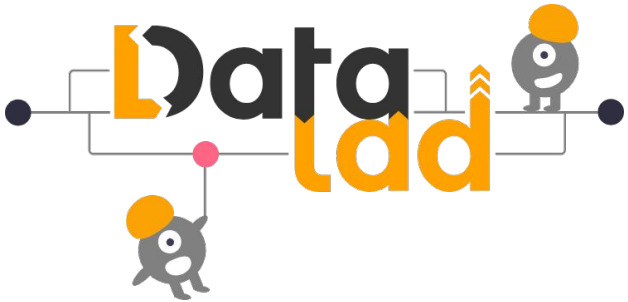
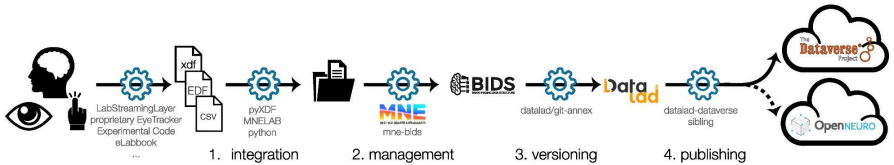
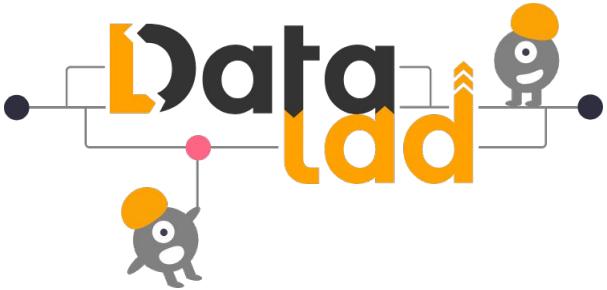
A Gentle Intro to Datalad and LSLAutoBIDS







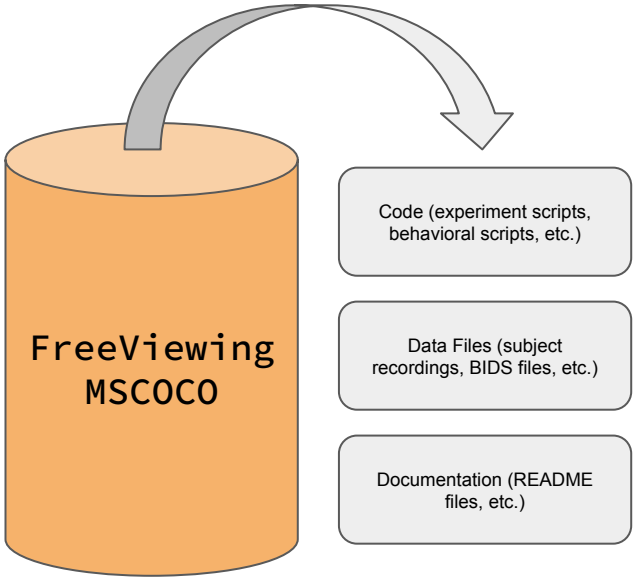
- **Big Picture :** Datalad is distributed data management system - which does ***version control***, ***structure data***, ***supports collaboration***, ***support integration to wide data infrastructure*** (e.g. cloud storage services, repository hosting services, etc)

- Built upon `git` and `git-annex`

- Our use case - version control of **large files/binary files** (data version control) as well as **code/text files** (code version control) of our projects (e.g. FreeViewingMSCOCO)

- Used as CLI or with Python API

Datalad is built upon `git` and `git-annex`



git

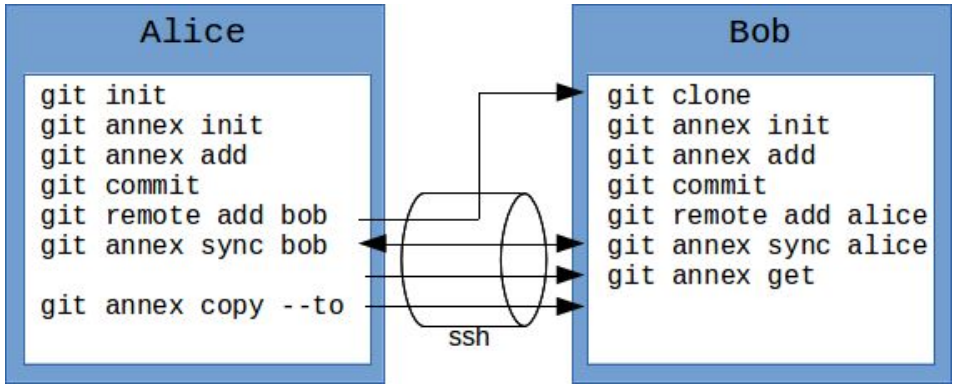
Version control system suitable for text, code, documentation etc.

> git annex

Built on top of git and **manages large files** storing only **a pointer** (shortcut (symlink) or metadata) to the actual file and not the actual contents

What is git-annex?

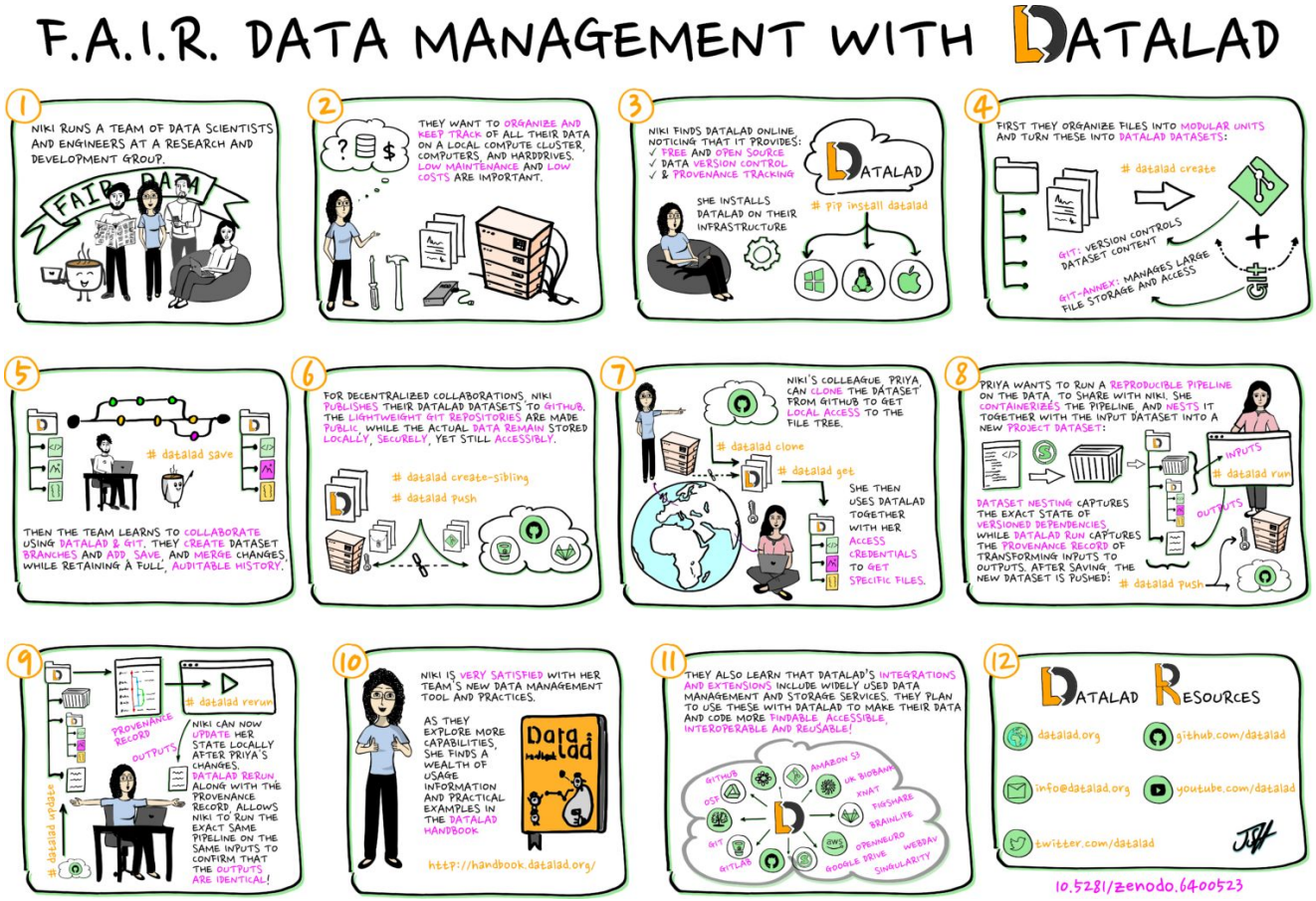
- A distributed file synchronization system written in Haskell.
- Based on git, but instead of tracking file contents it tracks files metadata (symlinks)
- Useful specifically when we deal with **large files** (>10 MB).



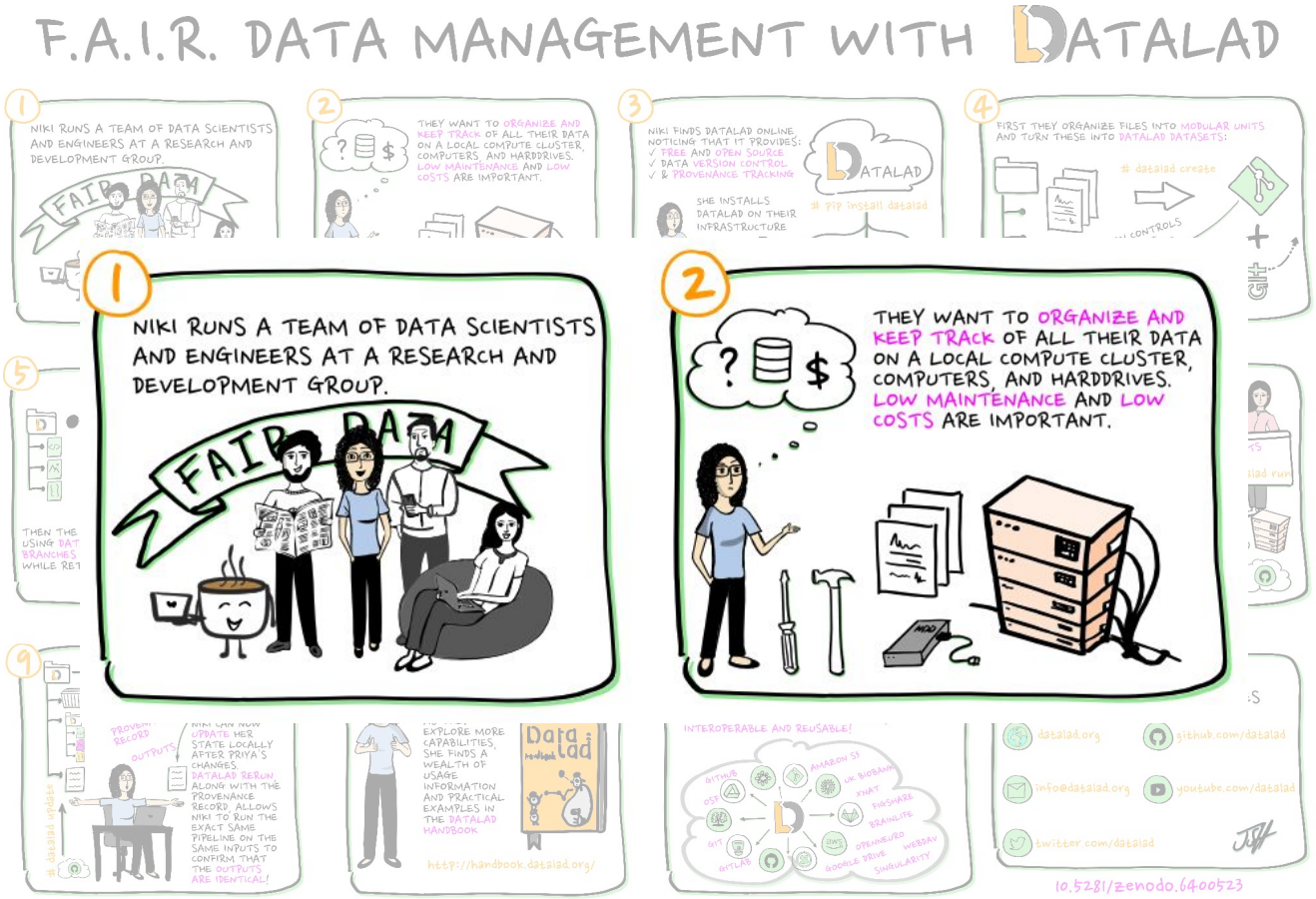
Resources : https://wgpages.netlify.app/annex/

Some Internal Workings of git-annex

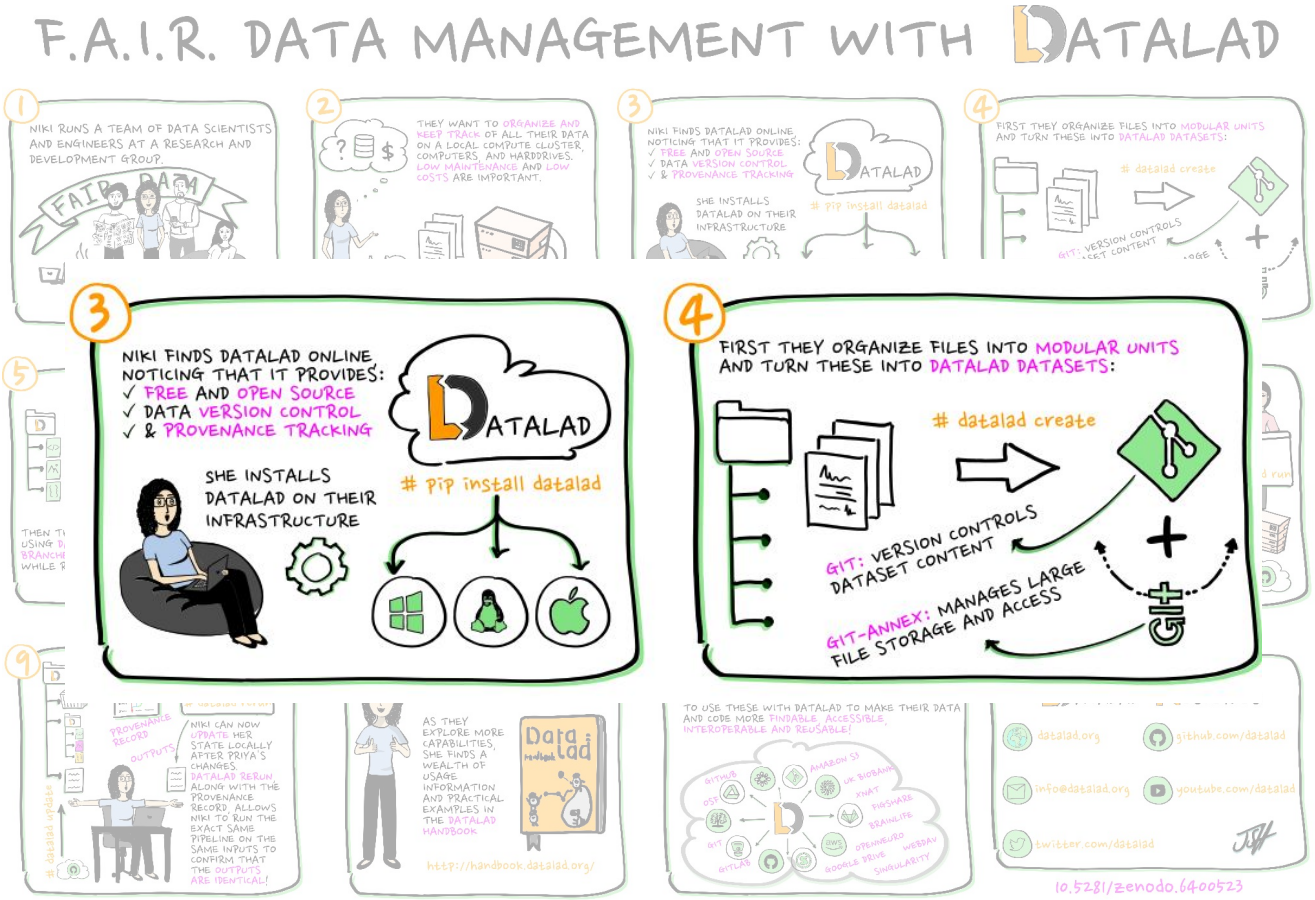
- Files are kept as **read-only** and can only be changed with git annex unlock. (Preventing accidental changes)
- A branch (*git-annex*) keeps logs of when the files are changed.
- Stores data in non git annex remotes (special remotes like web, S3 bucket etc.)
- Useful to keep multiple backups (across different drives, etc), and git annex sync makes the location available everywhere.



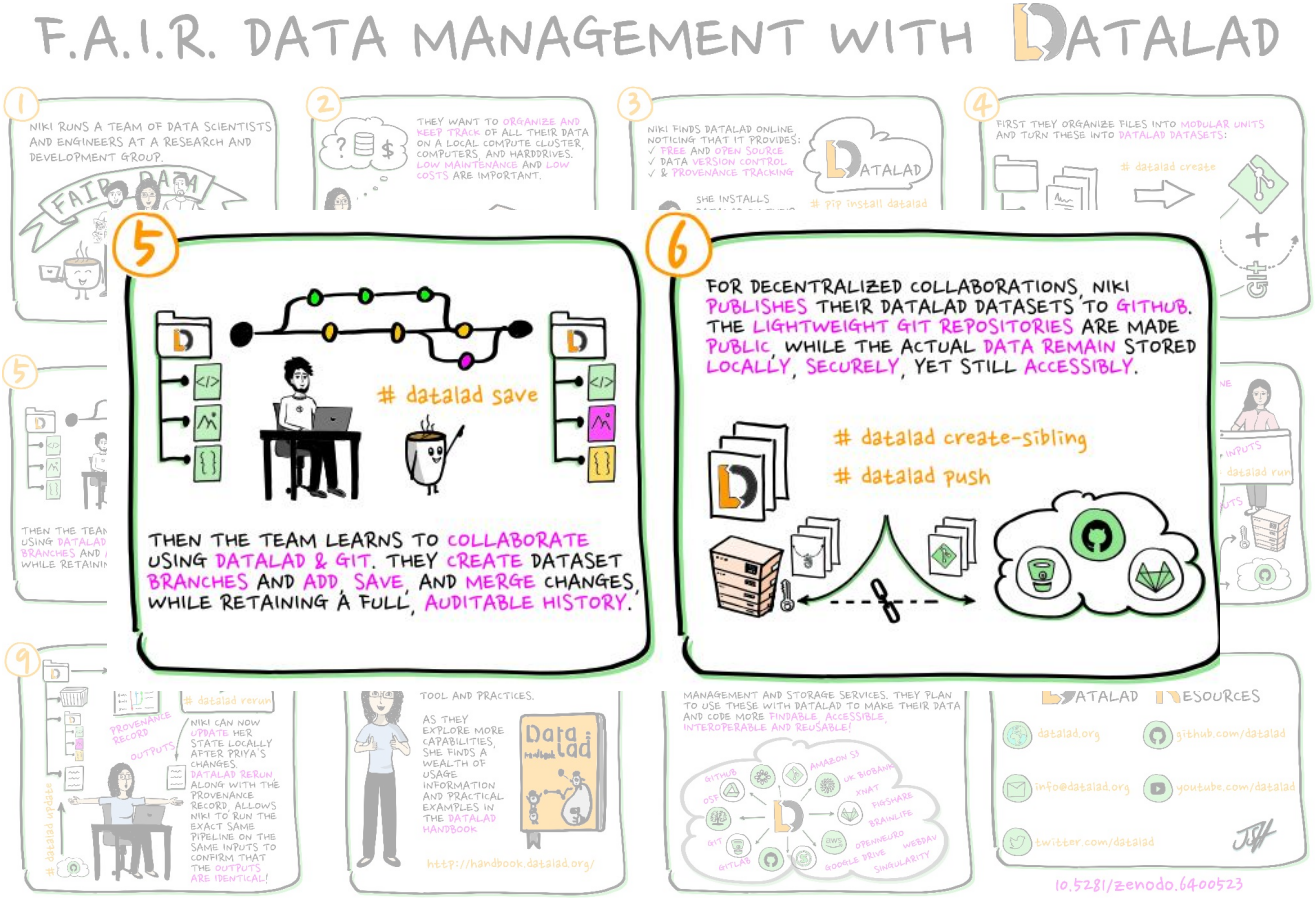
Heunis, Stephan. (2022). F.A.I.R Data Management with DataLad – a comic strip. Zenodo. https://doi.org/10.5281/zenodo.6400523



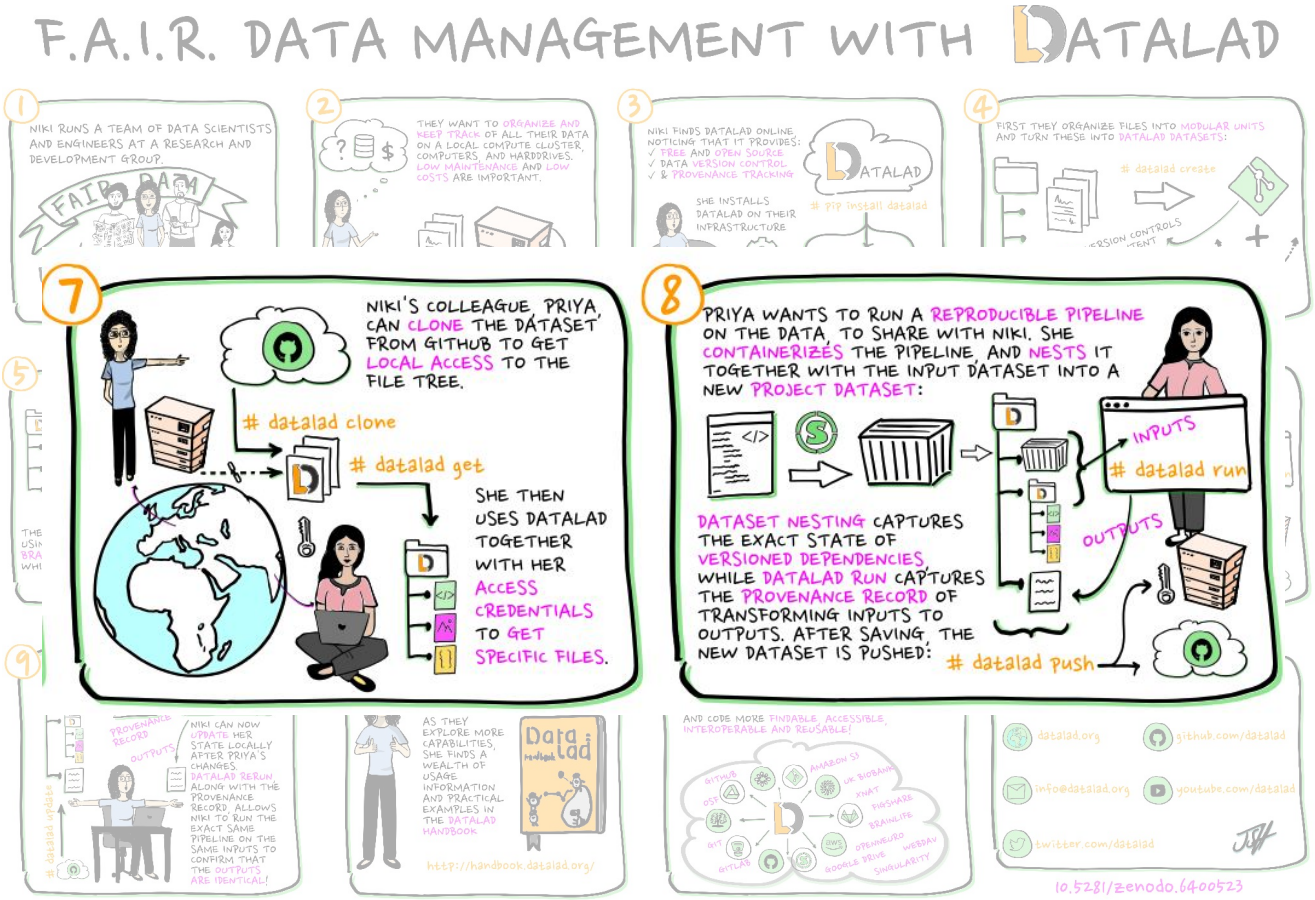
Heunis, Stephan. (2022). F.A.I.R Data Management with DataLad – a comic strip. Zenodo. https://doi.org/10.5281/zenodo.6400523



Heunis, Stephan. (2022). F.A.I.R Data Management with DataLad – a comic strip. Zenodo. https://doi.org/10.5281/zenodo.6400523

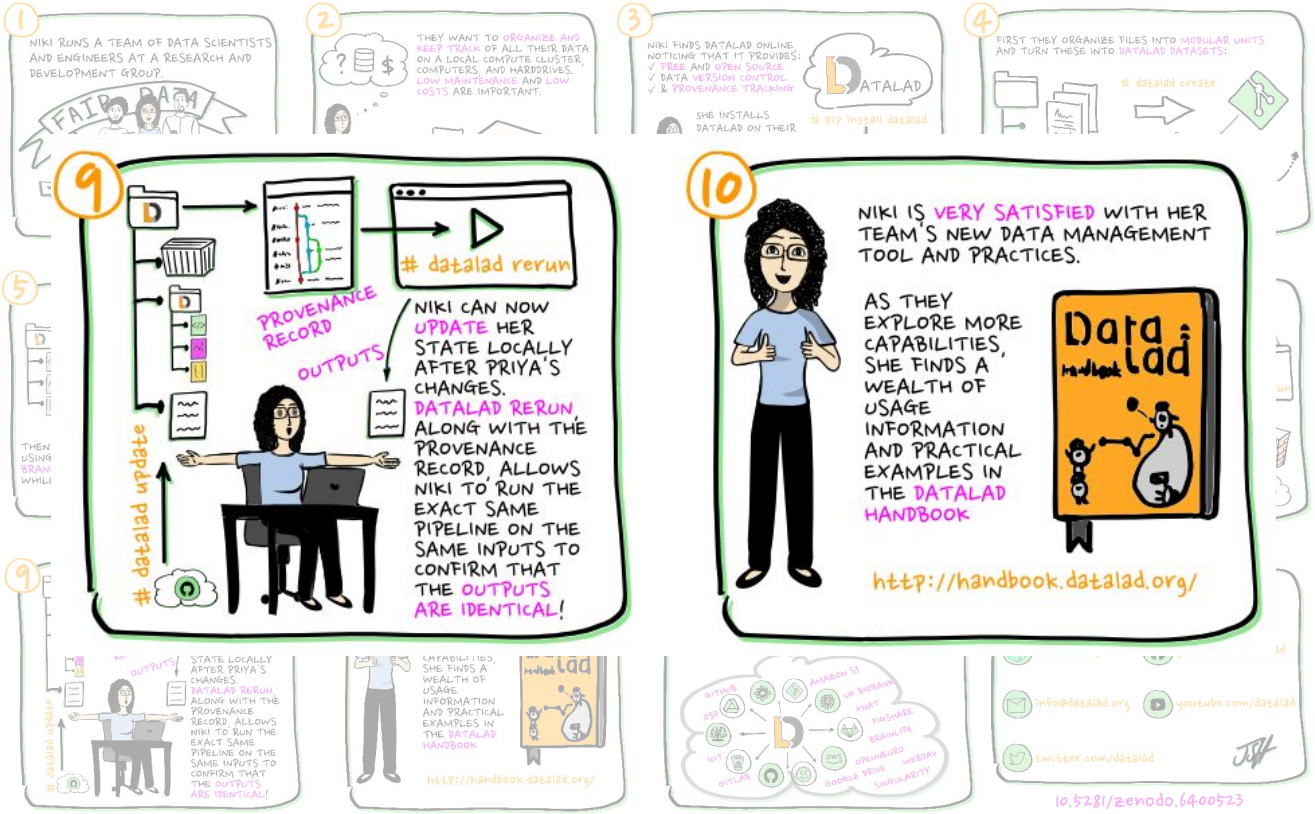


Heunis, Stephan. (2022). F.A.I.R Data Management with DataLad – a comic strip. Zenodo. https://doi.org/10.5281/zenodo.6400523

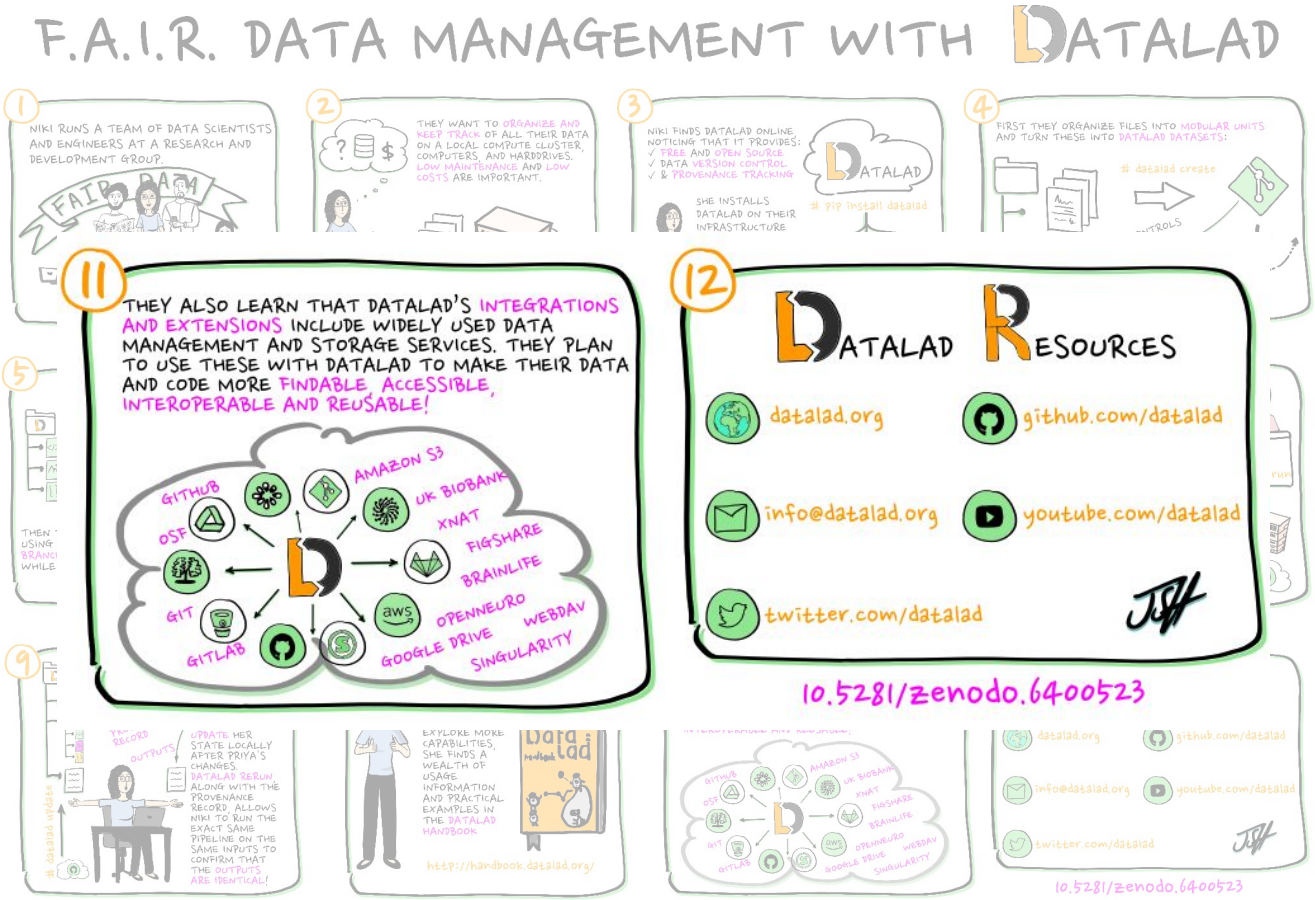


Heunis, Stephan. (2022). F.A.I.R Data Management with DataLad – a comic strip. Zenodo. https://doi.org/10.5281/zenodo.6400523

F.A.I.R. DATA MANAGEMENT WITH DATALAD

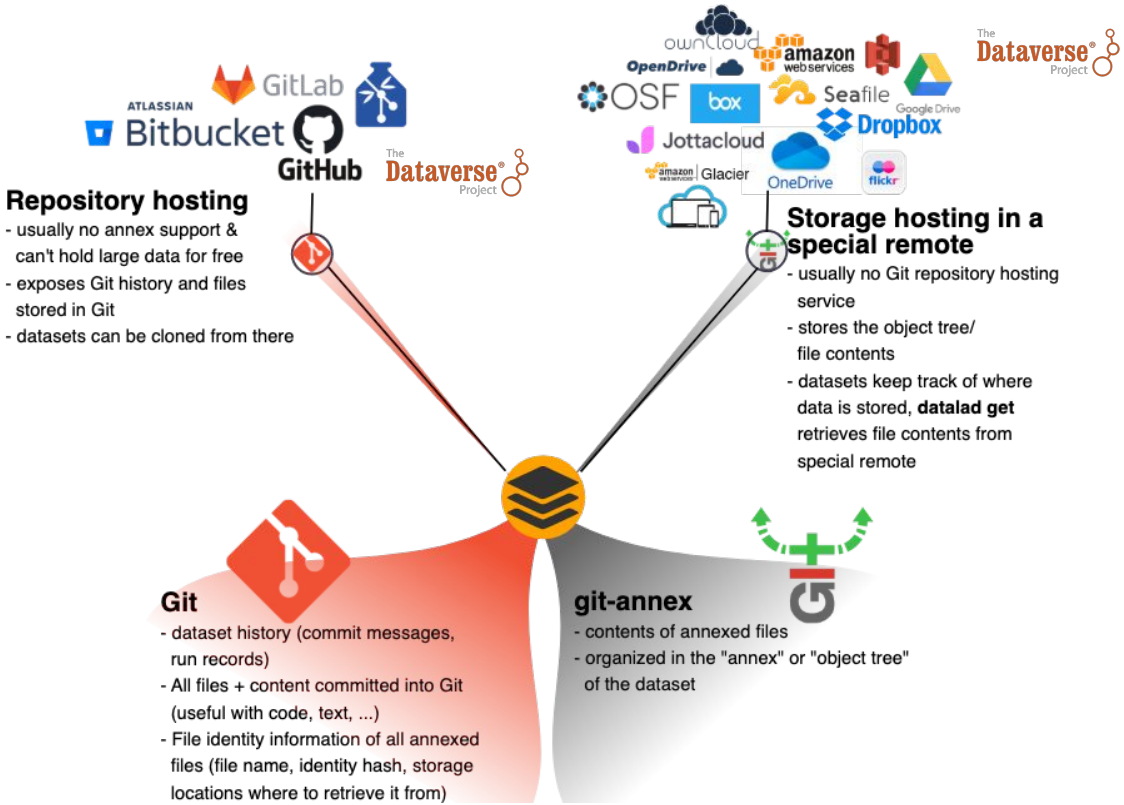


Heunis, Stephan. (2022). F.A.I.R Data Management with DataLad – a comic strip. Zenodo. https://doi.org/10.5281/zenodo.6400523



Heunis, Stephan. (2022). F.A.I.R Data Management with DataLad – a comic strip. Zenodo. https://doi.org/10.5281/zenodo.6400523

Where does Datalad and Dataverse stand in our setup?



Dealing with large and small files in Datalad

Small files like code, text, etc.

- Version controlled using `git`.
- File contents are available directly when we do `datalad clone`

Large Files like data files, binary files, etc

- Version Control handled by git-annex.
- This files are stored as shortcuts/symlinks and when we `datalad clone` **only the symlinks are retrieved.**
- The file contents are stored in some storage repository and can only be retrieved using `datalad get`.

Getting Started with Datalad

Creating a Datalad dataset and committing changes

- **Datalad Dataset :** Core data structure of a datalad (a directory)
- Create a datalad dataset

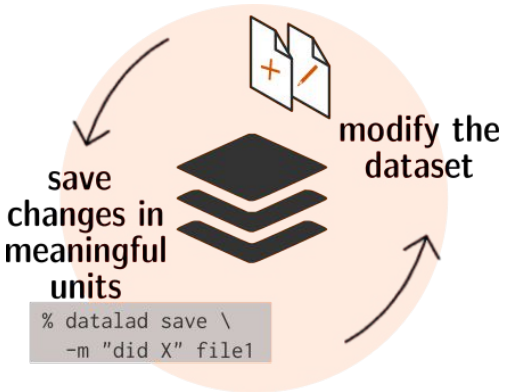
```
datalad create -c text2git datalad-test
```

This creates a .git and .datalad directories, useful for version controlling our data and code.

- Populate it with some data and add it to version control

```
datalad save -m "added two random data pdfs" <filename>
```

Similar to git add and git commit commands.



Getting Started with Datalad

Cloning a datalad dataset for our use

- Cloning from local path/server/dataset repositories

```
datalad clone -d . https://github.com/OpenNeuroDatasets/ds005346.git MEG/recordings/
```

or

```
datalad clone <user@server>:path/to/dataset/
```

*This commands however only retrieves **file availability metadata** and not the contents.*

- In our case we clone from Dataverse (e.g FreeViewingMSCOCO)

```
datalad clone \
'datalad-annex::?type=external&externaltype=dataverse&encryption=none&exporttree=no&url
=https://darus.uni-stuttgart.de&doi=DATASET-DOI' my-dataset
```

Getting Started with Datalad

Retrieving and Deleting the actual content

- To retrieve the actual content (annexed files from local or server) we need to get it via the datalad get command

```
datalad get <path/to/file/in/the/clone/repository>
```

Datalad or git annex gives us access to all data, allowing us to only retrieve what we need.

- To remove the content and again just keep the symlink from working directory

```
datalad drop <path/to/file/in/the/clone/repository>
```

Only drop data which we know has a origin (e.g installed datasets)

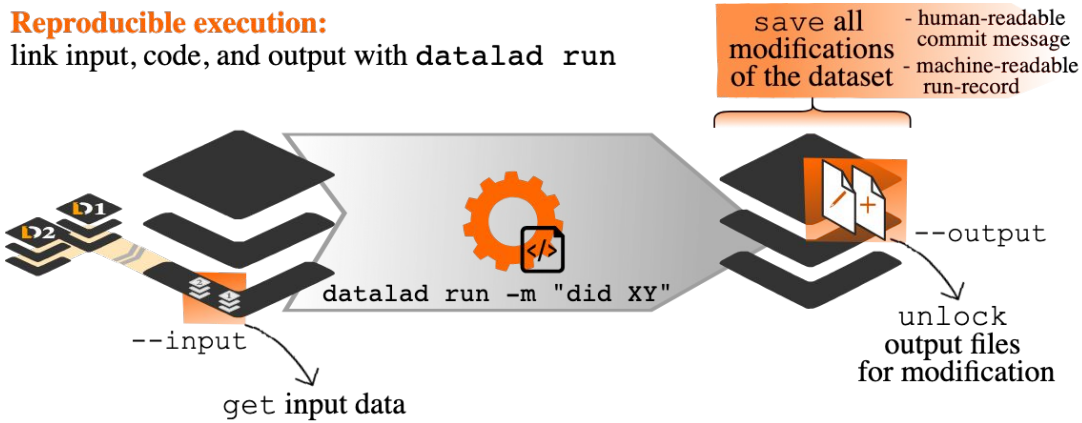
Getting started with Datalad

Bonus 🐙

To link the current state of the datalad dataset to the current output

```
datalad run -m “<commit_message>” -i ‘data/sample.csv’ -o ‘out/sample_pic.jpg’
‘code/script.sh’

datalad rerun <commit-hash>     [To reproduce some results]
```



Getting started with Datalad

Pushing/Pulling changes from a linked Datalad dataset

- Siblings in datalad are **linked clones** of a dataset (*Say we want to host our dataset in Dataverse*)

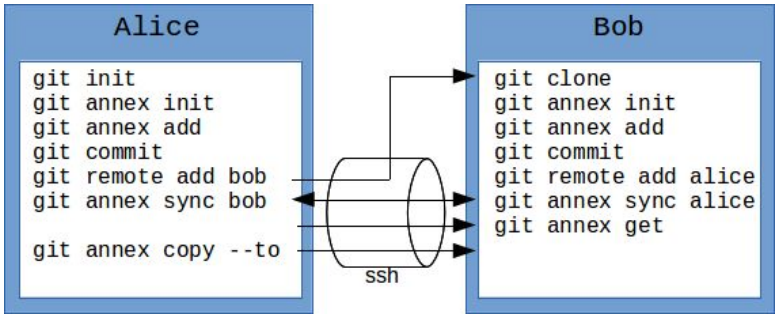
  `datalad add-sibling-dataverse <dataverse-url> <dataverse-doi>`

- Publishing/ Pushing a dataset

  `datalad push —-to <sibling>`

- Pulling changes

  `datalad update --merge` **



** `datalad update` only fetches the changes (without applying it to the current state). To apply it to the current state and see the changes we do `datalad update --how=merge`.

More information : https://handbook.datalad.org/en/latest/

Online Data Repositories

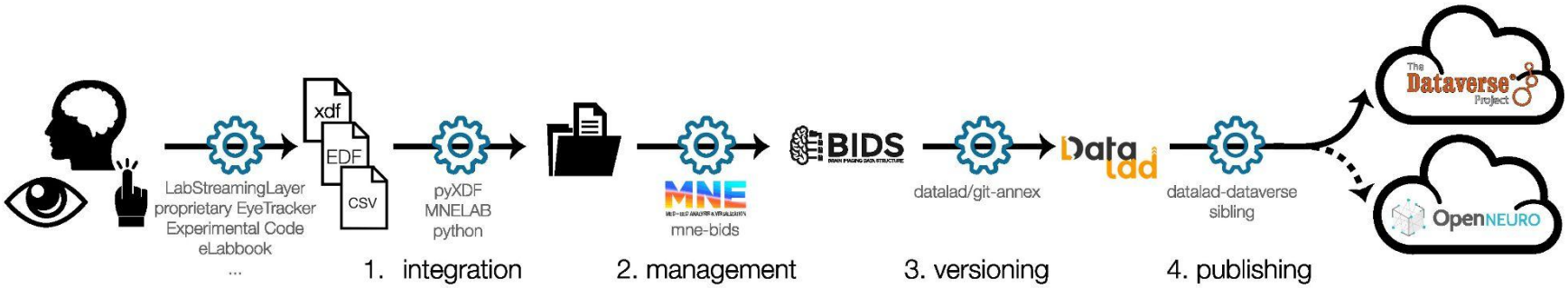
- Open Neuro hosts a lot of datalad compatible datasets : https://github.com/OpenNeuroDatasets/
- Dataverse Collections also hosts a lot of datasets. For example: DaRUS Dataverse from University of Stuttgart

.....

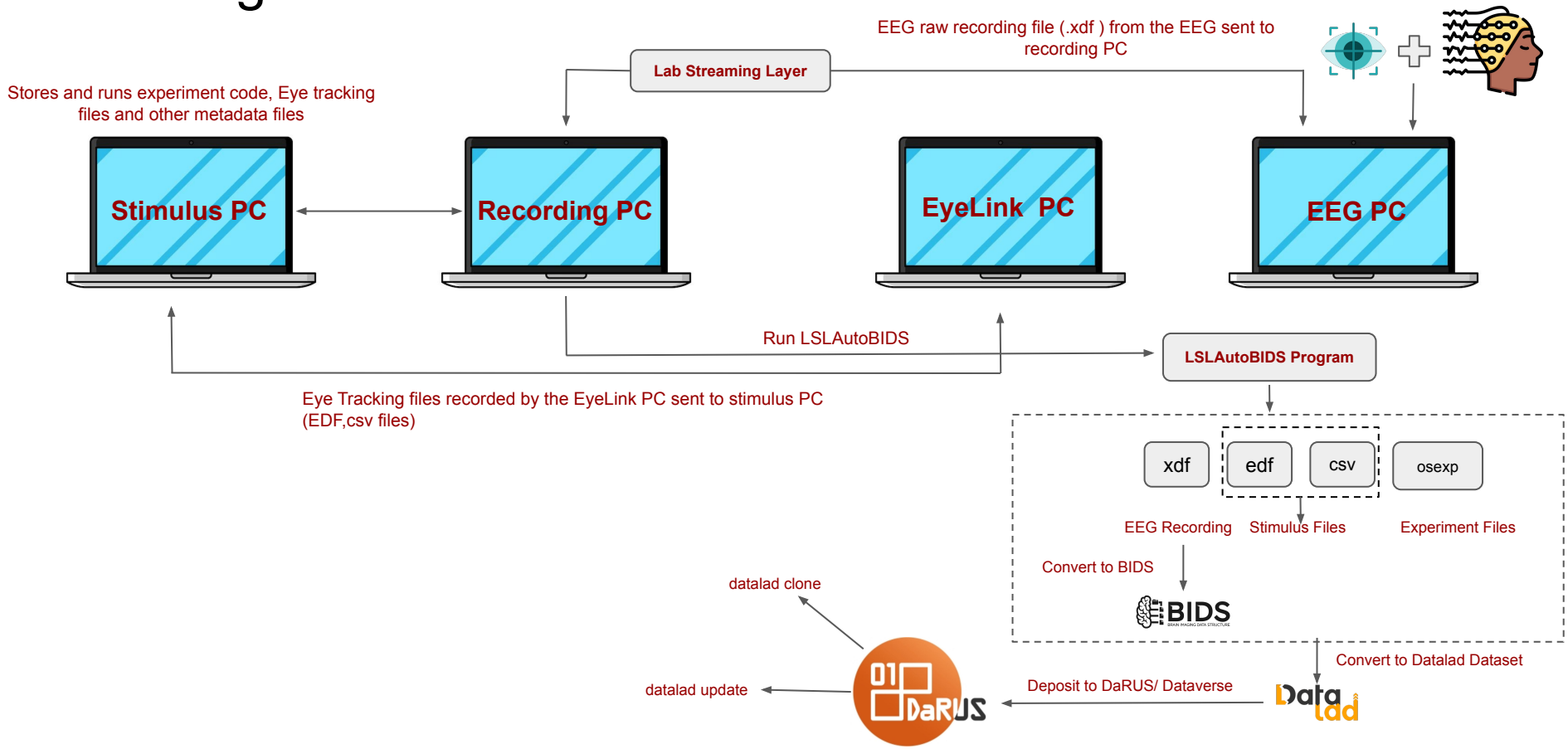
The Dataverse® Project

- Open source research data repository
- Platform to *share*, *cite*, *archive*, *download and explore* scientific research data.
- 100+ Universities/Research Institutions use Dataverse as their data repository.
- Over 65000 published datasets for open science.
- Obtain DOI upon publishing.

LSLAutoBIDS using Datalad and Dataverse in our Lab



Working of LSLAutoBIDS



Demo Time