PUNCH use case: Compute4PUNCH for data-intensive analysis applications

Matthias Hoeft

Manuel Giffels, Andreas Henkel, Dominik Schwarz,
Kilian Schwarz, Christoph Wissing

for PUNCH4NFDI Task Area 2

Examples for scientific questions in the PUNCH sciences

Are the laws of physics valid in **extreme environments**?

What happens when **stars explode**?

What forms the **dark matter**?

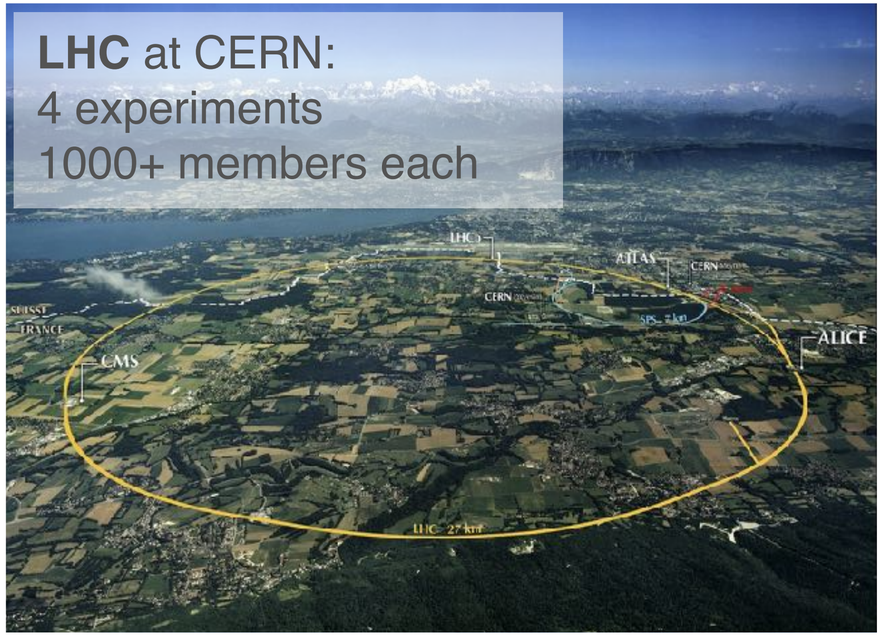
Are we **alone in the universe**?

What is behind the **slight asymmetry** ?

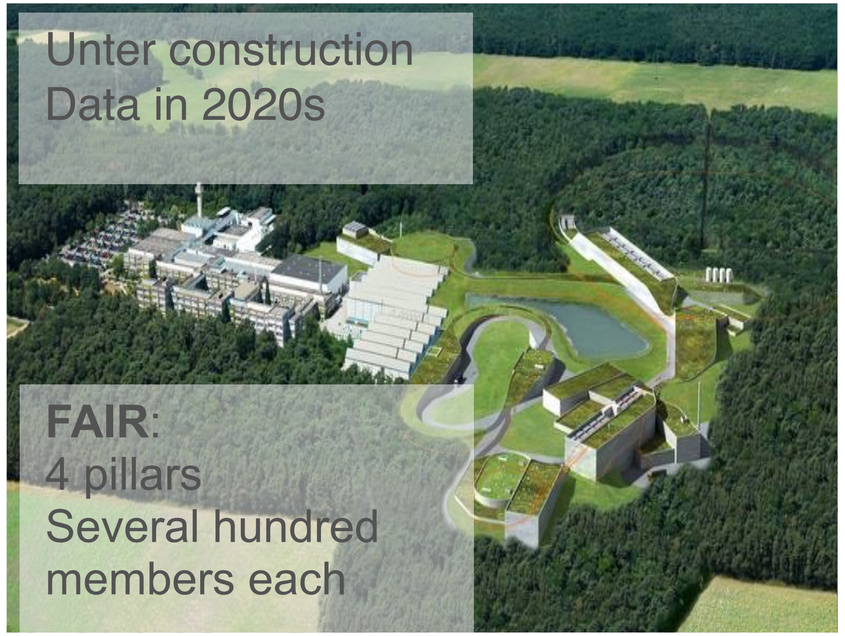
How and when did **stars and galaxies form** in the universe?

How does **black holes** form and evolve ?

Answering these questions needs large facilities

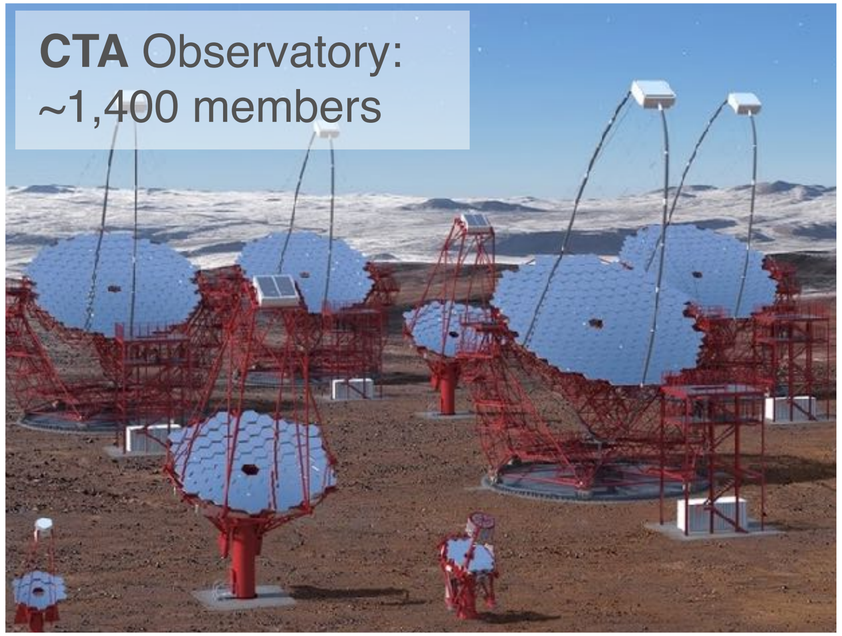


LHC at CERN:
4 experiments
1000+ members each



Unter construction
Data in 2020s

FAIR:
4 pillars
Several hundred
members each



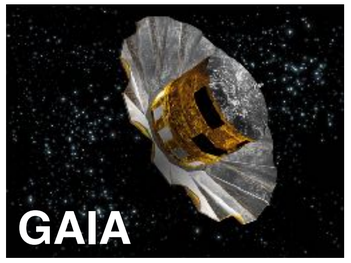
CTA Observatory:
~1,400 members

MeerKAT (South Africa):
Pathfinder for Square
Kilometre Array (SKA)

Supercomputing:
Theoretical physics
Cosmology

JUWELS
at FZ Jülich

And many more
examples



(Some) Challenges

PUNCH communities deal with increasingly large datasets

for instance:

High-Luminosity LHC and the Square Kilometre Array

Advanced studies often require
combination of very different observations, experiments and simulations

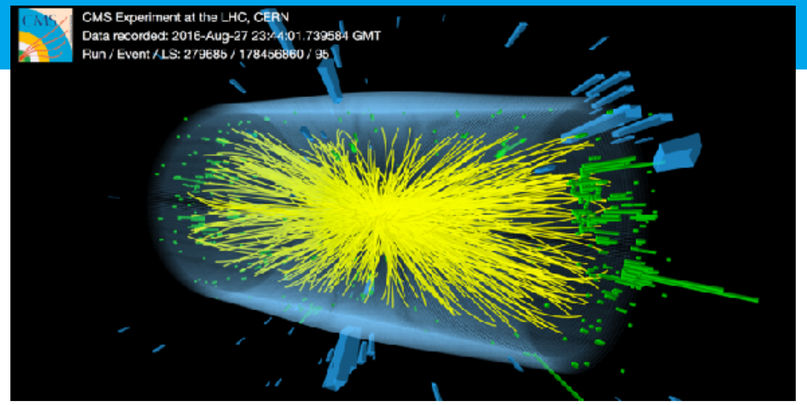
for instance:

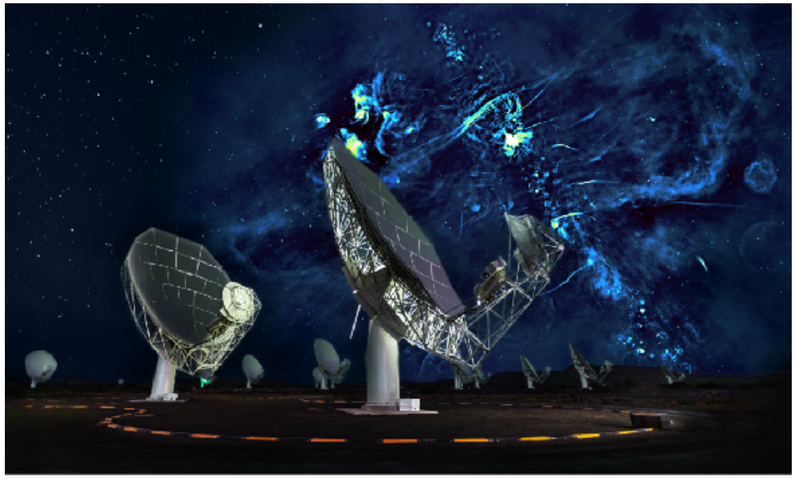
constraining dark matter properties from
astronomical observations and particle experiments

Peak usage of resources

for instance:

reprocessing of archival data with improved analysis scheme





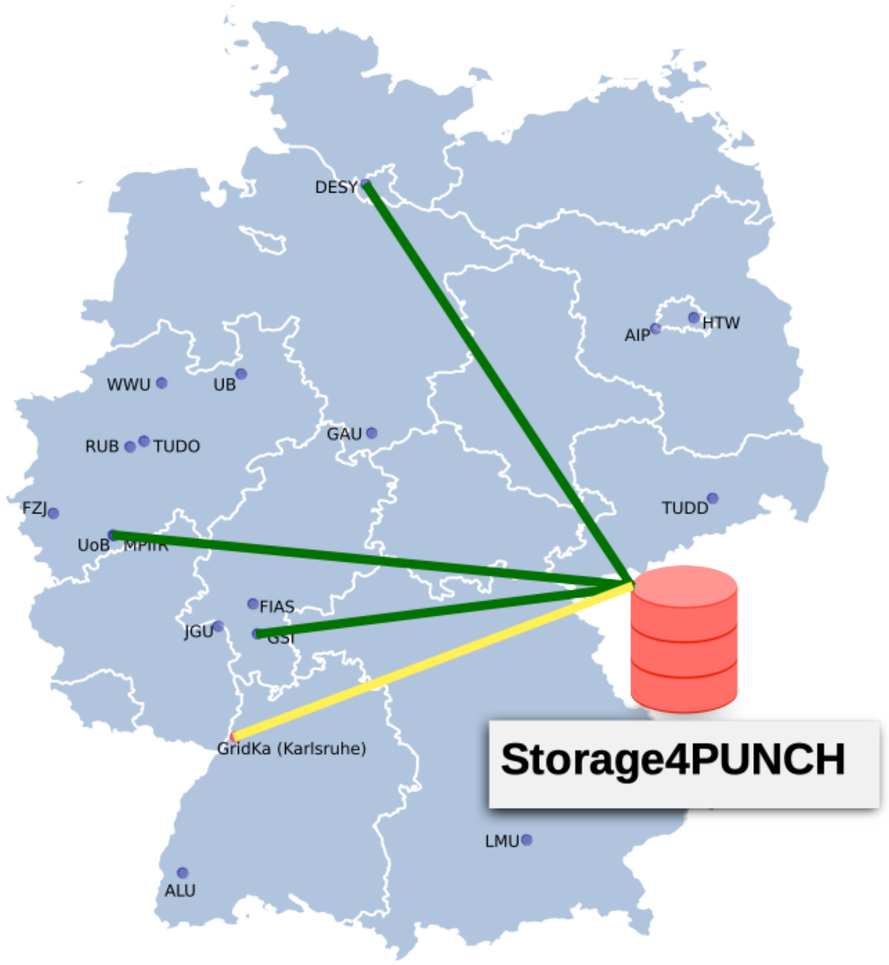
Our Goals

- Federate existing storage and compute infrastructure
- Provide ‘easy-to-use’ entry points to storage and computing resources
- Provide an easy access to a large variety of data collections

Demonstrator Workflows

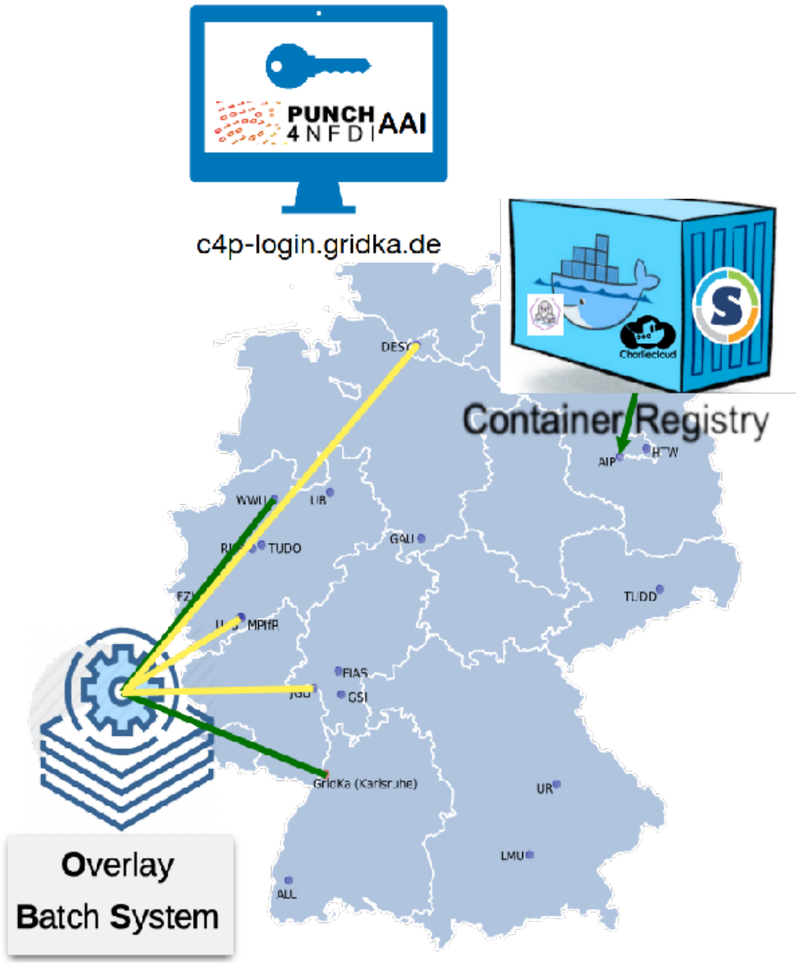
Storage4PUNCH Prototype

- Based on two storage technologies
  - dCache (Instance at DESY & KIT)
  - XrootD (Test instances at U Bonn & GSI)
- Token based access using PUNCH AAI
- Supported protocols: WebDAV & XrootD
- Further integration & evaluation:
  - Systems for file & replica catalogs
  - Rucio – Common DM tool in HEP



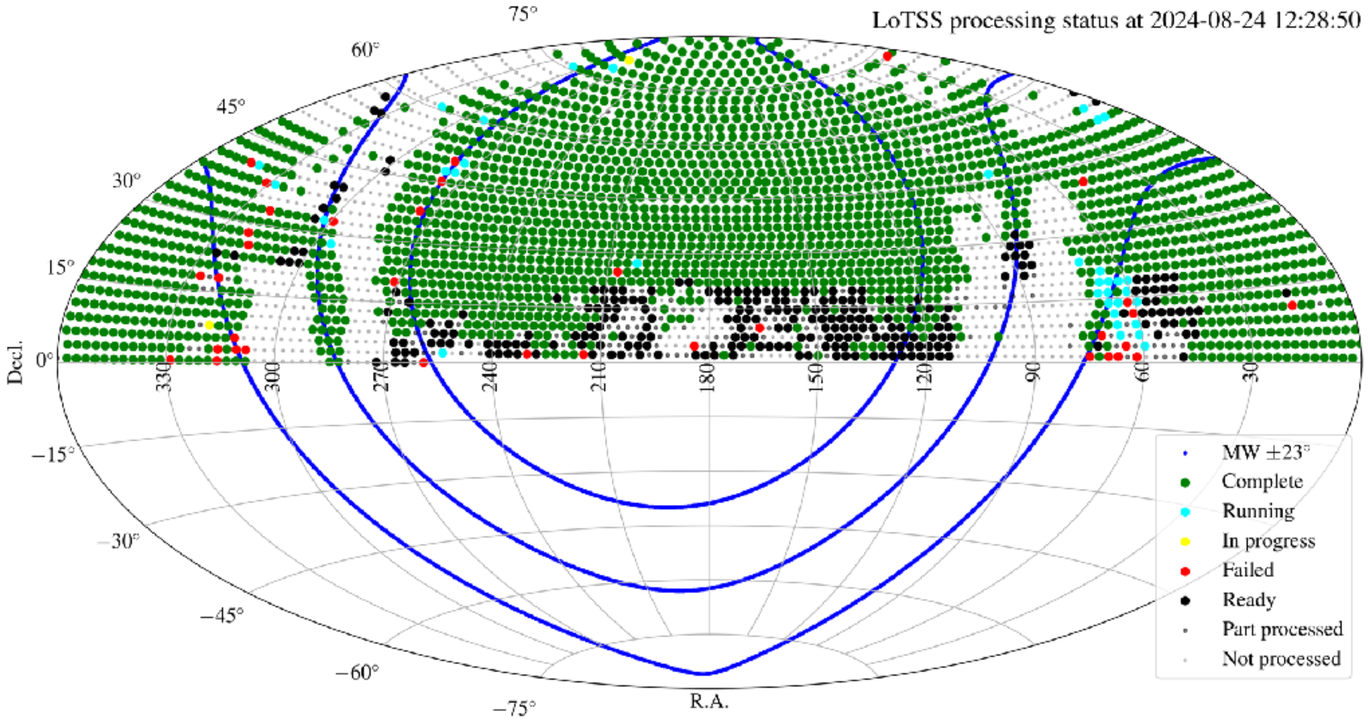
Compute4PUNCH Prototype

- Prototype of federated Compute4PUNCH infrastructure is set up
- AAI based login node of all PUNCH4NFDI members
- Container registry & Gitlab CI/CD workflows
- Dynamic integration of several sites
  U Bonn, KIT, LMU, DESY, FZJ, U Mainz, U Göttingen
- Container distribution via CVMFS
  (CERN Virtual Machine File System)

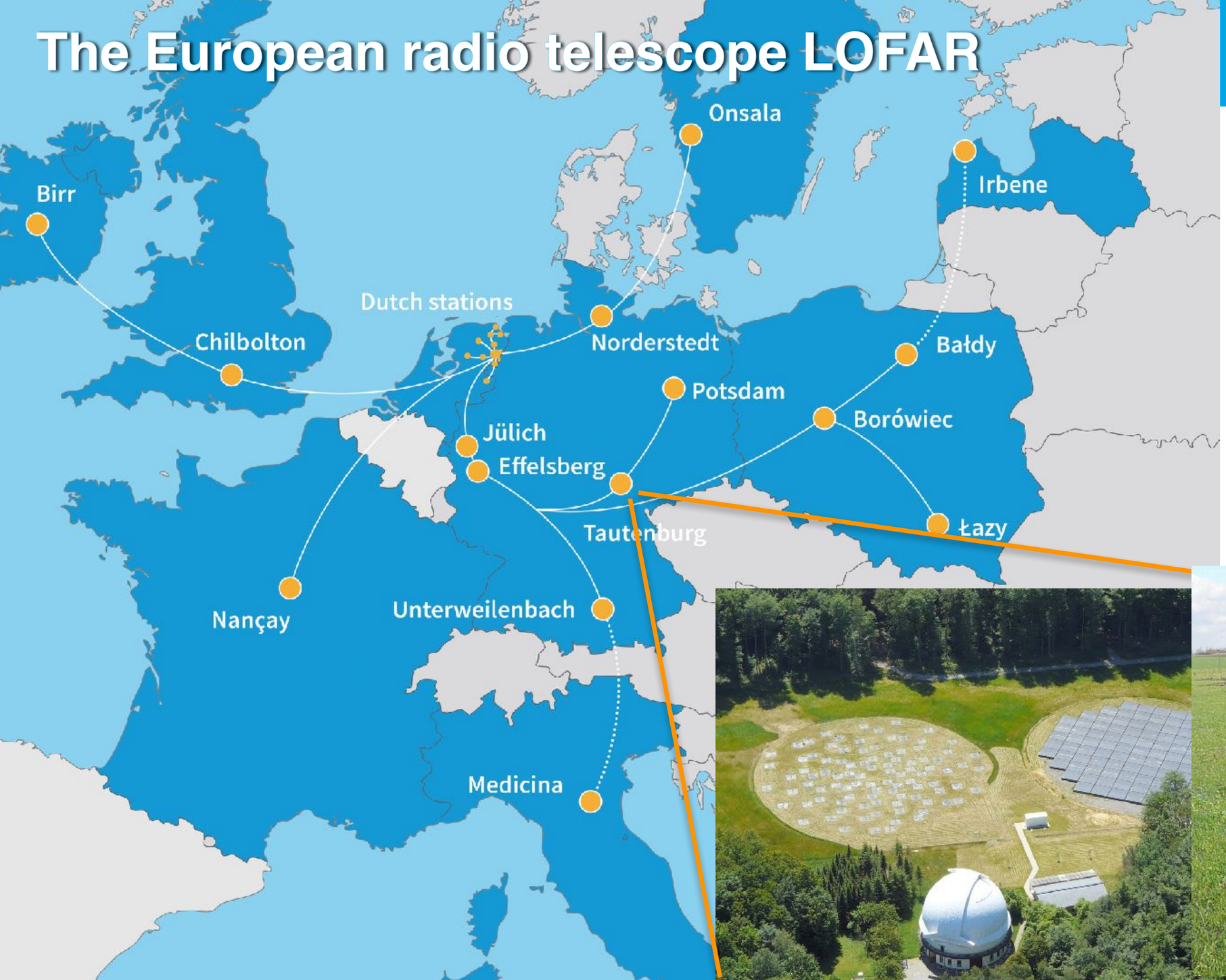


Workflow 1: LoTSS image processing

- LoTSS: Multi-year observing campaign to image the northern sky at very low radio frequencies
- About 1500 ‘observations’ (8hrs pointing)
- Data taken with the Low Frequency Array (LOFAR) imaging requires substantial computing (several mio core-h / year)
- Our aim: Demonstrate that Compute4PUNCH may serve as ‘science data processor’
- Pipeline available as container



The European radio telescope LOFAR



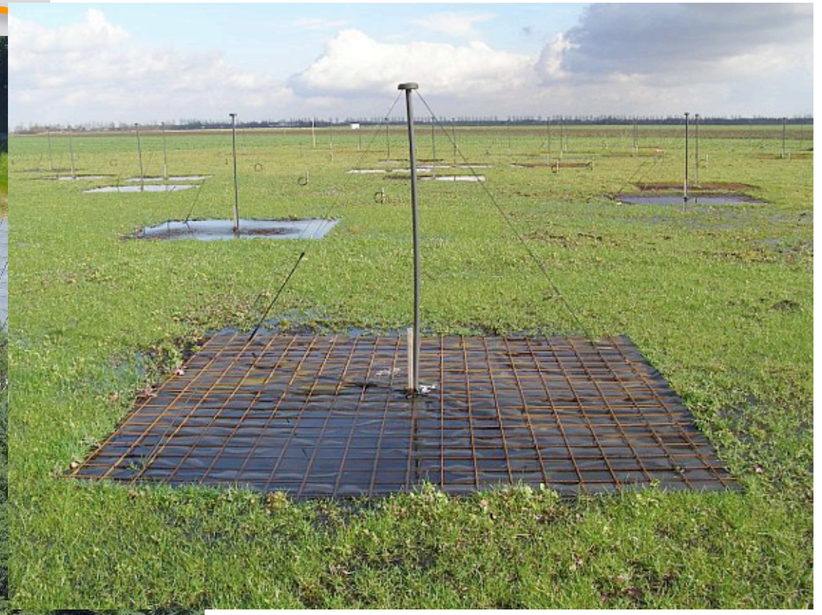
~ 50 stations in Europe

Correlator in Groningen (NL)
'Online' processing

connected via fast data links
data rate at station: 4 Gb/s

100,000 very simple antennas

currently ***upgrade to LOFAR2.0***



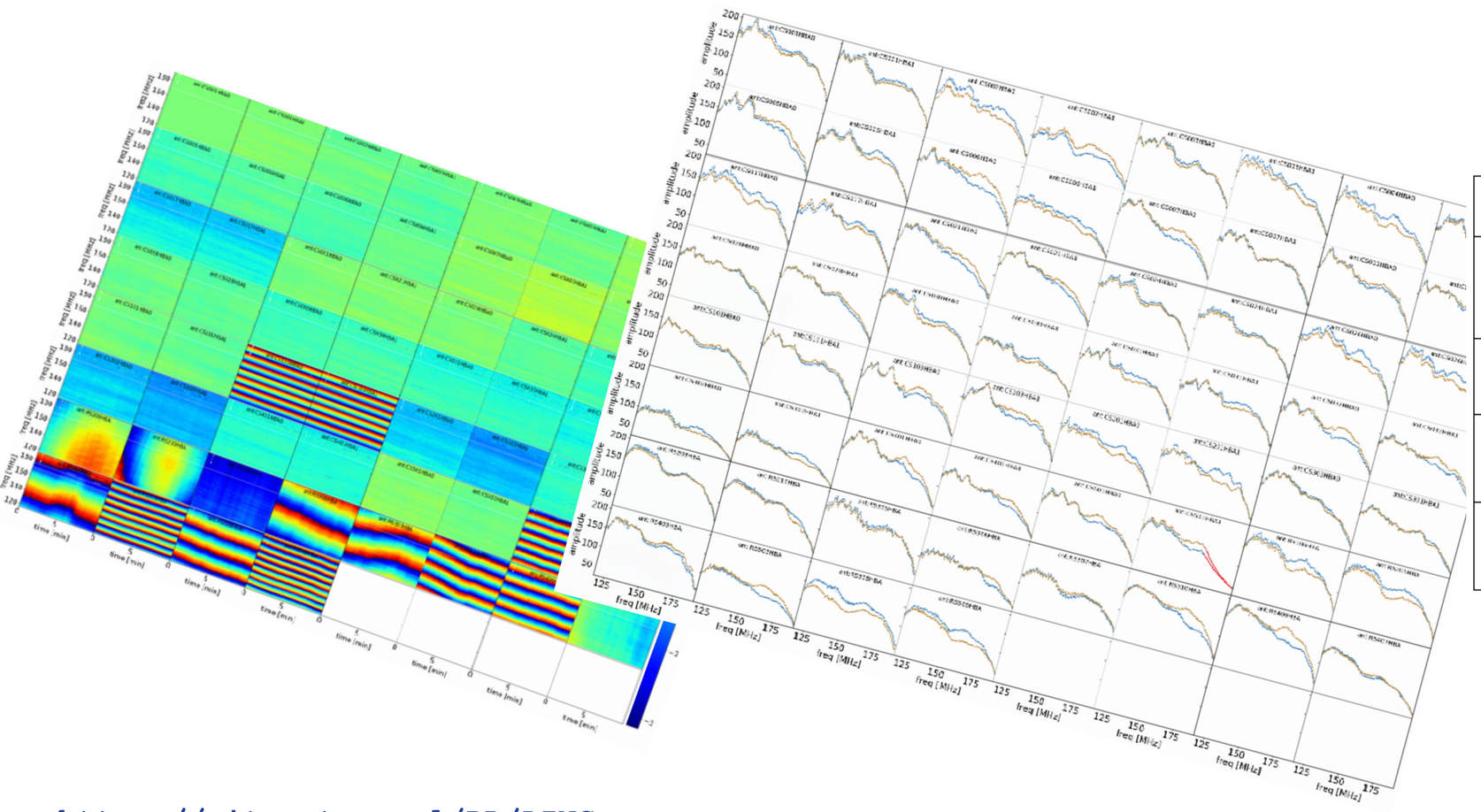
Special requirements for this workflow

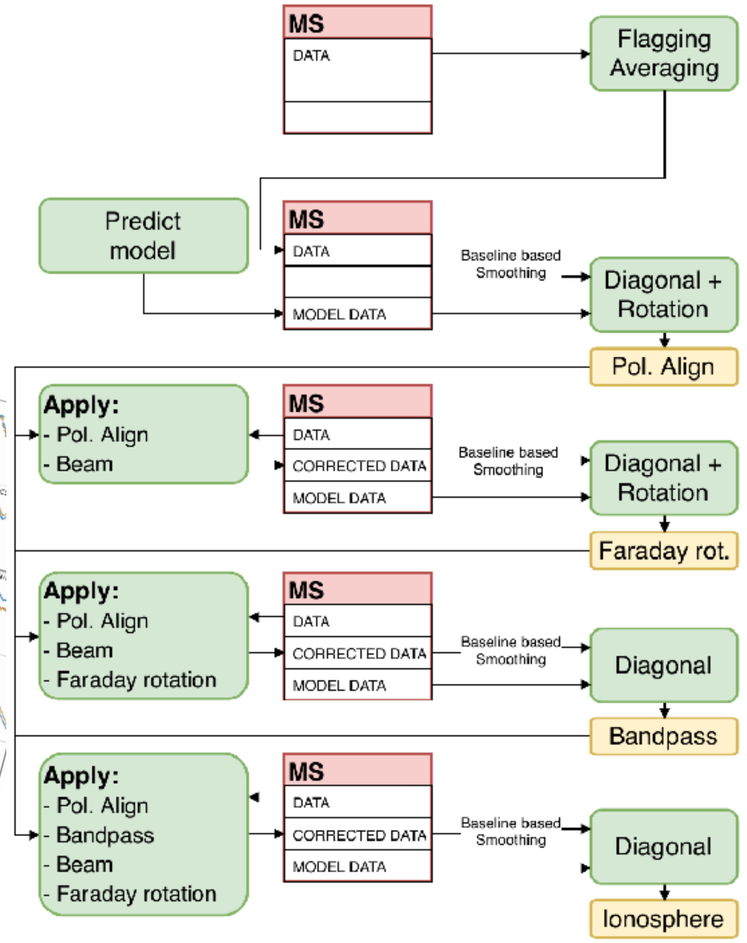
- Data are stored on tapes at FZ Jülich (long-term archive, also other sites)
- Data volume of one observation ~ 4 - 16 TB
- processing of one observations typically once or a few times
- current software design requires powerful (single) nodes
  and sufficient scratch space (~ 5 … 20 TB)
  processing a few 10000 core-h

Radio Sky Image from LOFAR

Workflow 1: Radio interferometry analysis workflow

- ‘Historically’ grown software suite
- CWL pipeline



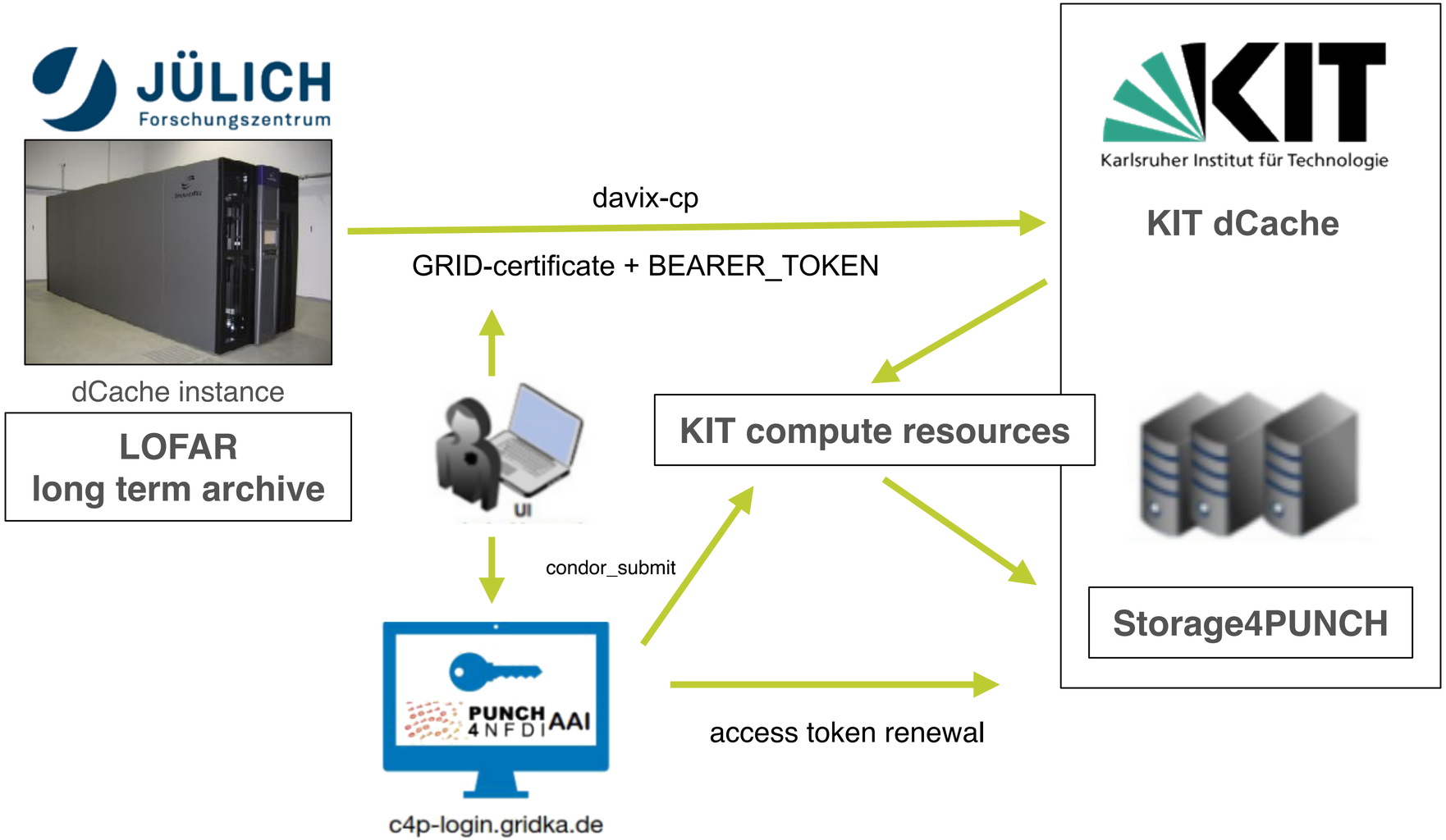


Slide: A. Drabent

`https://git.astron.nl/RD/LINC`

Workflow 1: LoTSS radio interferometry processing

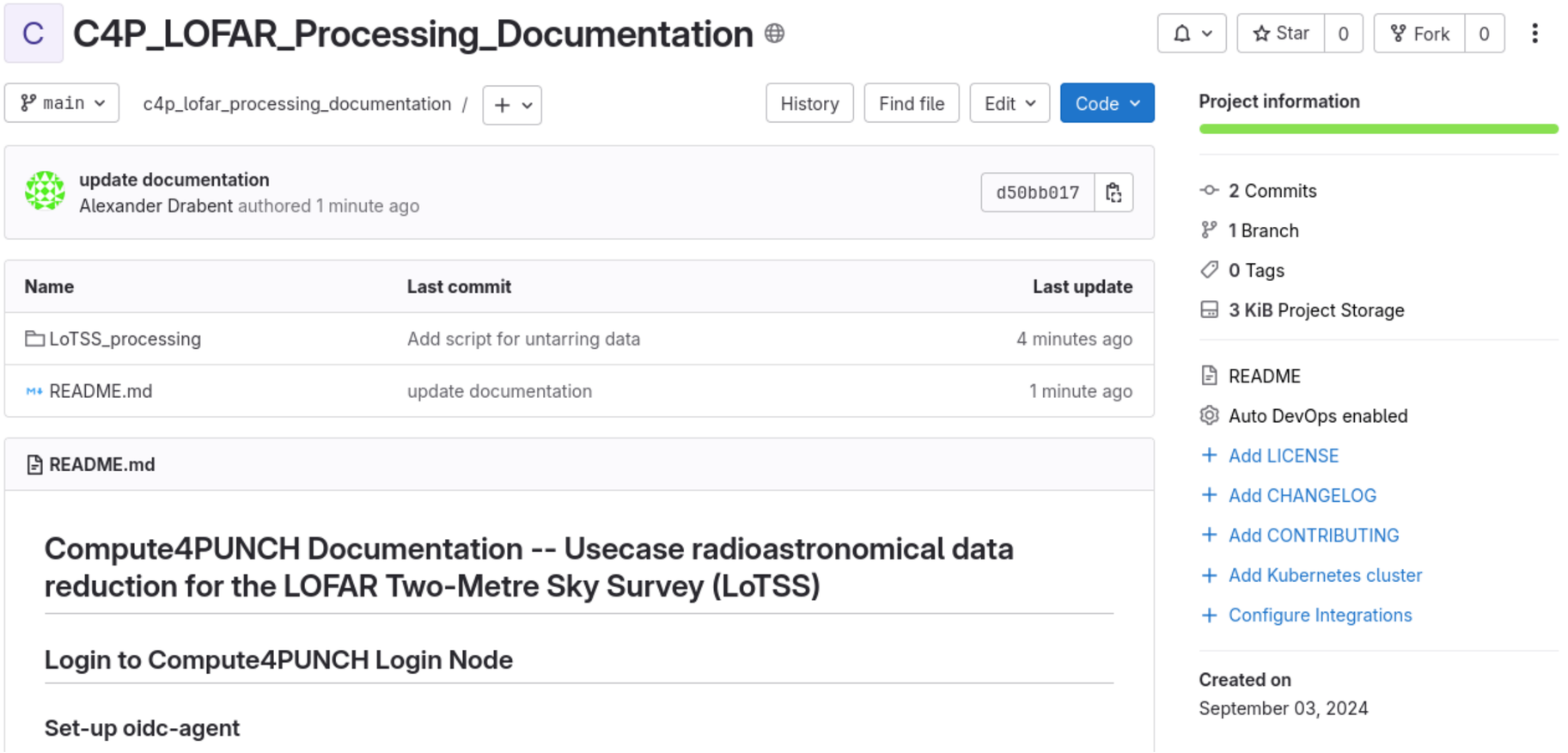
Data transfer challenge :



Slide: A. Drabent

Workflow 1: LoTSS radio interferometry processing

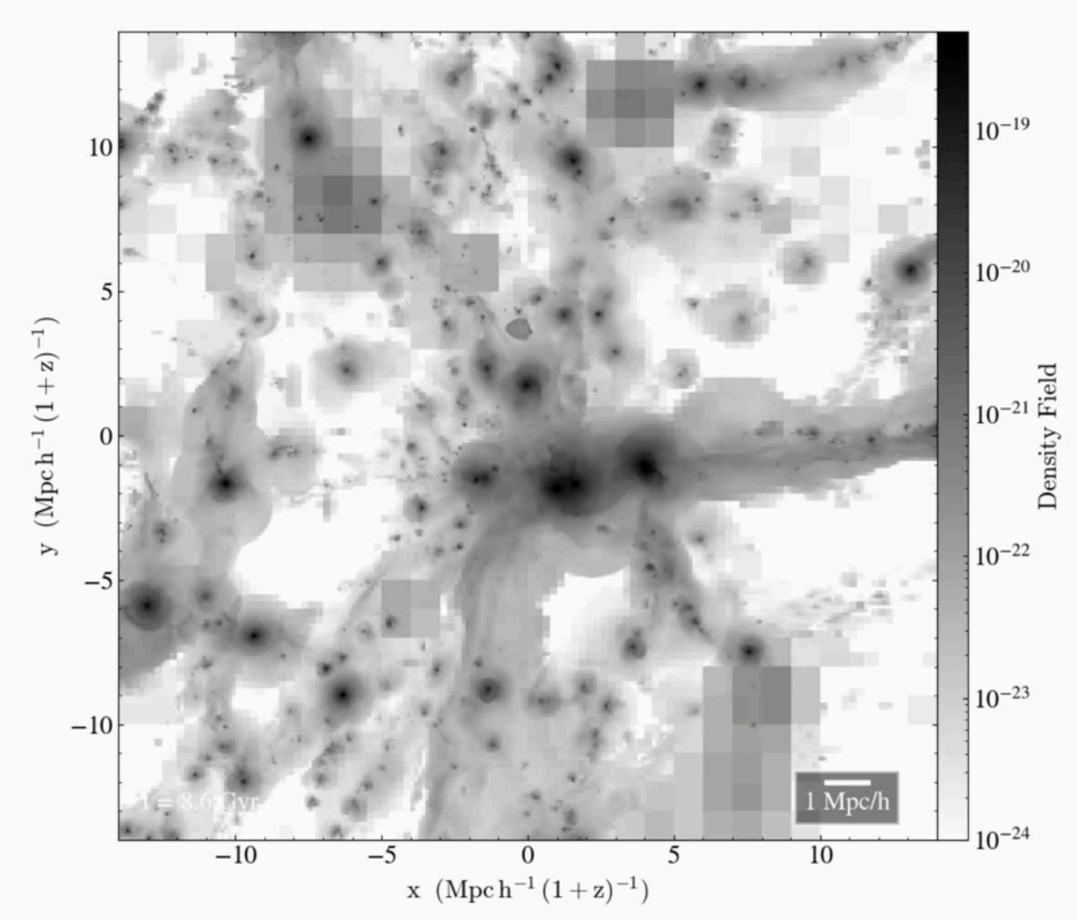
Documentation :



https://gitlab-p4n.aip.de/compute4punch/tutorials/c4p_lofar_processing_documentation

Workflow 2: Postprocessing of cosmological simulations

- Large outputs of numerically expensive simulations
  Postprocessing: requires large analysis facilities
- Data: in Storage4PUNCH or external
- Analysis software in container
- Entry point: reana
  can access storage with token
  (‘secret’ in reana)

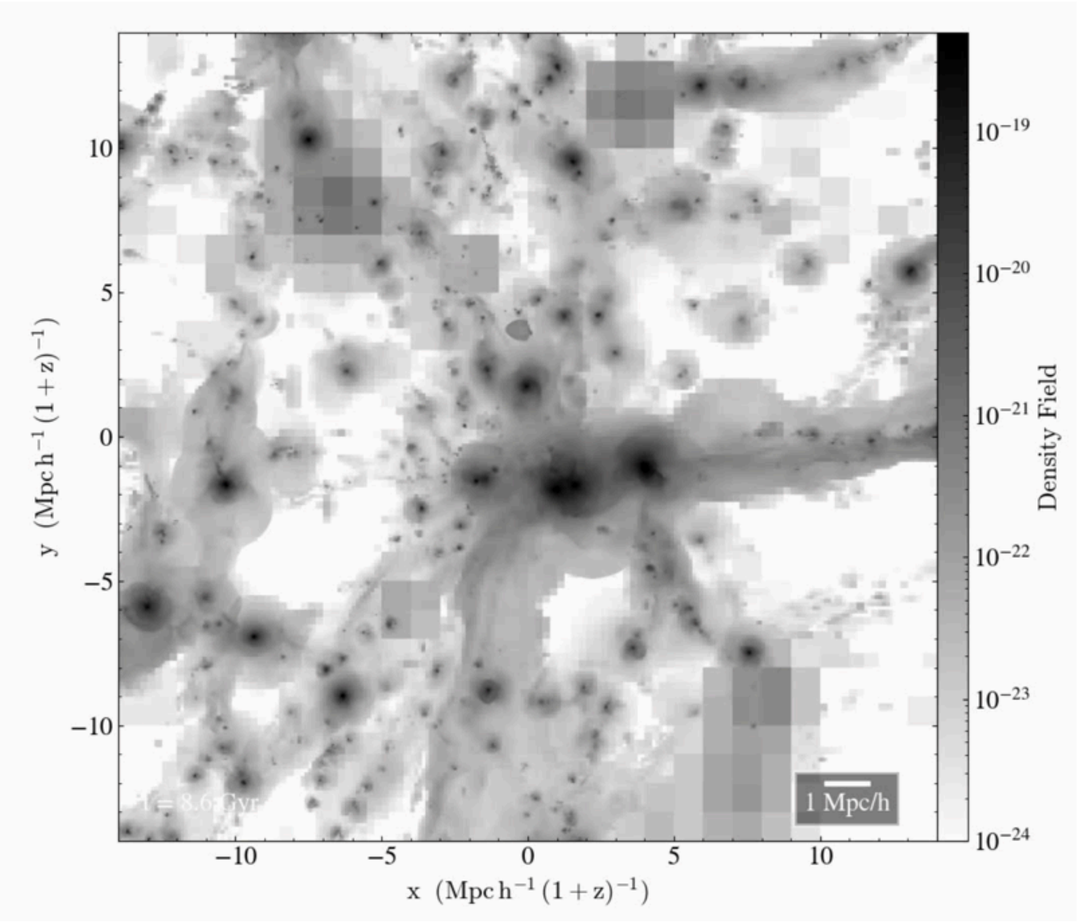


Slide: P. Gupta

Workflow 2: Postprocessing of cosmological simulations

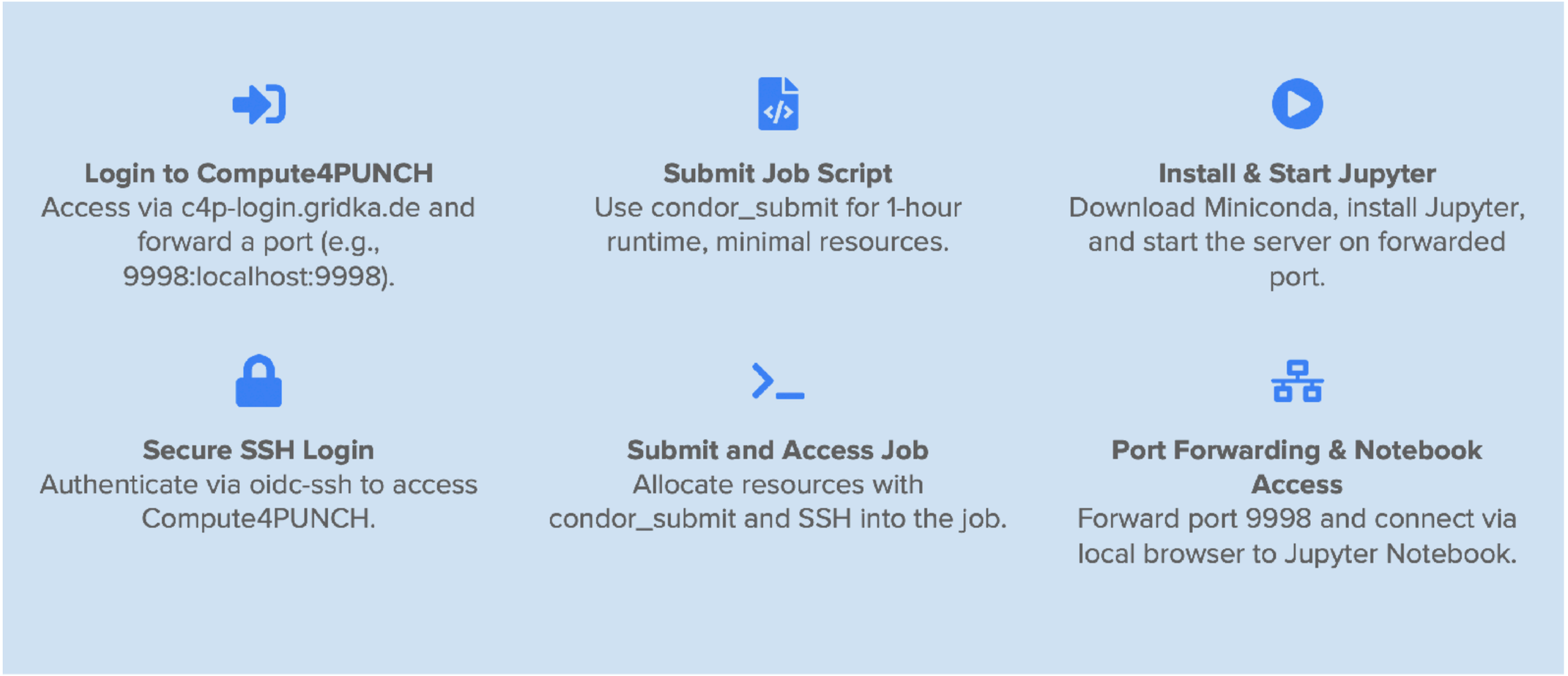
reana.yaml

```
inputs:
  files:
    - multiple_plot_test.py
    - download_data.sh
    - upload_data.sh
  parameters:
    maincode: multiple_plot_test.py
    getdata: download_data.sh
    uploadResult: upload_data.sh

workflow:
  type: serial
  specification:
    steps:
      - name: Make Directories and Download Simulat
        environment: 'docker.io/tlsprateek/davixdoc
        commands:
          - mkdir Simulated_Data
          - mkdir Output
          - bash "${getdata}"

      - name: Prepare Plots
        environment: 'docker.io/tlsprateek/ytprojec
        commands:
          - python3 "${maincode}"

      - name: Upload the Output
        environment: 'docker.io/tlsprateek/davixdoc
        commands:
```



Example 3: Jupyter Environment for ML-PPA

Machine Learning-based Pipeline for Pulsar Analysis



https://gitlab-p4n.aip.de/punch/intra-docs-content/-/blob/master/files/TA5/Compute4PUNCH_Access_Usability_JupyterNotebooks.pdf

Slide: A. Redelbach
G. Ravindra Dange

Conclusions

- Storage4PUNCH & Compute4PUNCH prototypes are available

- Prototypes needs to be tested and commissioned by community specific workflows (Demonstrator workflows)

- Several use case examples are implemented, for instance
  - LoTSS observation processing → data transfer
  - Postprocessing of cosmological simulation output → reana
  - ML-PPA : → enable jupyter noteboks

- Lessons learned:
  - Requires extra effort to implement workflows
  - allows to use federated infrastructure