-DAT ?



Jupyter at scale

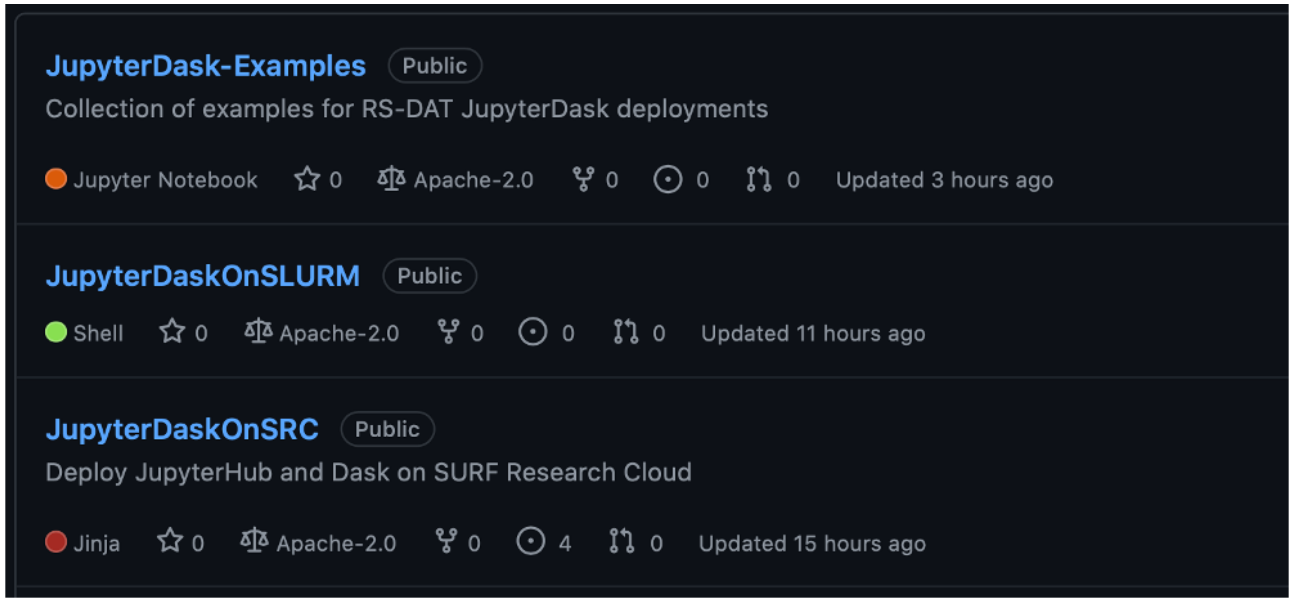


- Scale analysis: accustomed interactive workflows, but backed by HPC/Cloud
- Combine Jupyter server with Dask cluster on HPC/HTC/Cluster/Cloud system

Thanks to team Beta for fruitful discussions and inspiration

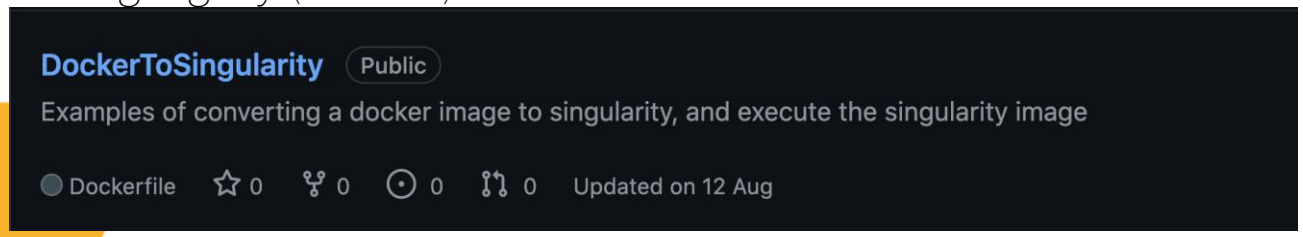
RS-DAT Features

https://github.com/RS-DAT/JupyterDaskOnSLURM



Scalable analysis with Jupyter

Using legacy (Docker) containers for HPC

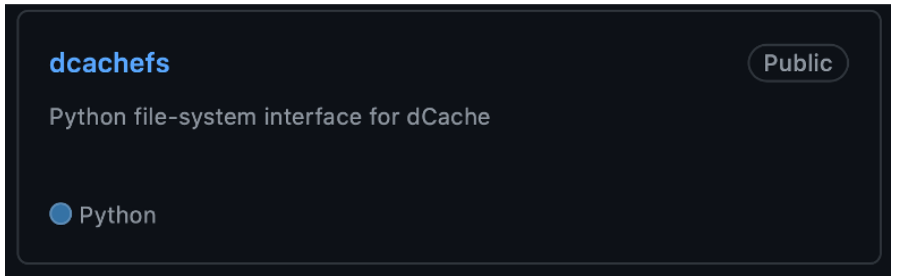


https://github.com/RS-DAT/DockerToSingularity

https://github.com/NLeSC-GO-common-infrastructure/stac2dcache

utility functions to manage STAC catalogs (and the underlying data) on dCache.

https://github.com/NLeSC-GO-common-infrastructure/dcachefs



Python interface to dCache

netherlands eScience center

The PANGEO community

Pangeo is first and foremost a community of people working collaboratively to develop software and infrastructure to enable Big Data geoscience research.

Some of the products produced by this community include interconnected software package and deployments of this software in cloud and high-performance-computing environments. Such a deployment is sometimes referred to as a *Pangeo Environment*.

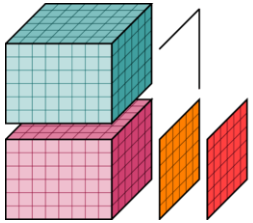
PANGEO

A community platform for Big Data geoscience

Core:
Interoperable, scalable, Python data science stack with geospatial focus

dask

Scalable computation



xarray

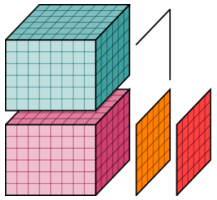
Rich data mode w/ out-of-core support

jupyter

Interactive analysis and execution

netherlands eScience center

RS-DAT core in brief



xarray

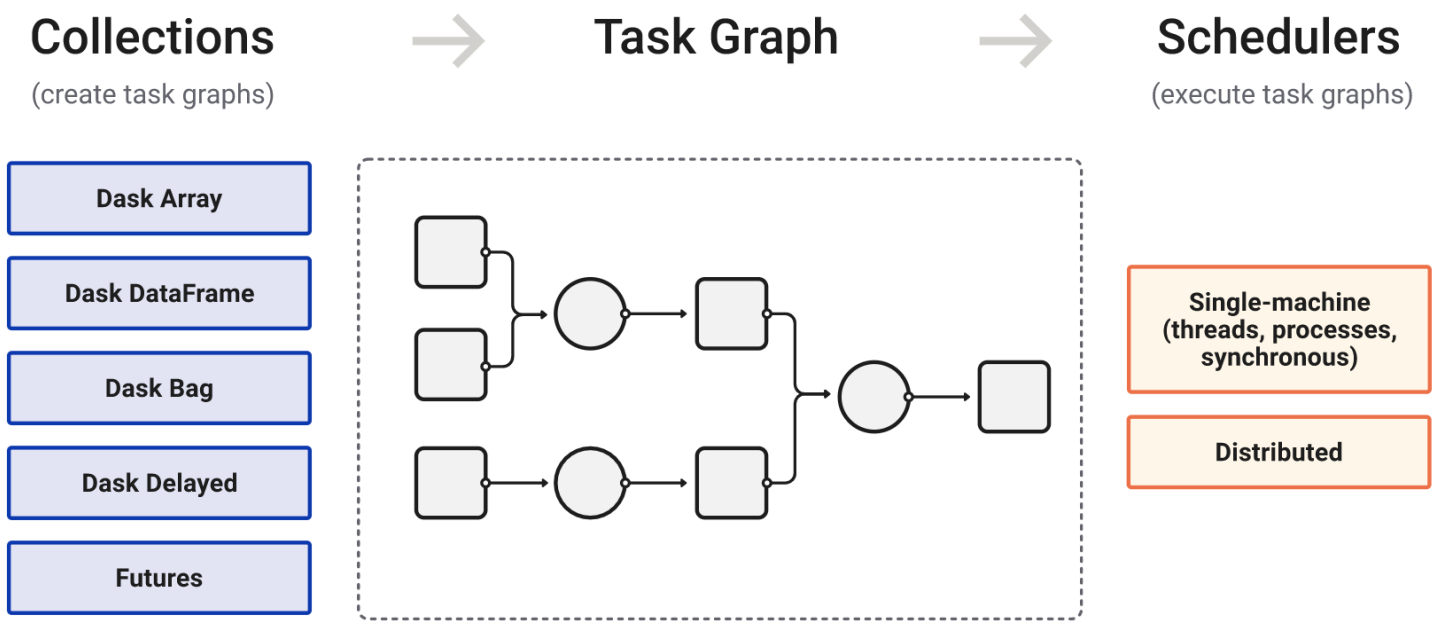
dask

netherlands eScience center

dask

Flexible library for parallel and distributed computing on "big (larger than memory) data".

- Dynamic task scheduling optimized for computation
- Extends pandas and NumPy APIs to larger-than-memory data (chunks)
- Fully customizable
- Scales from laptop to cluster (and back)
- Designed for interactive use
- Native python
- Almost like writing serial code (some idiosyncracies)



High level collections are used to generate task graphs which can be executed by schedulers on a single machine or a cluster.

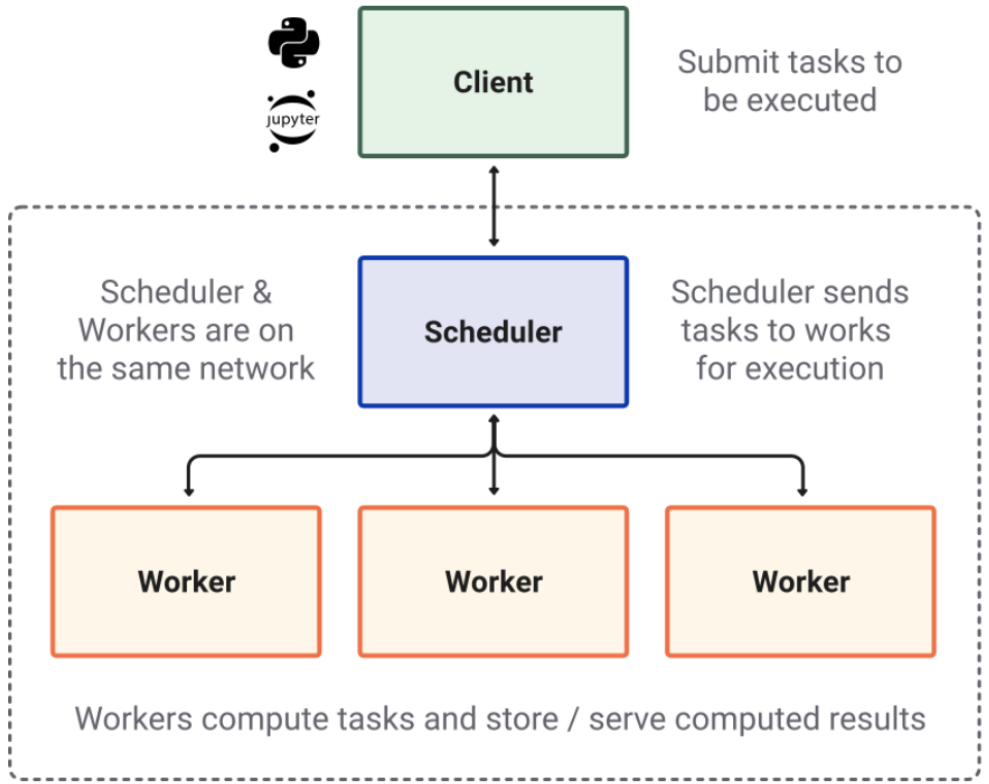
netherlands eScience center

dask

Dask distributed scheduler

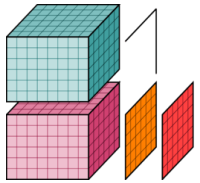
Supports fully asynchronous workflows and provides sophisticated data locality.

Seamlessly integrates with many/most HPC and cloud systems



Dask Cluster

netherlands eScience center



xarray

Support for N-dimensional labelled arrays (extends NumPy and pandas), with rich meta data

Compatible with NetCDF file format

Support for larger-than-memory data sets via chunking and lazy read/execution using dask

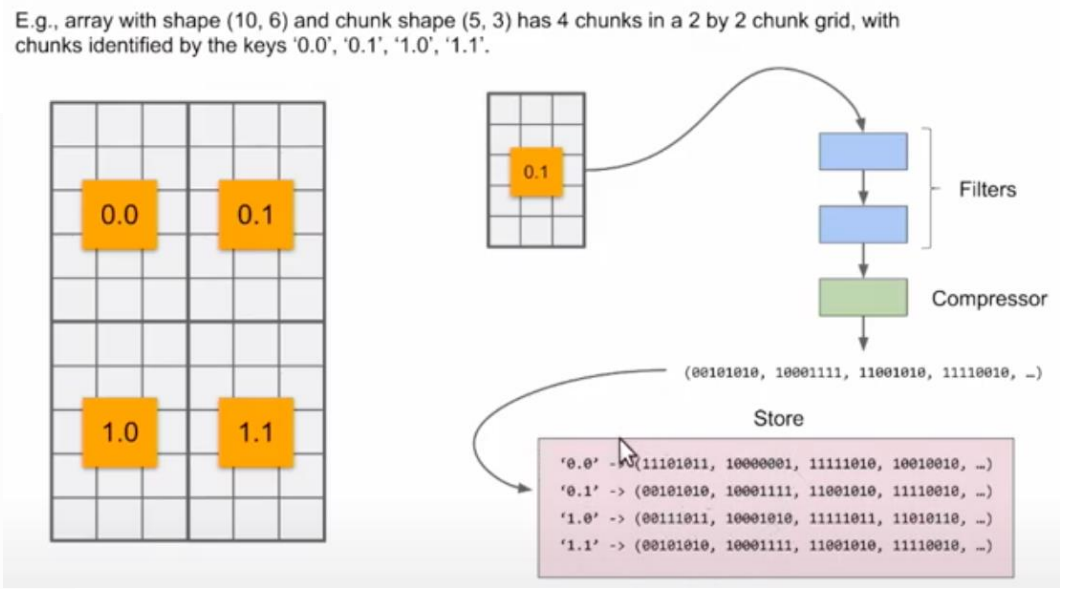
Customizable as data access layer across applications. Extendable with own methods.

netherlands eScience center

Zarr

- “Zarr is a format for the storage of chunked, compressed, N-dimensional arrays”
- Integrates with dask and Xarray

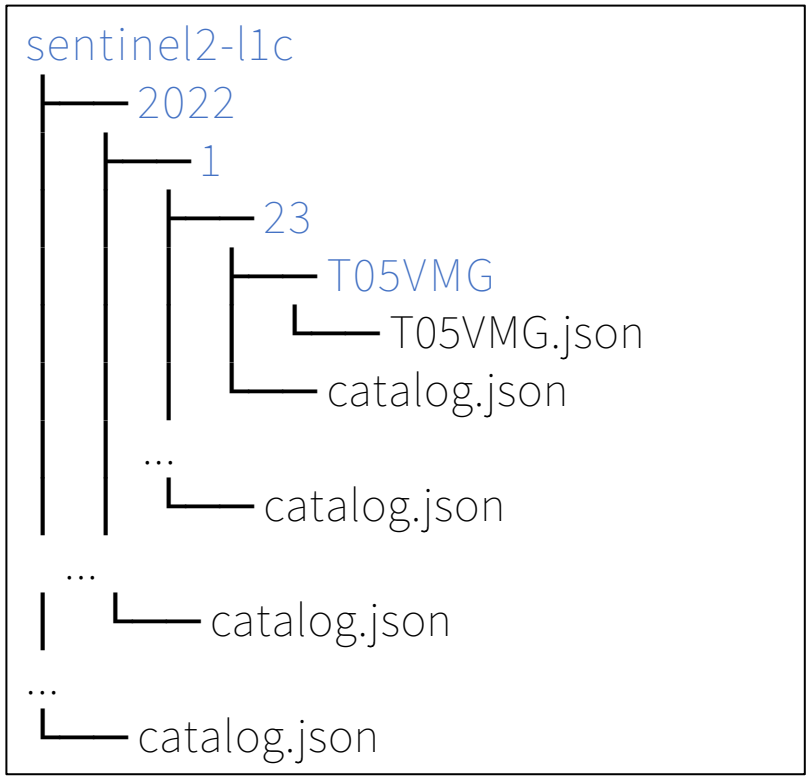
- Create N-dimensional arrays with any NumPy dtype.
- Chunk arrays along any dimension.
- Compress and/or filter chunks using any NumCodecs codec.
- Store arrays in memory, on disk, inside a Zip file, on S3, ...
- Read an array concurrently from multiple threads or processes.
- Write to an array concurrently from multiple threads or processes.
- Organize arrays into hierarchies via groups.



STAC and dCache

Spatio-Temporal Asset Catalogue

- Common structure to describe spatio-temporal data.
- Standard way to catalog geospatial data files.
- All about metadata, linked to data.
- STAC ecosystem includes:
  - Specifications (core elements definitions);
  - API;
  - Tools.



dCache

- Mass storage system supporting heterogenous nodes as single virtual filesystem
- Multiple I/O, incl. HTTP/Webdav

netherlands eScience center

HPC/HTC vs Cloud

HPC/HTC +

- " No cost" , merit based
- Highly flexible and configurable (Hardware)
- Simple environment customization and dissemination
- Well established filesystem support
- Highly optimized I/O infrastructure
- High level of support
- **No dependency on commercial providers**

Cloud +

- No/little need to download data
- Easy configuration via (VM) presets
- Available on demand at requested level
- Possibly available high-level convenience functions (e.g. mosaicing)
- Professional support (if paid for)

HPC/HTC −

- Must retrieve data from remote
- Storage can be problematic (not so for SURF)
- Workflow must be implemented at fairly low level
- Possibility of resource competition
- Delay between first request and access to resources

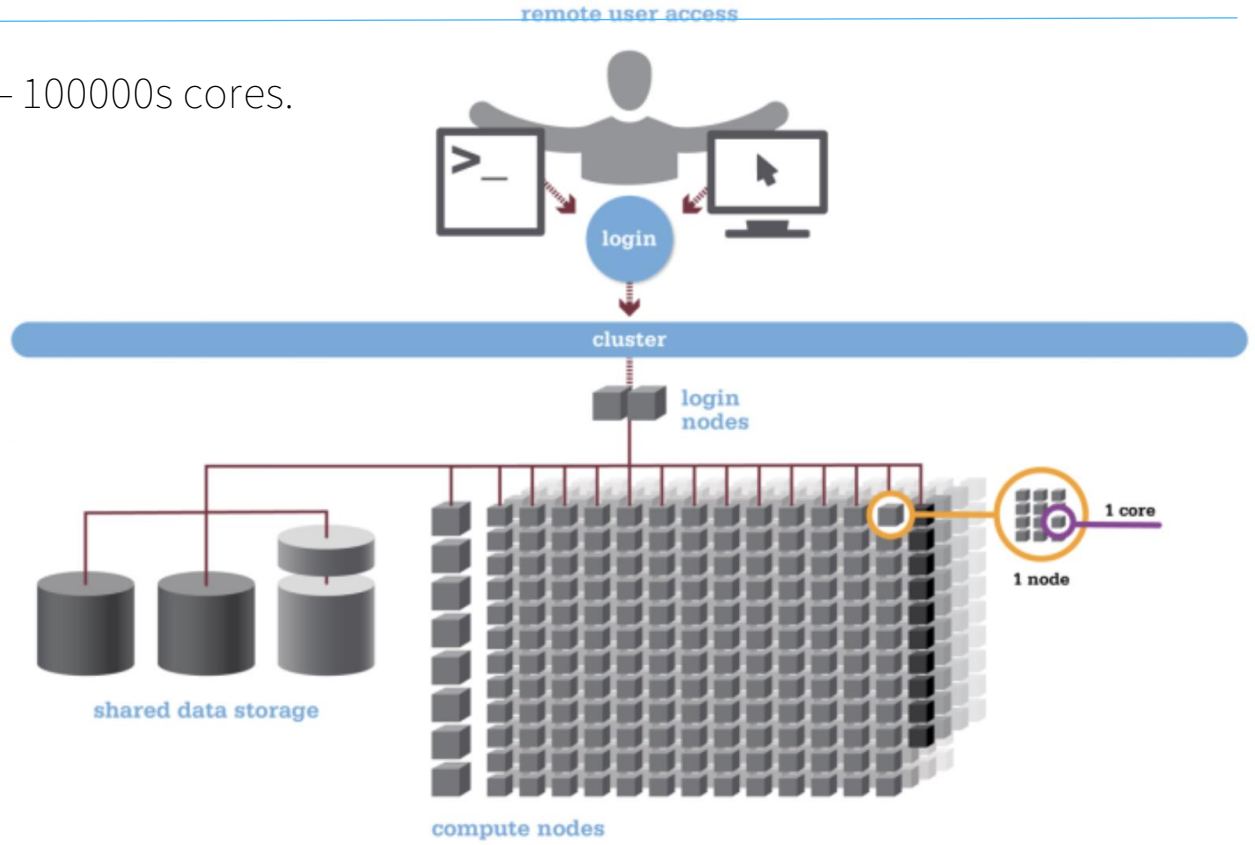
Cloud −

- Commerical provider; paid access
- Limited access (on free research allotments)
- Intricate config for non-standard requirements
- Support (only) at cost
- Scaling can lead to high costs

netherlands eScience center

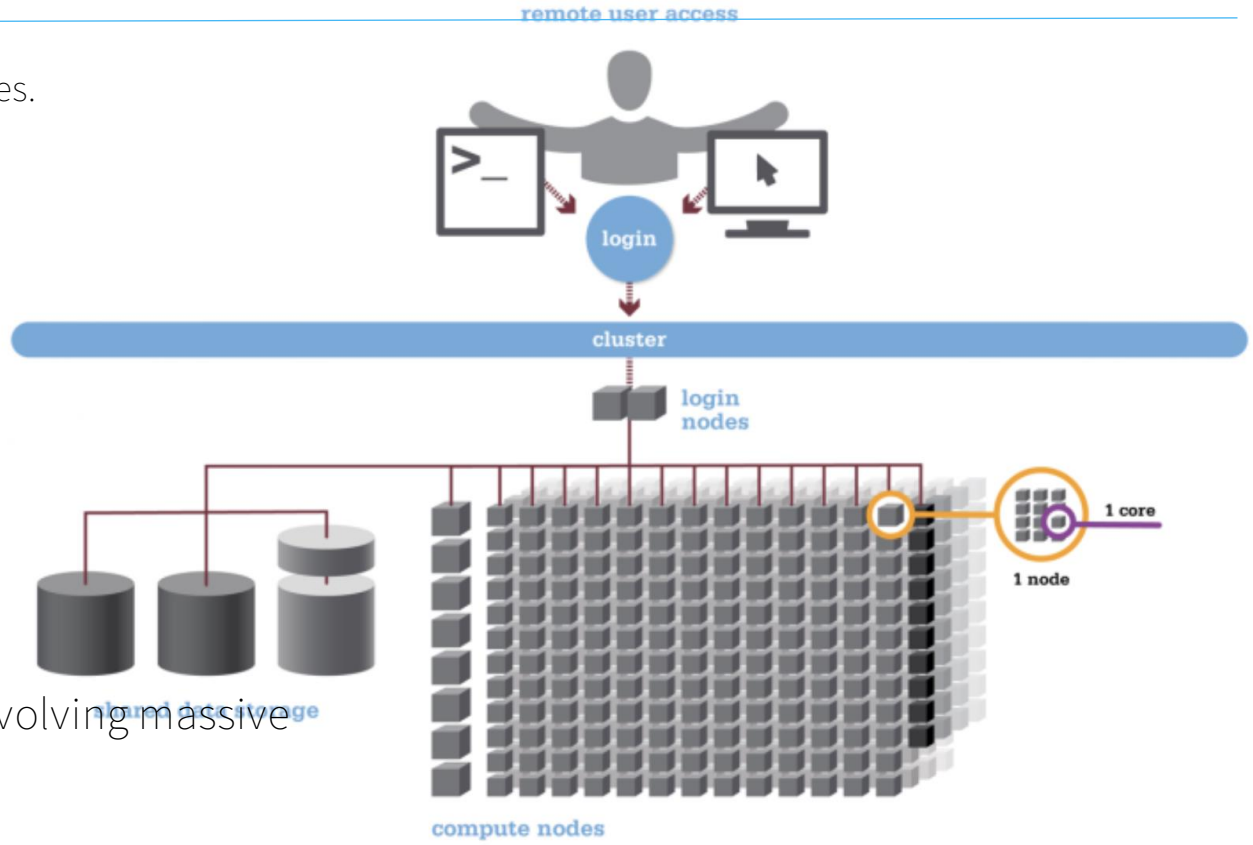
HPC/HTC basics

- Clusters of compute servers (100s –1000s nodes), with 10000s – 100000s cores.
- High bandwidth internal network between nodes
- Shared RAM per node; configurable
- Shared storage/mounted FS
- Generally Linux based (command line, SSH keys)



HPC/HTC basics

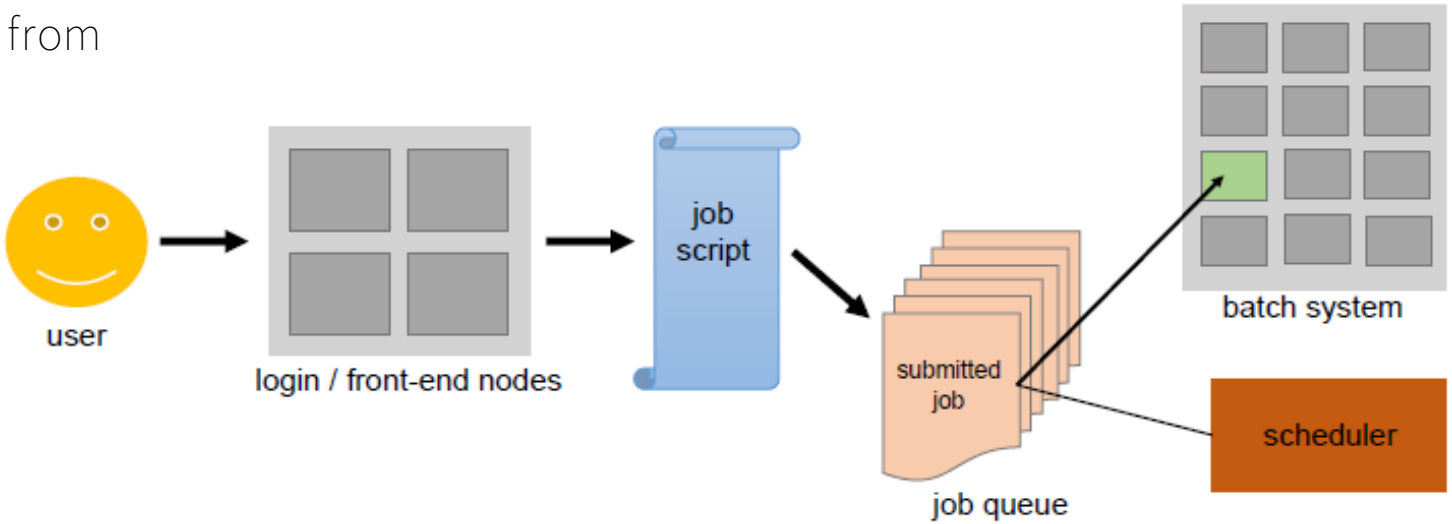
- Clusters of compute servers (100s –1000s nodes), with 10000s – 100000s cores.
- High bandwidth internal network between nodes
- Shared RAM per node; configurable
- Shared storage/mounted FS
- Generally Linux based (command line, SSH keys)



- Designed for non-interactive, large jobs or batch jobs
- Either VERY computationally expensive (classic HPC) and/or involving massive data volumes tto process (HTC)
- HPC systems are a SHARED resource.
(login, etiquette)
- Resources must be allocated and jobs scheduled

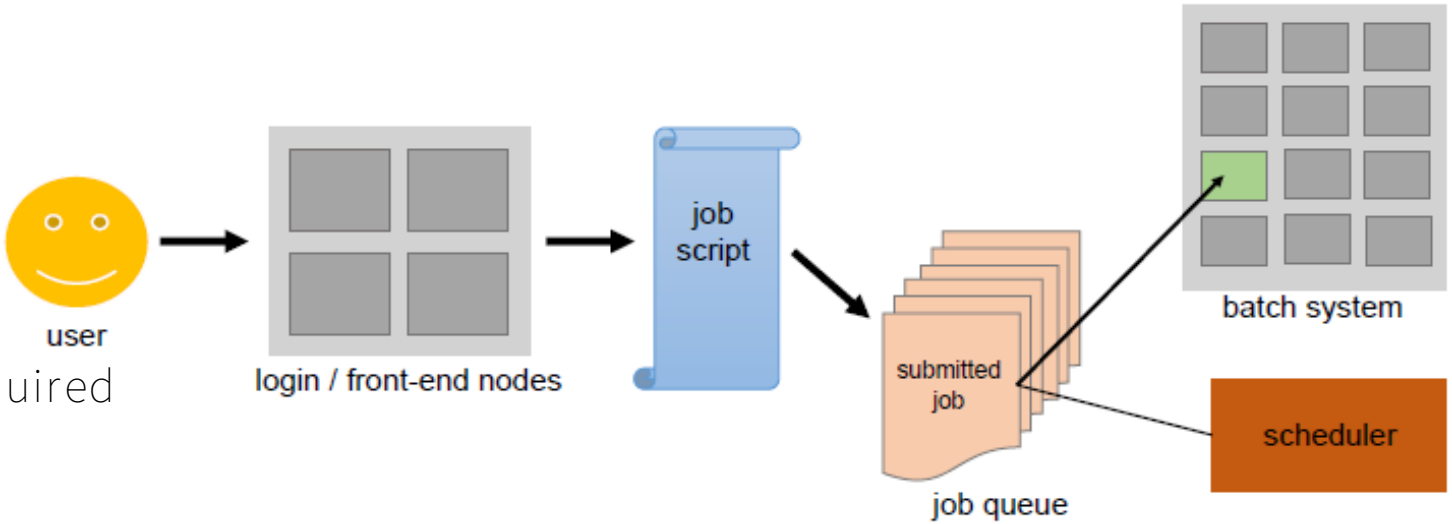
scheduling

- HPC is shared system running multiple jobs from users.
- Need system/software to handle resource allocation
  The batch scheduler



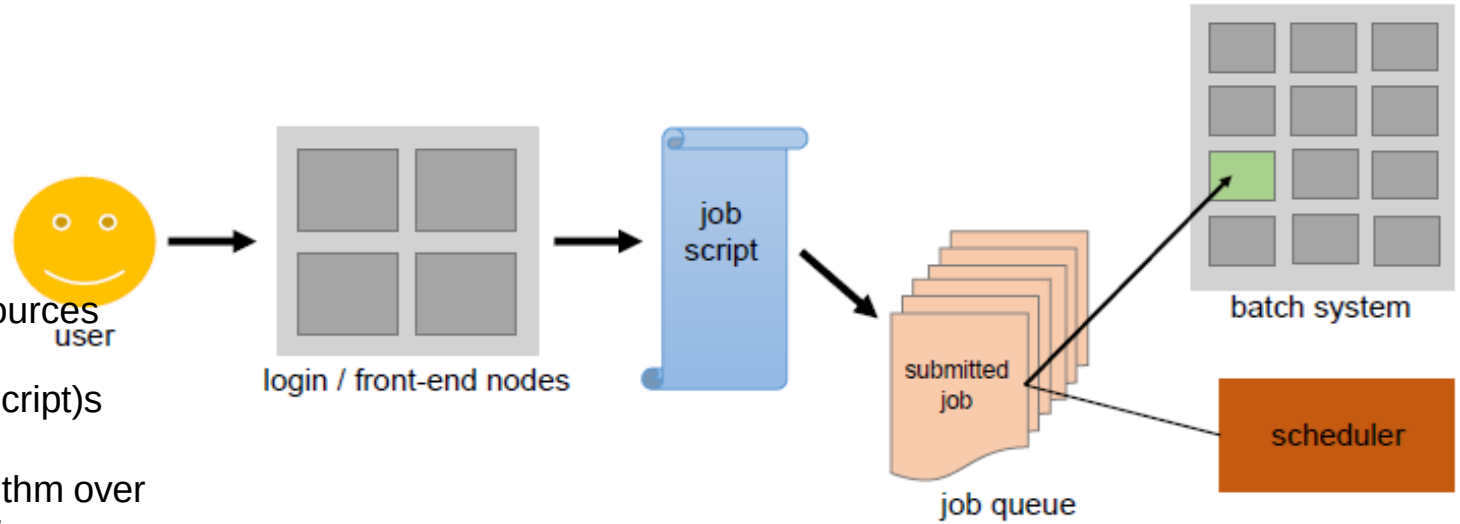
scheduling

- HPC is shared system running multiple jobs from users.
- Need system/software to handle resource allocation
  The batch scheduler
- Elements of scheduling:
  - Job script: declaration of job and required resources
  - Job queue: ordered registry of submitted job(script)s
  - Scheduler: Software running scheduling algorithm over job queue and allocating resources and priority



scheduling

- HPC is shared system running multiple jobs from users.
- Need system/software to handle resource allocation
  The batch scheduler
- Elements of scheduling:
  - Job script: declaration of job and required resources
  - Job queue: ordered registry of submitted job(script)s
  - Scheduler: Software running scheduling algorithm over job queue and allocating resources and priorities

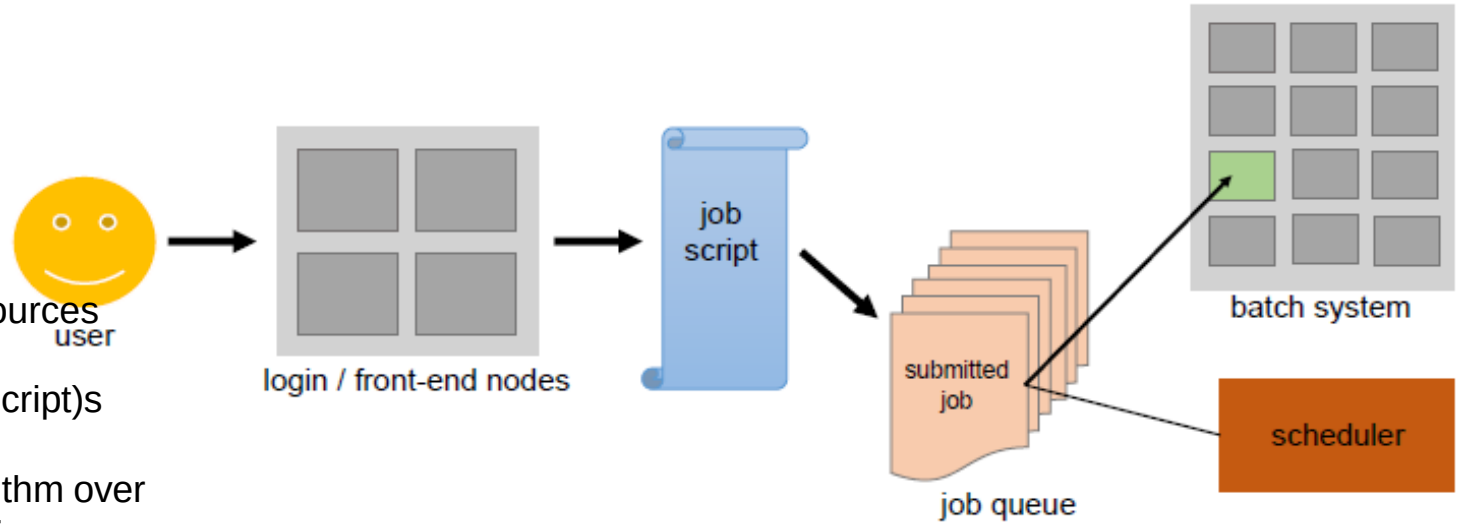


- Scheduling algorithms:
  - First come, first served
    Everything scheduled, but inefficient
  - Shortest job first
    Long/resource heavy may never be scheduled

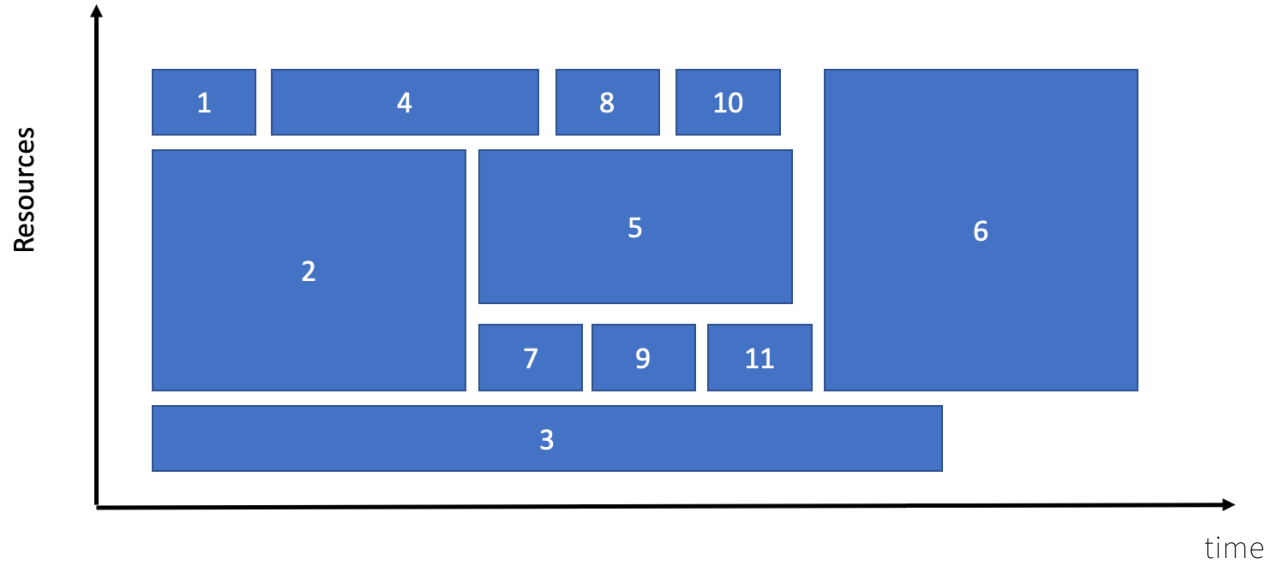
netherlands eScience center

scheduling

- HPC is shared system running multiple jobs from users.
- Need system/software to handle resource allocation
  The batch scheduler
- Elements of scheduling:
  - Job script: declaration of job and required resources
  - Job queue: ordered registry of submitted job(script)s
  - Scheduler: Software running scheduling algorithm over job queue and allocating resources and priorities



- Scheduling algorithms:
  - First come, first served
    Everything scheduled, but inefficient, long wait
  - Shortest job first
    Long/resource heavy may never be scheduled
  - Backfilling
    - Combine and interleave, all scheduled



scheduling

- SLURM scheduler on SURF systems (Spider, Snellius)
- SLURM at a glance https://slurm.schedmd.com/quickstart.html

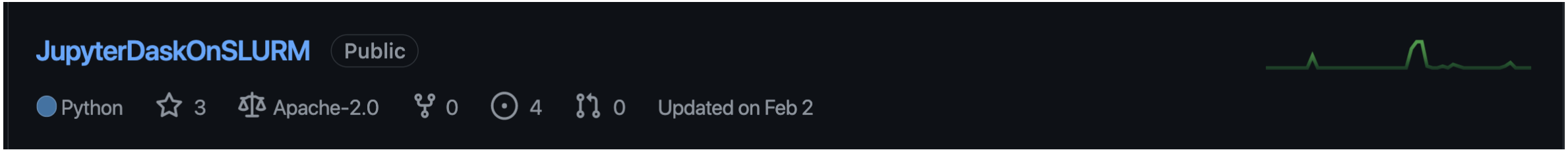
slurm workload manager

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=24:00:00
#SBATCH --cpus-per-task=2
#SBATCH --partition=normal
#SBATCH --reservation=rsdat_course

source ~/.bashrc
conda activate jupyter_dask

node=`hostname -s`
port=`shuf -i 8400-9400 -n 1`
if [ -z ${lport:+x} ]; then lport="8889" ; else lport=${lport}; fi

echo "Run the following on your local machine: "
echo "ssh -i /path/to/private/ssh/key -N -L ${lport}:${node}:${port} ${USER}@spider.surf.nl"

jupyter lab --no-browser --port=${port} --ip=${node}
```

netherlands eScience center

Jupyter Dask on Slurm

SURF H*C Infrastructure: Spider (HTC), Snellius (HPC); SLURM scheduler



- JupyterLab instance with Dask and Git extensions, scalable Dask cluster, running on SLURM managed HTC/HPC system
- Basic idea
  - Launch JupyterLab server and Dask scheduler as long-running batch job
  - SSH port-forwarding to connect to JupyterLab server
  - Launch workers as short-lived batch jobs (fast thru queue)