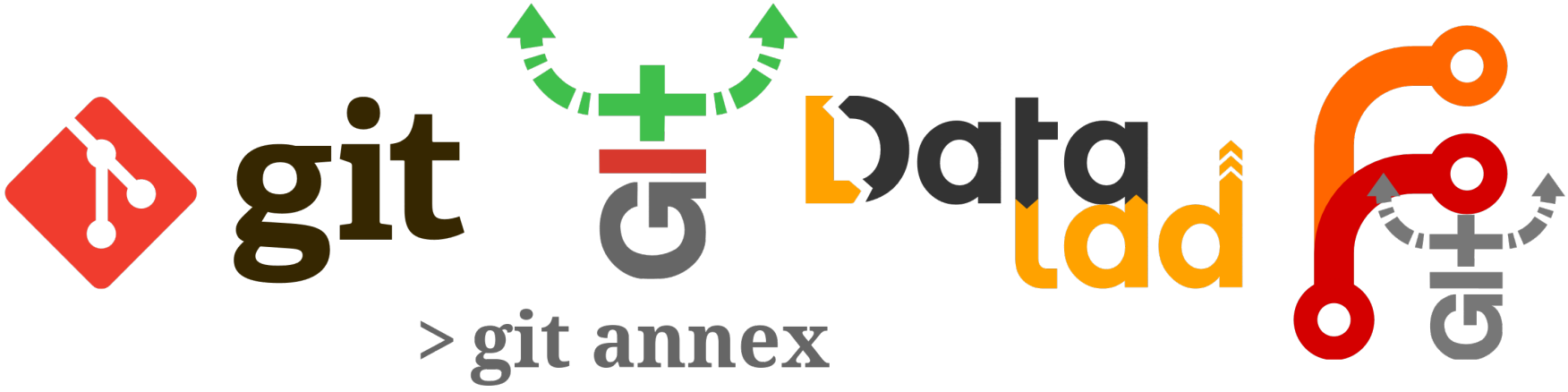


GIT/GIT-ANNEX/DATALAD/FORGEJO-ANEKSAJO

A pragmatic data collaboration ecosystem

2025-03-21 | Matthias Rißе | ICE-4

Who am I?

- Matthias Rißbe
- MSc in Applied Mathematics and Computer Science and MaTSE
- Part of the IT team at ICE-4
- Software developer, Sysadmin, Data Management guy, Technical support person, ...
- Maintainer of Forgejo-aneksajo
- Generally an open source enthusiast

Overview

Vision

Problem and Solution

The ecosystem and how it works

Forgejo-aneksajo

Vision

- Every publication should have an accompanying publishable artifact that fully describes the data, code and computational environment used to produce it
- Anyone[1] should be able to go from e.g. a plot in a paper back through the processes used to produce it, be able to follow any intermediate steps, and arrive at the data, or even the data acquisition process
- Questions like this need to be answerable:
  - Where did the raw data come from?
  - What processing steps were done?
  - Which software was used, and in what version?
- This ”log” should be a natural by-product of the creation process

[1] with sufficient determination

The issue

```
├── code/
│   ├── code_final/
│   │   ├── final_2/
│   │   │   ├── main_script_fixed.py
│   │   │   └──takethisscriptformostthingsnow.py
│   │   ├── utils_new.py
│   │   ├── main_script.py
│   │   ├── utils_new.py
│   │   ├── utils_2.py
│   │   └── main_analysis_newparameters.py
│   └── main_script_DONTUSE.py
├── data/
│   ├── data_updated/
│   │   └── dataset1/
│   │       └── datafile_a
│   ├── dataset1/
│   │    └── datafile_a
│   ├── outputs/
│   │   ├── figures/
│   │   │   ├── figures_new.py
│   │   │   └── figures_final_forreal.py
│   │   ├── important_results/
│   │   ├── random_results_file.tsv
│   │   ├── results_for_paper/
│   │   ├── results_for_paper_revised/
│   │   └── results_new_data/
│   ├── random_results_file.tsv
│   ├── random_results_file_v2.tsv

[...]
```

The solution

Use a version control system for everything:

- VCS for code: git
- VCS for data: git-annex/DataLad
- VCS for computational environments: still git, but use a package manager that can uniquely define an environment for each commit: pixi, nix, guix, ...

$\rightarrow$ every aspect is part of a self-contained repository, i.e. an artifact that fully describes the research results

git/git-annex/DataLad

Git is a decent version control system:

- tracks the evolution of content (a hierarchical directory structure) through annotated snapshots in time (commits)
- allows concurrent diverging developments (branches) ...
- ... as well as reconciliation of those developments (merge, rebase, fast-forward, cherry-pick, ...)
- distributed and local-first:
  - works offline, e.g. on an airplane or a field campaign
  - good sync support to keep different ”clones” up-to-date
- essentially a standard tool already

In summary:

- git is a good decentralized, distributed, versioned database ...
- ... but it is bad at handling large files

git/git-annex/DataLad

Enter git-annex:

- a layer of indirection: git repository is an index of ”annexed files”
- git stores filenames and some metadata, the actual content is in a distributed key-value store (in .git/annex/objects)
- each clone of a repository can contain none, some, or all of the annexed files
- git-annex provides commands and mechanism to modify what is stored where (get, drop, copy, move, sync preferred content ...)
- special (non-git) remotes incorporate other storage platforms (web, torrent, S3, webdav, rsync, rclone, ..., or build your own)

git/git-annex/DataLad

And DataLad:

- terminology: datasets are git-annex-enabled git repositories with a dataset ID
- streamlines more common workflows
  - updating the index (i.e. push/pull of the git repository, downloading files, uploading new or changed files)
- provenance tracking of dataset mutations
  - What command was run to create some output(s) from some input(s)?
- Fantastic documentation: `https://handbook.datalad.org/`
- Better support for datasets as building blocks
  - nested, reusable datasets

Datasets are building blocks

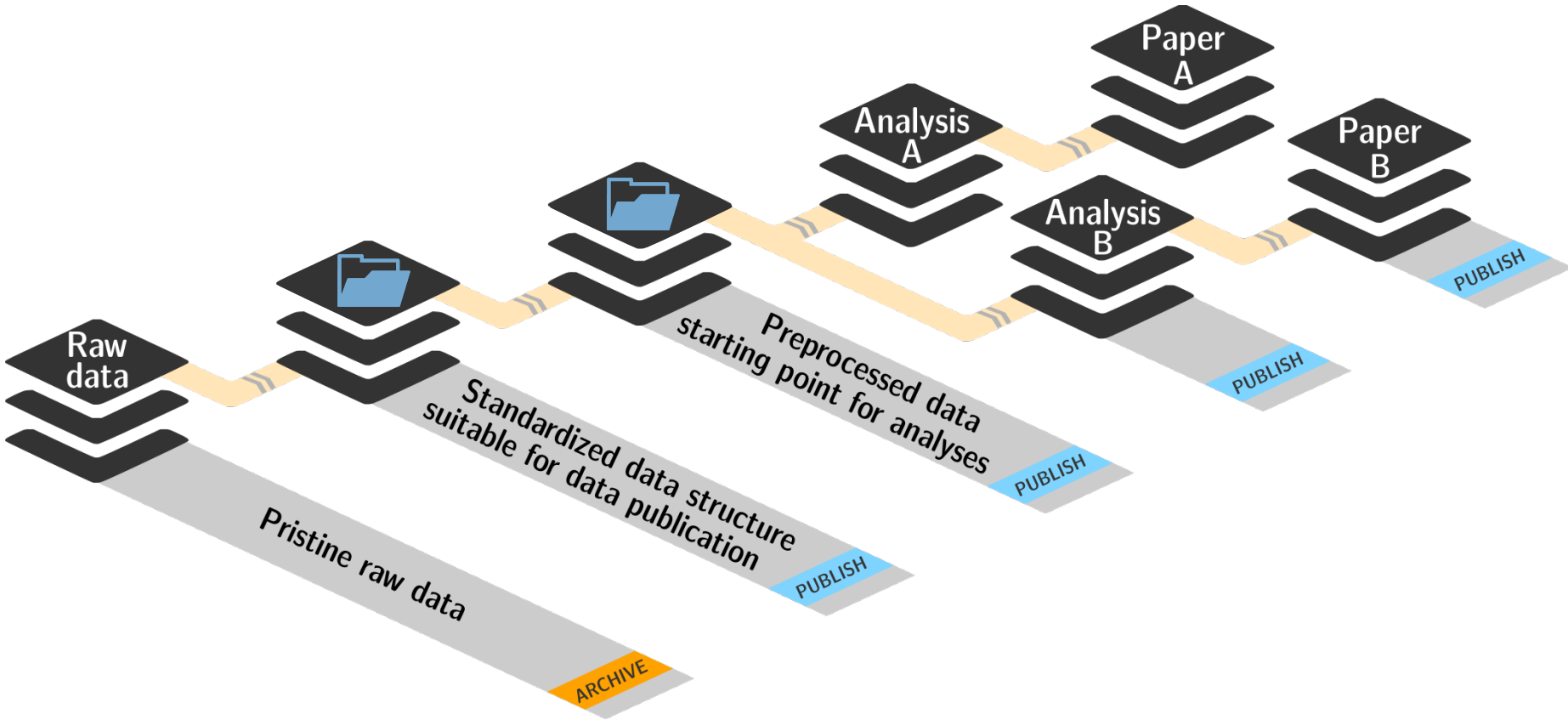


Figure:
`https://handbook.datalad.org/en/latest/basics/101-127-yoda.html`

Live demo

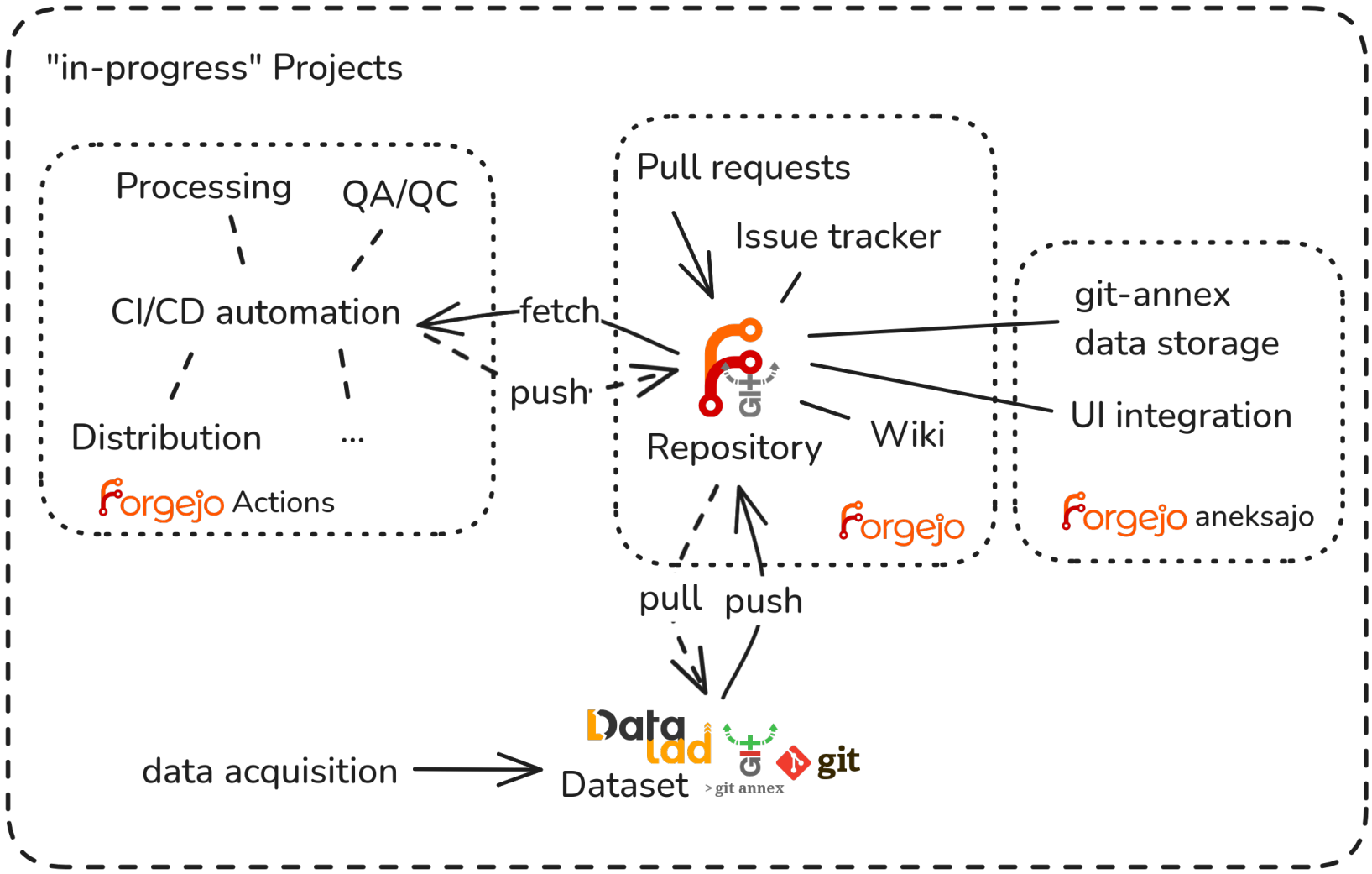
```
# To start off, we will clone the datasets repository ...
$ datalad clone git@atris.fz-juelich.de:m.risse/era5-sat-ts-analysis-paper.git
install(ok): /tmp/tmp.wor9L9LSFn/era5-sat-ts-analysis-paper (dataset)

# ... and cd into it.
$ cd era5-sat-ts-analysis-paper

# Since we will be digging a bit deeper into it, we can prefetch all contained sub-datasets.
$ datalad get -n -r .
[INFO   ] Ensuring presence of Dataset(/tmp/tmp.wor9L9LSFn/era5-sat-ts-analysis-paper) to get /tmp/tmp.wor9L9LSFn/era5-sat-ts-analysis-paper
install(ok): /tmp/tmp.wor9L9LSFn/era5-sat-ts-analysis-paper/inputs/era5-sat-ts-analysis (dataset)
install(ok): /tmp/tmp.wor9L9LSFn/era5-sat-ts-analysis-paper/inputs/era5-sat-ts-analysis/inputs/era5-sat-timeseries (dataset)
install(ok): /tmp/tmp.wor9L9LSFn/era5-sat-ts-analysis-paper/inputs/era5-sat-ts-analysis/inputs/era5-sat-timeseries/inputs/era5-2t (dataset)
action summary:
  install (ok: 3)

# This is what we are working with:
$ ls --color=always -l
total 420
-rw-rw-r-- 1 icg149 icg149     29 Mär 20 13:54 all.tikzdefs
-rw-rw-r-- 1 icg149 icg149    102 Mär 20 13:54 all.tikzdefs.license
-rw-rw-r-- 1 icg149 icg149    693 Mär 20 13:54 all.tikzstyles
-rw-rw-r-- 1 icg149 icg149    102 Mär 20 13:54 all.tikzstyles.license
-rw-rw-r-- 1 icg149 icg149  83509 Mär 20 13:54 climate-ml.bib
-rw-rw-r-- 1 icg149 icg149    102 Mär 20 13:54 climate-ml.bib.license
drwxrwxr-x 2 icg149 icg149   4096 Mär 20 13:54 code
-rw-rw-r-- 1 icg149 icg149   4489 Mär 20 13:54 fhacThesis.cls
-rw-rw-r-- 1 icg149 icg149    567 Mär 20 13:54 flake.lock
-rw-rw-r-- 1 icg149 icg149   2326 Mär 20 13:54 flake.nix
lrwxrwxrwx 1 icg149 icg149    128 Mär 20 13:54 folien_kolloquium.pdf -> .git/annex/objects/Kg/Pk/MD5E-s4052039--1a437138c79caf63904c774f57dcdd13.pdf/MD5E-s4052039--1a437138c79caf63904c774f57dcdd13.pdf
```

Forgejo-aneksajo



JÜLICH Forschungszentrum

Forgejo-aneksajo

- a data collaboration platform based on Forgejo
  - Forgejo is a git forge like GitHub or GitLab
- one possible storage location for git-annex repositories
- datasets are repositories and have some common features:
  - issue trackers for discussion and coordination
  - pull requests for change proposals and review
  - CI support for quality assurance, common processing steps, or other automation
  - tags and releases
  - a wiki
  - ...
- organisations provide groups and users can have different access levels in orgs and repositories

Live showcase

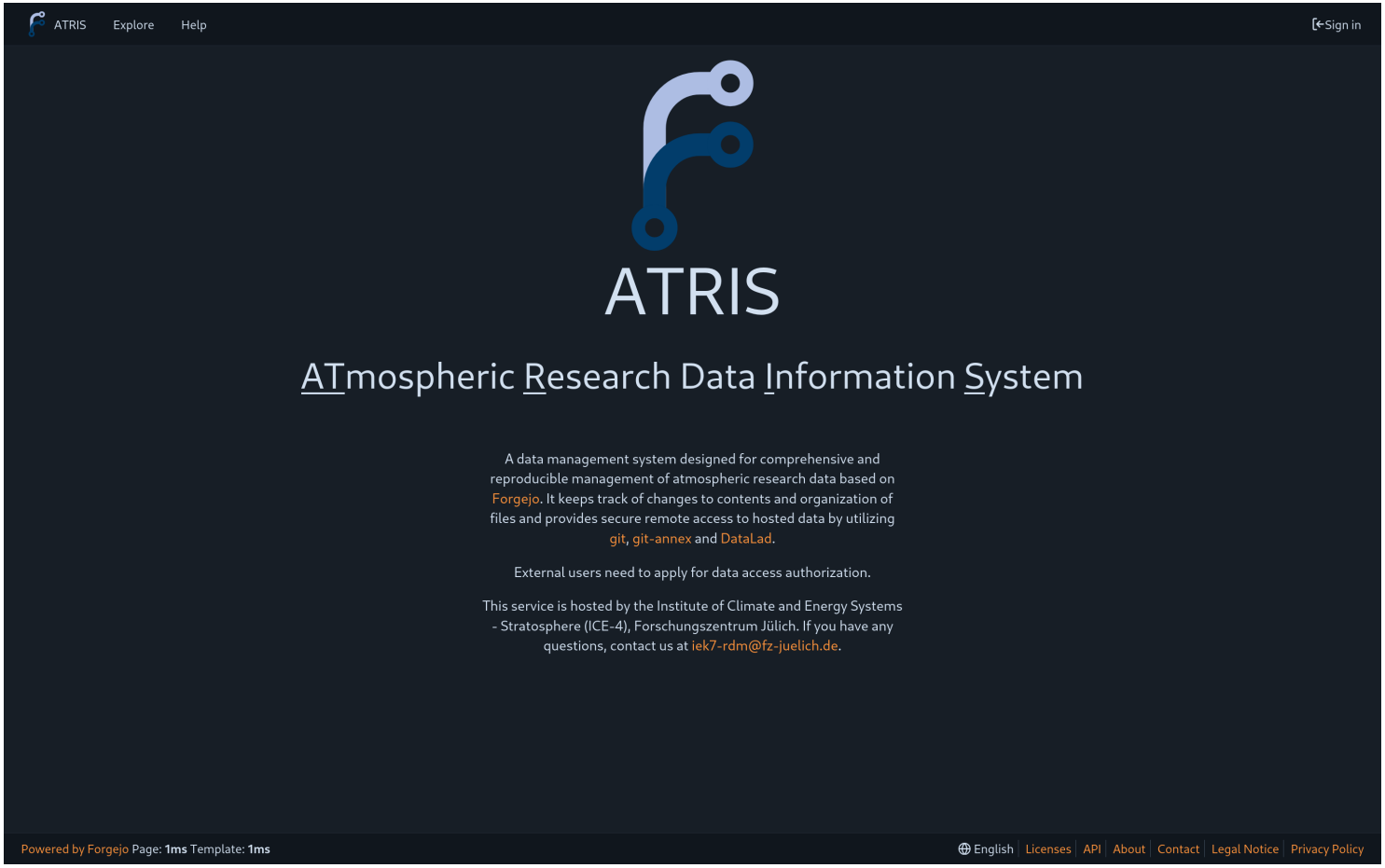


Figure: `https://atris.fz-juelich.de`

Summary

- git/git-annex/DataLad/Forgejo-aneksajo extend established tools and practices from software development to data projects
- You can version control code, data, and software environments within one unified system
- Fits into ”legacy” projects as long as you are dealing with files and directories
- Repositories become self-contained research artifacts
- Forgejo-aneksajo provides a mature collaboration platform backed by a large developer community

Some links

- DataLad: `https://www.datalad.org/`
- DataLad Handbook: `https://handbook.datalad.org/`
- Distribits, a community conference on distributed data management (23.10. - 25.10.2025): `https://distribits.live/`
- git-annex: `https://git-annex.branchable.com/`
- git: `https://git-scm.com/`
- Forgejo-aneksajo: `https://codeberg.org/forgejo-aneksajo/forgejo-aneksajo`

Thank you for your attention!