

# Python for Big Data in Climate Science

Aaron Spring (Max-Planck-Institute for Meteorology)

# Goals

- Demonstrate the data challenges of a climate scientists
- Show the analysis workflow of a climate scientist using python

# Outline

- What's a climate model? How does climate data output look like?
- Who to get climate data to shine (easily)? Live demo
- Science in the cloud

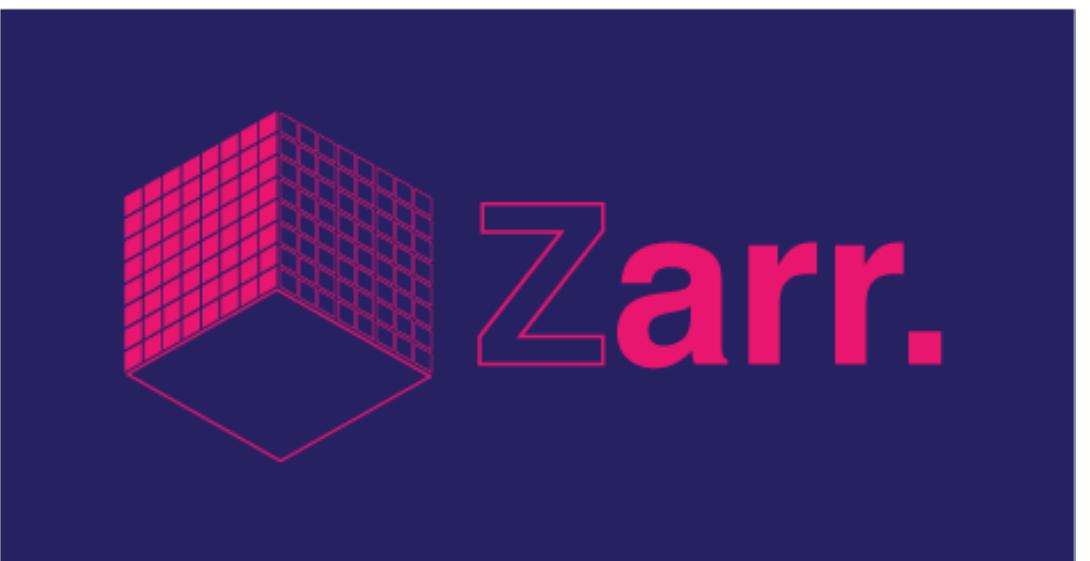
# This is Aaron

- Background in physics
- Currently PhD candidate
  - ▶ Max-Planck-Institute for Meteorology in Hamburg
  - ▶ Topic: Variability and Predictability in the Global Carbon Cycle
  - ▶ Graduate early 2021
- Co-founder of `climpred`: ensemble forecast verification python package
- Loves ☀️, plays ⚽️🎹, enjoys 🏜️



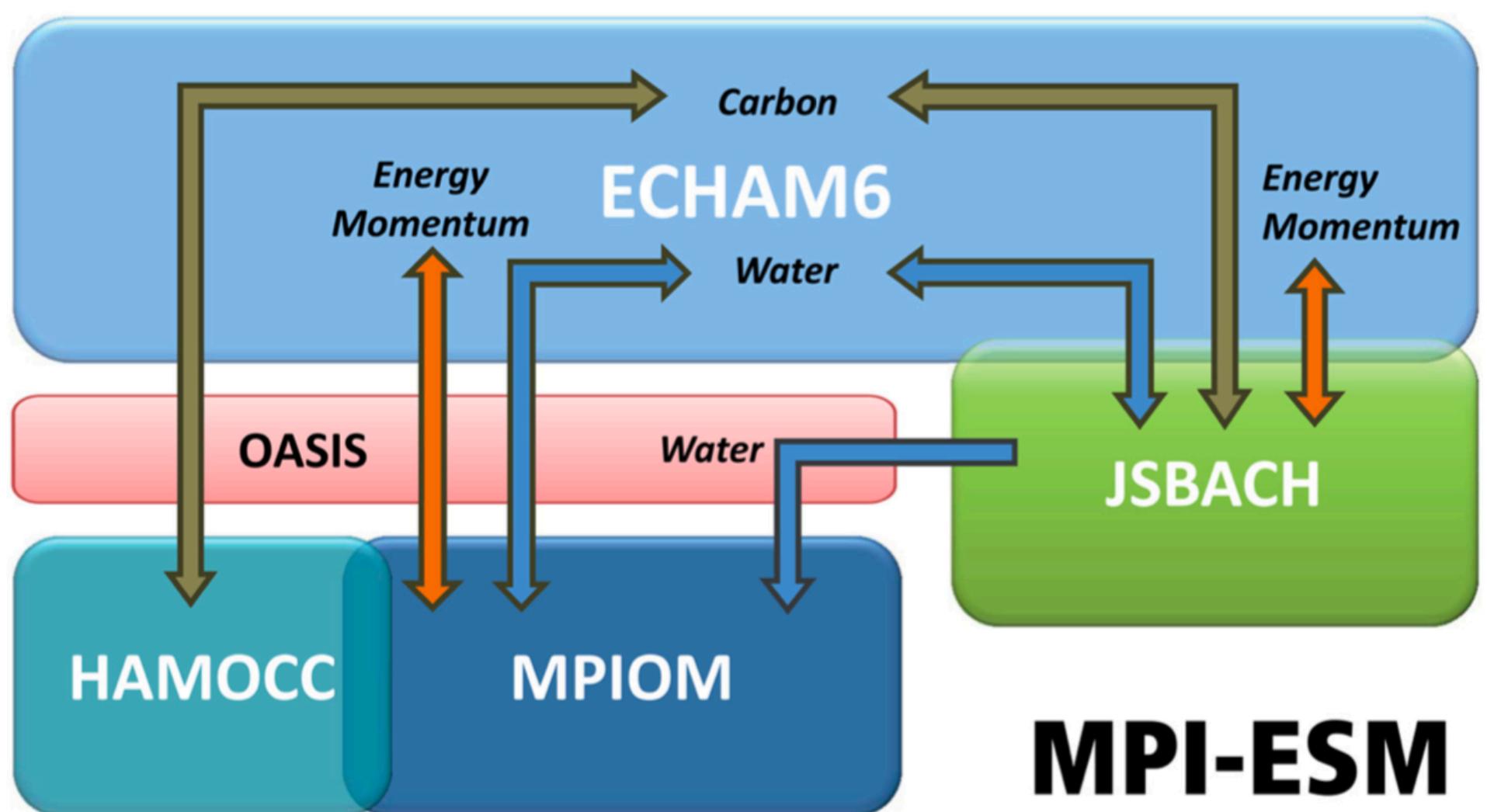
# Climate Data

- Multi-dimensional, labeled and homogeneously formatted data
- Geosciences: (dimensions `time`, `longitude`, `latitude`, `depth/height`, ...)
  - ▶ Weather forecast
  - ▶ Climate simulations
  - ▶ Satellite product
- Finance: Stock prices
- Format:
  - ▶ `netcdf`: Network Common Data Form builds upon HDF5
  - ▶ `zarr`: optimized for chunked data in the cloud



# Climate Modeling

- Earth System Model (ESM)
- Simulating one year with MPI-ESM-LR:
  - ▶ 14 GB output
  - ▶ 5 Node hours (1 Node = 48 CPU)



# Workflow without python

- Domain specific languages and tools:
  - ▶ NCL: NCAR Command Language
  - ▶ CDO: Climate Data Operators
- Proprietary software:
  - ▶ MATLAB
- Disadvantages:
  - ▶ Long scripts, copy&paste, little software engineering, different software for computation and visualisation
  - ▶ Little reproducibility: intermediate files, no code sharing culture

# Telling a story based on data

- Analyse available data and visualise results
- Danger of “Computing Too Much and Thinking Too Little” [Emanuel, 2020]
- Ideally:
  - ▶ technically: intuitive, easy, no need to understand every detail under the hood
  - ▶ conceptually: demanding and genius

→ Python ecosystem and

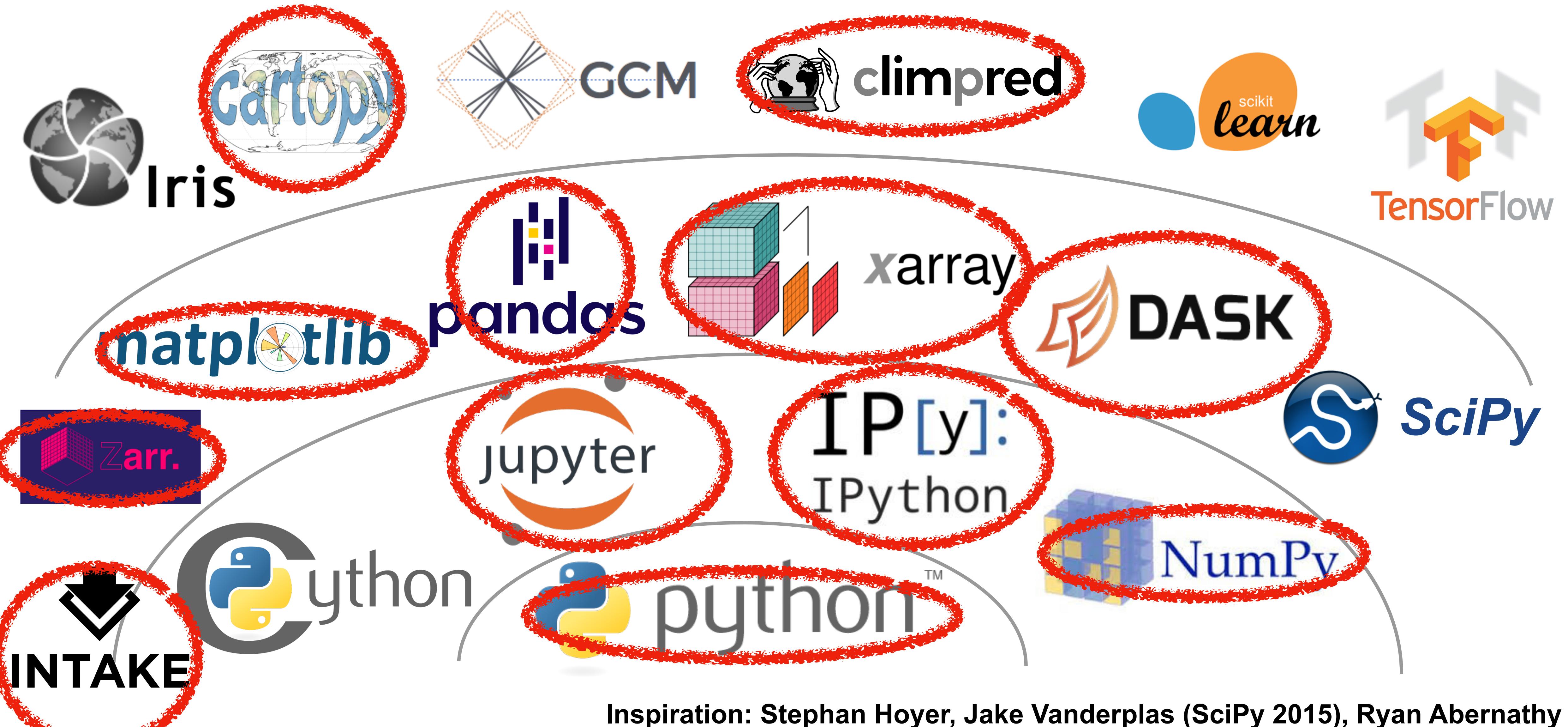


# What is Pangeo?

*“A community platform for Big Data geoscience”*

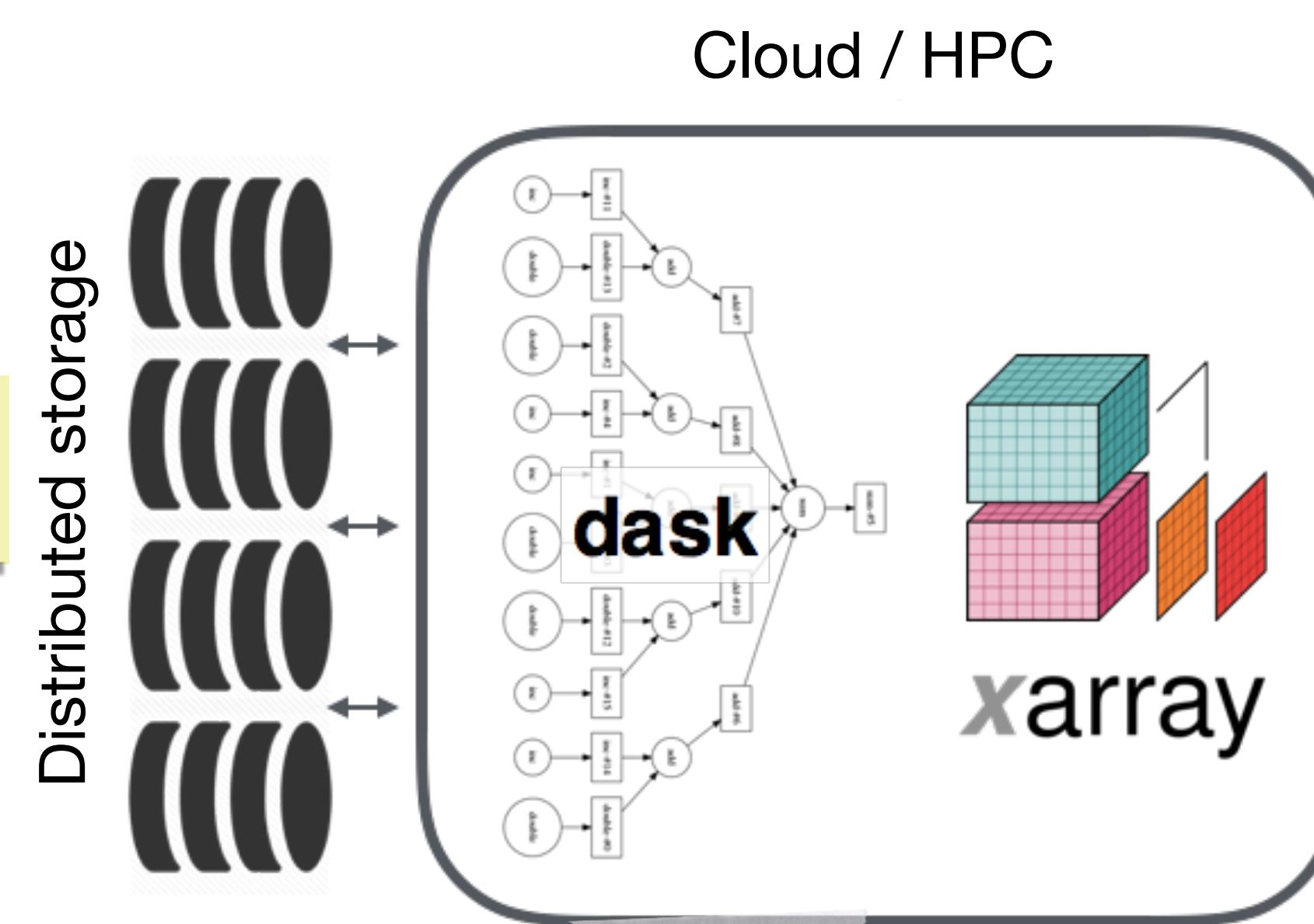
- Open Community
- Open Source Software
- Open Source Infrastructure

# Pangeo Software Ecosystem



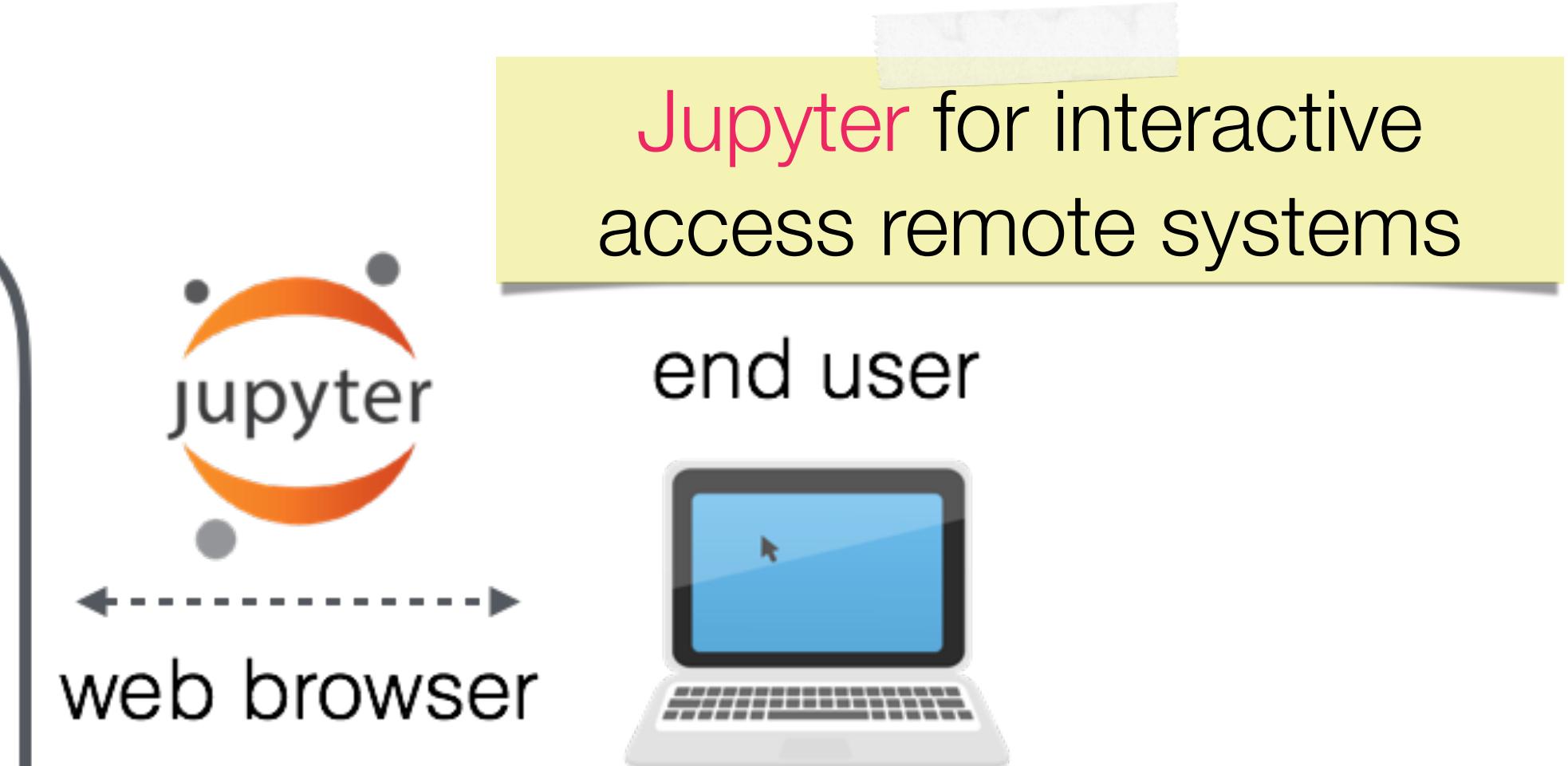
# HPC Architecture

“Analysis Ready Data”  
stored on distributed storage.



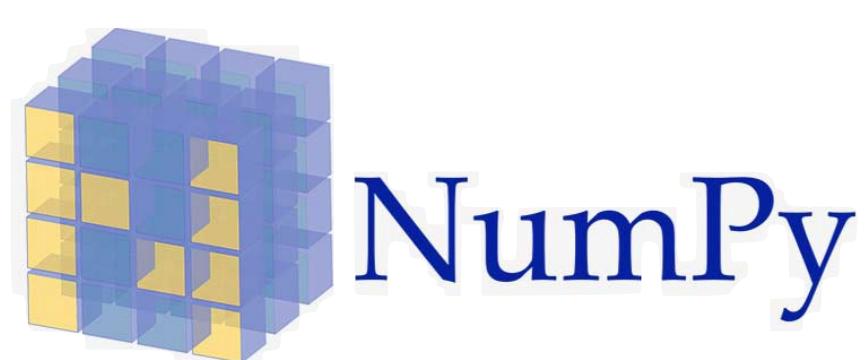
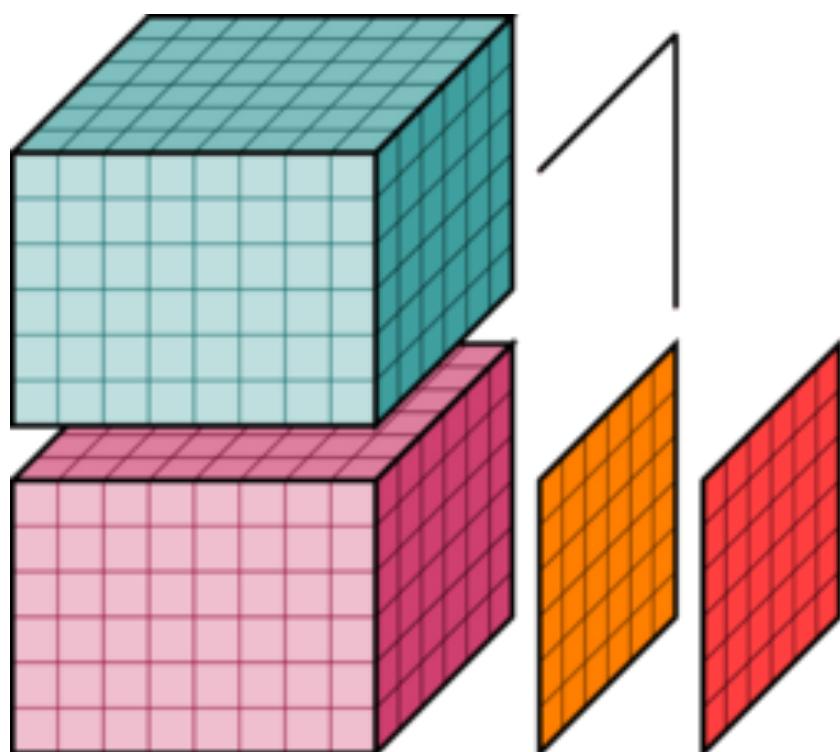
Parallel computing system allows users deploy clusters of compute nodes for data processing.

Dask tells the nodes what to do.

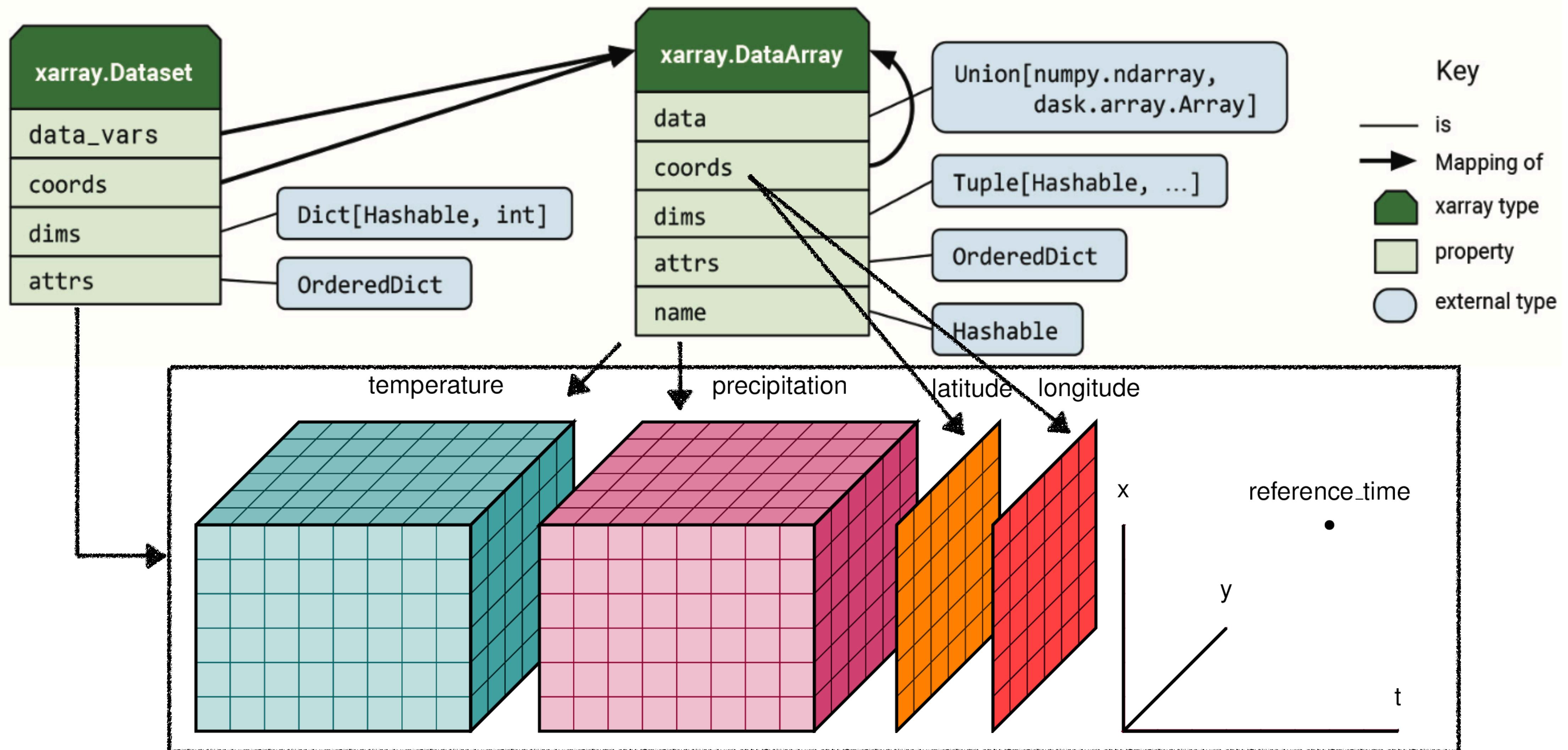


# xarray

- Analysis of multi-dimensional data
- Self-describing data
- Efficient: based on numpy and dask
- Simple: API inspired by numpy and pandas
- Stephan Hoyer and Joe Hamman (2017) “Xarray: N-D Labeled Arrays and Datasets in Python”



# xarray data types

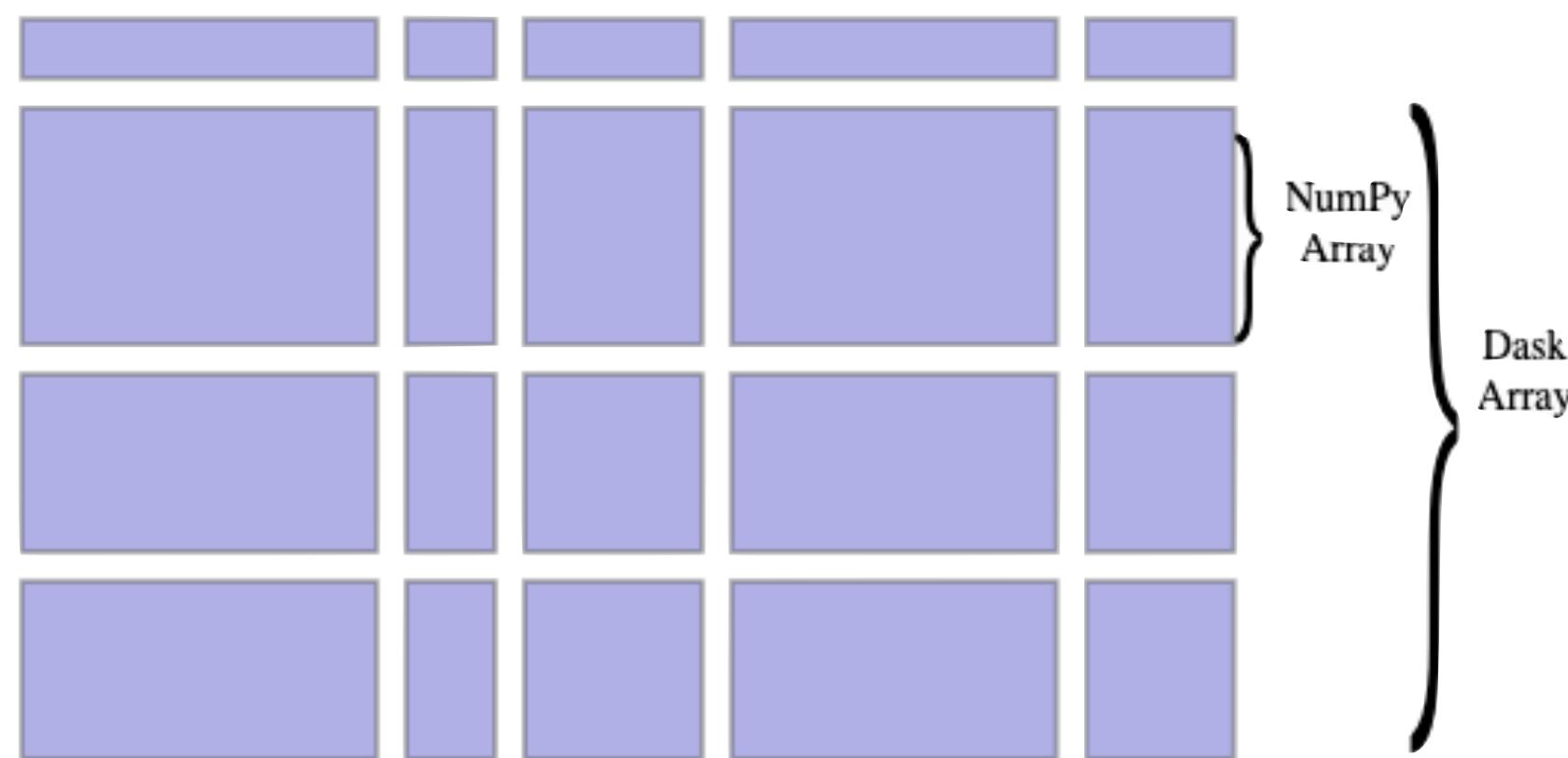


# dask



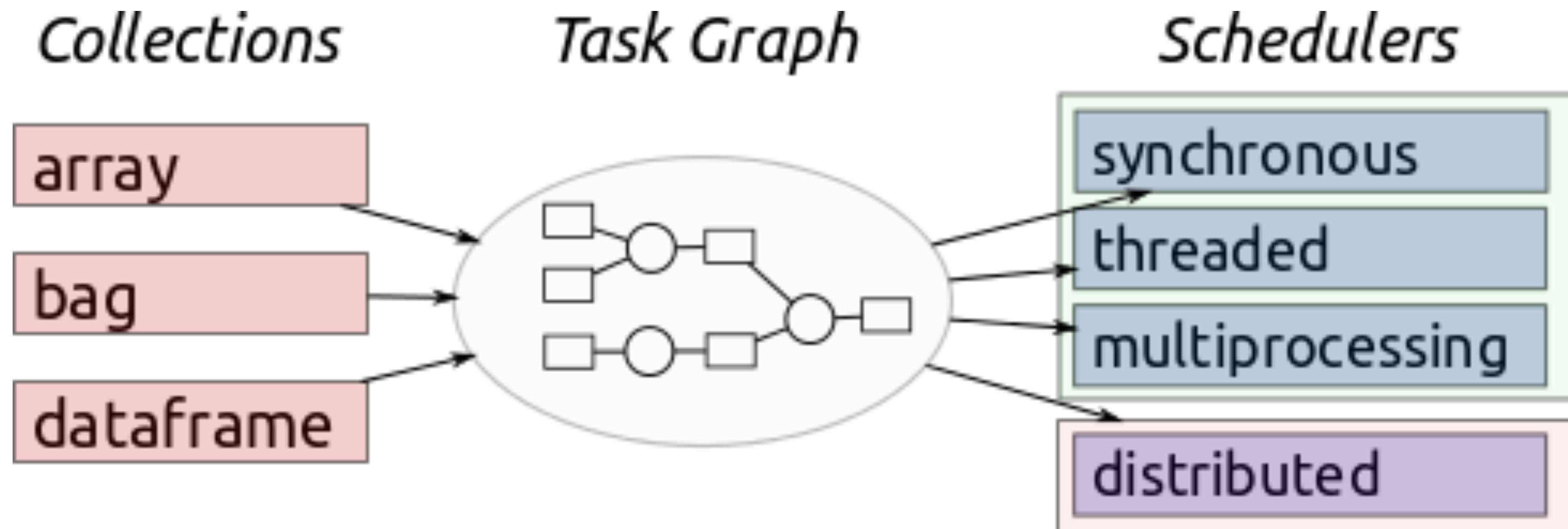
# DASK

- Dynamic task scheduling
- Builds upon `multiprocessing`, `threading` and `concurrent`
- out-of-memory computation via chunking
- Scales from laptop to supercomputer
- Intuitive (known) API from `pandas` and `numpy`
- Matthew Rocklin (2015: “Dask: Parallel Computation with Blocked Algorithms and Task Scheduling”)

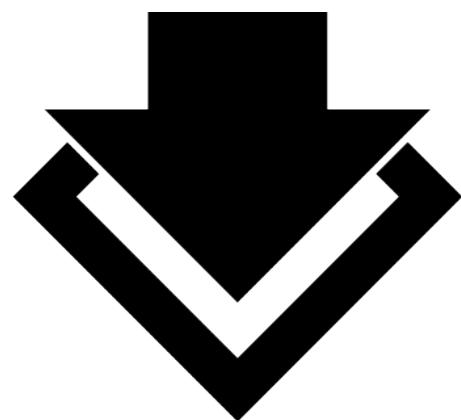


# dask workflow

- dask.array → numpy.array
- dask.bag → iterable
- dask.dataframe → pandas.DataFrame



# intake



## INTAKE

- Taking the pain out of data access
- Finding, investigating, load and disseminating data with catalogs
- version control data sources
- distinction between data curator and analyst roles
- `intake-xarray` plugin: Catalogs pointing to netcdf files
- `intake-esm` plugin: Search and load ESM output
  - Query in `pandas.DataFrame`
  - Load data with `dask` into `xarray`
  - Developed by Anderson Banahirwe (**andersy005**)

# Live demo



OR



DKRZ mistral: <https://www.dkrz.de/up/systems/mistral>

# My best practices

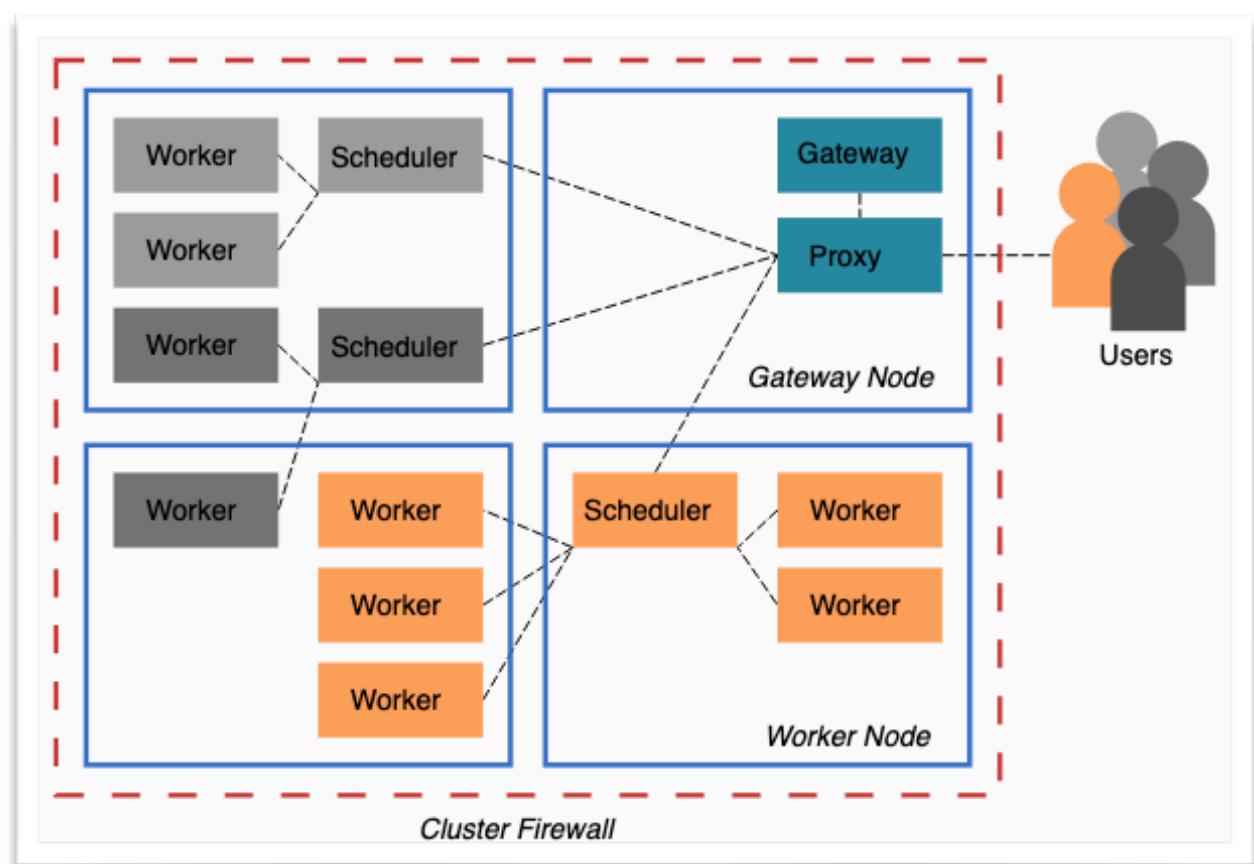
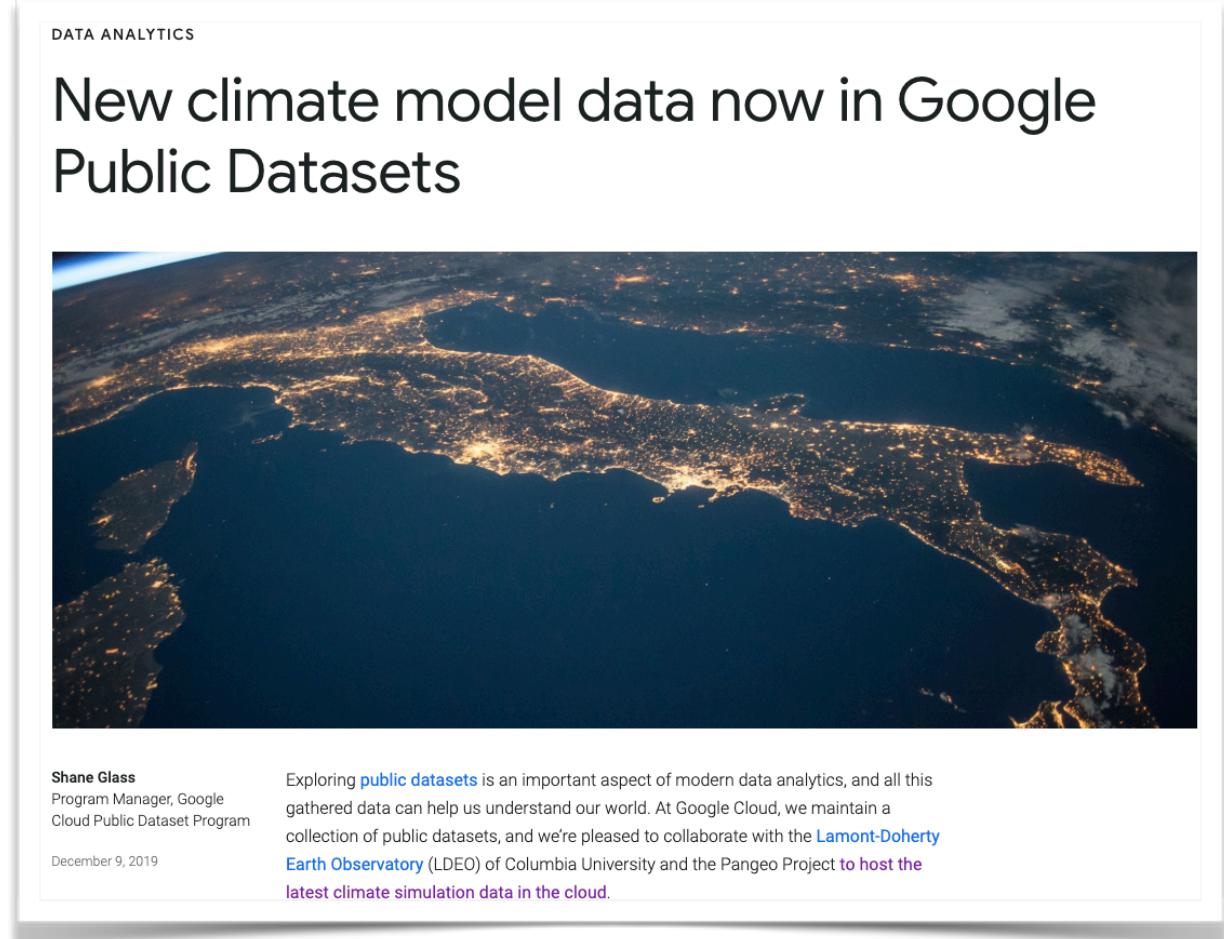
- Aim to concat all your data into one `xr.Dataset` and save to disk (preferably in `zarr`)
- Understand chunking and use `dask` incl. dashboard
- Read the `xarray` and `dask` docs.
- Write re-usable code and share
- Use `intake`
- Pair programming (PRs on GitHub: `climpred`, `xskillscore`, `esmtools`, ...)
- Browse GitHub for smart `xarray` use-cases (start with `pangeo`)

# Extensions to xarray

- `scipy` : (nearly) all functions callable with `xr.apply_ufunc`
- `dask_jobqueue` : parallelise dask across nodes
- `xskillscore` : verification metrics
- `cartopy` : projections of maps
- `geoviews` : dynamic visualisation of geo data
- `regionmask` : spatial aggregation based on shapefiles
- `xesmf` : regridding
- `xgcm` : grid aware operations
- `cmip6_preprocessing` : data cleaning for CMIP6 output
- `climpred` : verification of multi-dim ensemble forecasts
- `intake-xarray` : intake for netcdf files  
... <http://xarray.pydata.org/en/stable/related-projects.html>

# Challenges in Science Big Data

- Dataset size & I/O performance → solution proposal: pangeo cloud 
- Filesystems (lustre) not optimised to load many small files to load fast
- Little knowledge of climate scientists in software engineering, testing, continuous integration, parallelism

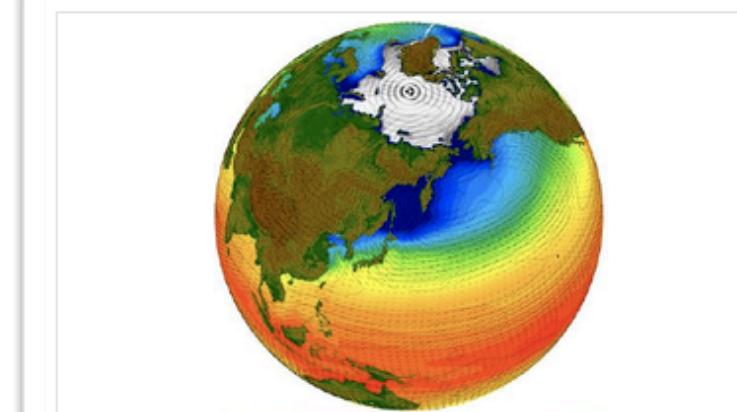


# Reproducible science

- Science in a GitHub repo:
  - ▶ <http://gallery.pangeo.io/>
  - ▶ Data in the cloud
  - ▶ nbgitpuller to execute and render notebook
  - ▶ reproducible with binder

## PANGEO GALLERY

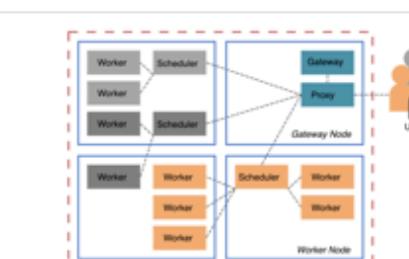
Welcome to the Pangeo Gallery website. This site allows you to browse different Pangeo use cases. The site is organized into galleries, listed below, containing one or more notebooks. Each gallery is hosted in a standalone GitHub repository. If you're interested in contributing a new gallery, please see the [Contributor Guide](#).



### GALLERY FOR CESM LENS ON AWS

A gallery of notebooks developed to demonstrate analysis of CESM LENS data publicly available on Amazon S3 (us-west-2 region) using xarray and dask

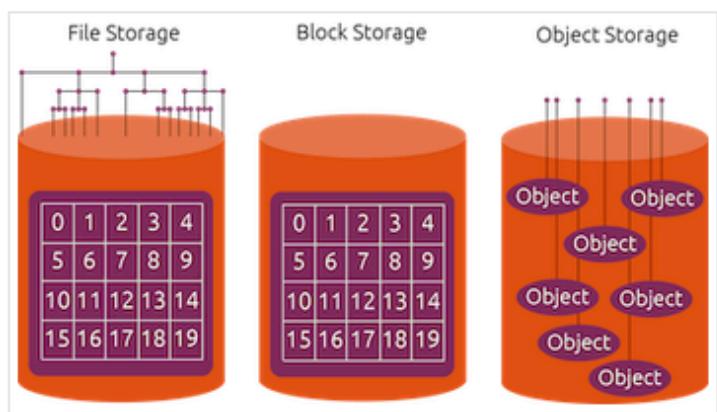
license [BSD-3-Clause](#) last commit [june](#)  
 [passing](#)  [launch](#) [binder](#)



### PANGEO & DASK GATEWAY.

How to use Dask Gateway on Pangeo Hubs and Binders for scalable computing.

license [MIT](#) last commit [april](#)  
 [passing](#)  [launch](#) [binder](#)



### CLOUD STORAGE BENCHMARKS

Investigation of the throughput of various cloud storage formats and services.  
 Prepared for the 2020 EarthCube Meeting by Ryan Abernathey.

license [MIT](#) last commit [june](#)  
 [passing](#)  [launch](#) [binder](#)

→ Play with climate data yourself: 

# References

- Papers:
  - ▶ Rocklin, M. (2015). Dask: Parallel Computation with Blocked algorithms and Task Scheduling. 126–132. doi: [10/gfz6s5](https://doi.org/10/gfz6s5)
  - ▶ Hoyer, S., & Hamman, J. (2017). xarray: N-D labeled Arrays and Datasets in Python. Journal of Open Research Software, 5(1). doi: [10/gdqdmw](https://doi.org/10/gdqdmw)
  - ▶ Emanuel, K. (2020). The Relevance of Theory for Contemporary Research in Atmospheres, Oceans, and Climate. *AGU Advances*, 1(2), e2019AV000129. doi: [10/gg3dzt](https://doi.org/10/gg3dzt)
- Pictures:
  - ▶ xarray website, dask website, MPIM, DKRZ, pangeo
- Tutorials:
  - ▶ xarray: [https://xarray-contrib.github.io/xarray-tutorial/scipy-tutorial/00\\_overview.html](https://xarray-contrib.github.io/xarray-tutorial/scipy-tutorial/00_overview.html)
  - ▶ dask: [https://tutorial.dask.org/03\\_array.html](https://tutorial.dask.org/03_array.html)
  - ▶ pangeo: <http://gallery.pangeo.io/>