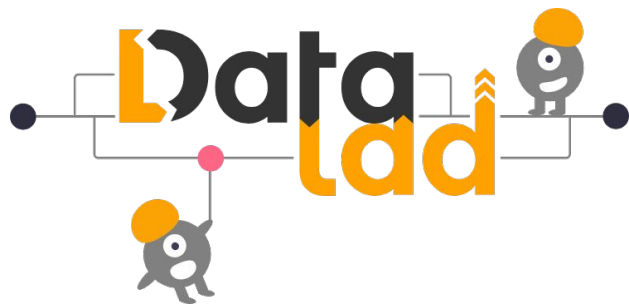
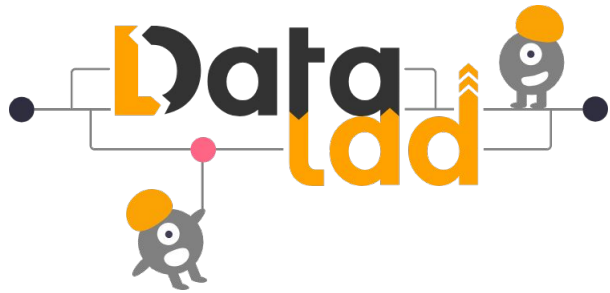


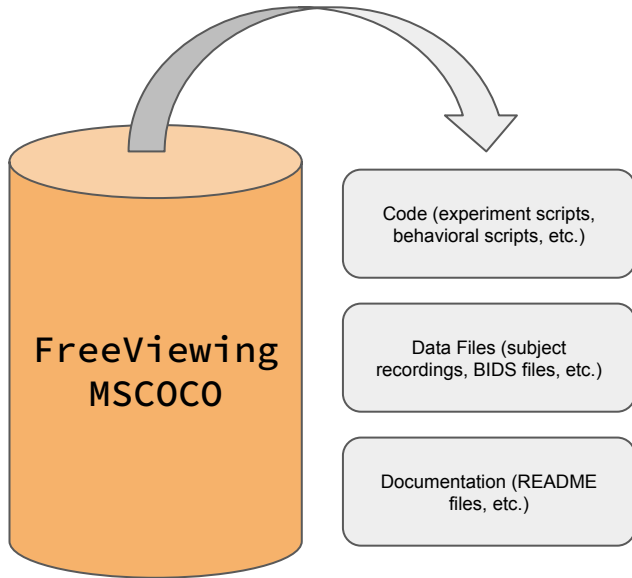
A Gentle Intro to Datalad and LSLAutoBIDS





- **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



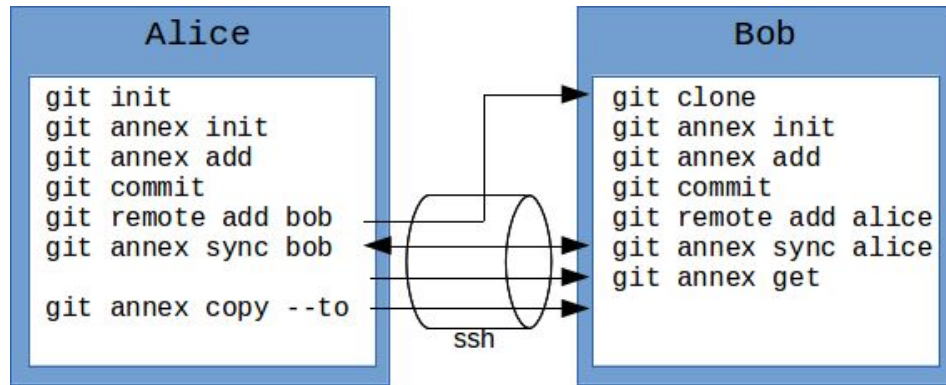
Version control system suitable for text, code, documentation etc.



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).



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.

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

1 NIKI RUNS A TEAM OF DATA SCIENTISTS AND ENGINEERS AT A RESEARCH AND DEVELOPMENT GROUP.

2 THEY WANT TO ORGANIZE AND KEEP TRACK OF ALL THEIR DATA ON A LOCAL COMPUTE CLUSTER, COMPUTERS, AND HARDDRIVES. LOW MAINTENANCE AND LOW COSTS ARE IMPORTANT.

3 NIKI FINDS DATALAD ONLINE, NOTICING THAT IT PROVIDES:
 ✓ FREE AND OPEN SOURCE
 ✓ DATA VERSION CONTROL
 ✓ & PROVENANCE TRACKING

SHE INSTALLS DATALAD ON THEIR INFRASTRUCTURE

pip install datalad

4 FIRST THEY ORGANIZE FILES INTO MODULAR UNITS AND TURN THESE INTO DATALAD DATASETS:

datalad create

GIT: VERSION CONTROLS DATASET CONTENT

GIT-ANNEX: MANAGES LARGE FILE STORAGE AND ACCESS

5 THEN THE TEAM LEARNS TO COLLABORATE USING DATALAD & GIT. THEY CREATE DATASET BRANCHES AND ADD, SAVE, AND MERGE CHANGES, WHILE RETAINING A FULL, AUDITABLE HISTORY.

datalad save

6 FOR DECENTRALIZED COLLABORATIONS, NIKI PUBLISHES THEIR DATALAD DATASETS TO GITHUB. THE LIGHTWEIGHT GIT REPOSITORIES ARE MADE PUBLIC, WHILE THE ACTUAL DATA REMAINS STORED LOCALLY, SECURELY, YET STILL ACCESSIBLY.

datalad create-sibling
 # datalad push

7 NIKI'S COLLEAGUE, PRIYA, CAN CLONE THE DATASET FROM GITHUB TO GET LOCAL ACCESS TO THE FILE TREE.

datalad clone
 # datalad get

SHE THEN USES DATALAD TOGETHER WITH HER ACCESS CREDENTIALS TO GET SPECIFIC FILES.

8 PRIYA WANTS TO RUN A REPRODUCIBLE PIPELINE ON THE DATA, TO SHARE WITH NIKI. SHE CONTAINERIZES THE PIPELINE, AND NESTS IT TOGETHER WITH THE INPUT DATASET INTO A NEW PROJECT DATASET:

DATASET NESTING CAPTURES THE EXACT STATE OF VERSIONED DEPENDENCIES WHILE DATALAD RUN CAPTURES THE PROVENANCE RECORD OF TRANSFORMING INPUTS TO OUTPUTS, AFTER SAVING, THE NEW DATASET IS PUSHED:

datalad push

INPUTS

OUTPUTS

9 NIKI CAN NOW UPDATE HER STATE LOCALLY AFTER PRIYA'S CHANGES. DATALAD RE-RUN, ALONG WITH THE PROVENANCE RECORD, ALLOWS NIKI TO RUN THE EXACT SAME PIPELINE ON THE SAME INPUTS TO CONFIRM THAT THE OUTPUTS ARE IDENTICAL!

datalad rerun

PROVENANCE RECORD

OUTPUTS

datalad update

10 NIKI IS VERY SATISFIED WITH HER TEAM'S NEW DATA MANAGEMENT TOOL AND PRACTICES. AS THEY EXPLORE MORE CAPABILITIES, SHE FINDS A WEALTH OF USAGE INFORMATION AND PRACTICAL EXAMPLES IN THE DATALAD HANDBOOK.

<http://handbook.datalad.org/>

11 THEY ALSO LEARN THAT DATALAD'S INTEGRATIONS AND EXTENSIONS INCLUDE WIDELY USED DATA MANAGEMENT AND STORAGE SERVICES. THEY PLAN TO USE THESE WITH DATALAD TO MAKE THEIR DATA AND CODE MORE FINDABLE, ACCESSIBLE, INTEROPERABLE AND REUSABLE!

12 DATALAD RESOURCES

datalad.org

github.com/datalad

info@datalad.org

youtube.com/datalad

twitter.com/datalad

10.5281/zenodo.6400523

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

1 NIKI RUNS A TEAM OF DATA SCIENTISTS AND ENGINEERS AT A RESEARCH AND DEVELOPMENT GROUP.

2 THEY WANT TO ORGANIZE AND KEEP TRACK OF ALL THEIR DATA ON A LOCAL COMPUTE CLUSTER, COMPUTERS, AND HARDDRIVES. LOW MAINTENANCE AND LOW COSTS ARE IMPORTANT.

3 NIKI FINDS DATALAD ONLINE NOTICING THAT IT PROVIDES:
 ✓ FREE AND OPEN SOURCE
 ✓ DATA VERSION CONTROL
 ✓ & PROVENANCE TRACKING

SHE INSTALLS DATALAD ON THEIR INFRASTRUCTURE

```
# pip install datalad
```

4 FIRST THEY ORGANIZE FILES INTO MODULAR UNITS AND TURN THESE INTO DATALAD DATASETS:

```
# datalad create
```

CONTROLS

5 THEN THE USING DAT BRANCHES WHILE REI

6 EXPLORE MORE CAPABILITIES, SHE FINDS A WEALTH OF USAGE INFORMATION AND PRACTICAL EXAMPLES IN THE DATALAD HANDBOOK

<http://handbook.datalad.org/>

7 INTEROPERABLE AND REUSABLE!

AMAZON S3
 PRIVATE
 PROSHARE
 BEANHOLE
 GITHUB
 BITBUCKET
 GIT
 BITLARK
 BOBBER VINE
 WEBDAV
 SINGULARITY

8 NIKI WOULD NOW UPDATE HER STATE LOCALLY AFTER PRIYA'S CHANGES. DATALAD REUN ALONG WITH THE PROVENANCE RECORD, ALLOWS NIKI TO RUN THE EXACT SAME PIPELINE ON THE SAME INPUTS TO CONFIRM THAT THE OUTPUTS ARE IDENTICAL!

9

social media links:
datalad.org
github.com/datalad
info@datalad.org
youtube.com/datalad
twitter.com/datalad

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

1 NIKI RUNS A TEAM OF DATA SCIENTISTS AND ENGINEERS AT A RESEARCH AND DEVELOPMENT GROUP.

2 THEY WANT TO ORGANIZE AND KEEP TRACK OF ALL THEIR DATA ON A LOCAL COMPUTE CLUSTER, COMPUTERS, AND HARDDRIVES. LOW MAINTENANCE AND LOW COSTS ARE IMPORTANT.

3 NIKI FINDS DATALAD ONLINE NOTICING THAT IT PROVIDES:
 ✓ FREE AND OPEN SOURCE
 ✓ DATA VERSION CONTROL
 ✓ & PROVENANCE TRACKING

SHE INSTALLS DATALAD ON THEIR INFRASTRUCTURE
`# pip install datalad`

4 FIRST THEY ORGANIZE FILES INTO MODULAR UNITS AND TURN THESE INTO DATALAD DATASETS:
`# datalad create`

`git: VERSION CONTROLS DATASET CONTENT`
`git-ANNEX: MANAGES LARGE FILE STORAGE AND ACCESS`

5 NIKI CAN NOW UPDATE HER STATE LOCALLY AFTER PRIVA'S CHANGES. DATALAD REBORN ALONG WITH THE PROVENANCE RECORD, ALLOWS NIKI TO RUN THE EXACT SAME PIPELINE ON THE SAME INPUTS TO CONFIRM THAT THE OUTPUTS ARE IDENTICAL!

AS THEY EXPLORE MORE CAPABILITIES, SHE FINDS A WEALTH OF USAGE INFORMATION AND PRACTICAL EXAMPLES IN THE DATALAD HANDBOOK.
<http://handbook.datalad.org/>

TO USE THESE WITH DATALAD TO MAKE THEIR DATA AND CODE MORE FINDABLE, ACCESSIBLE, INTEROPERABLE AND REUSABLE!

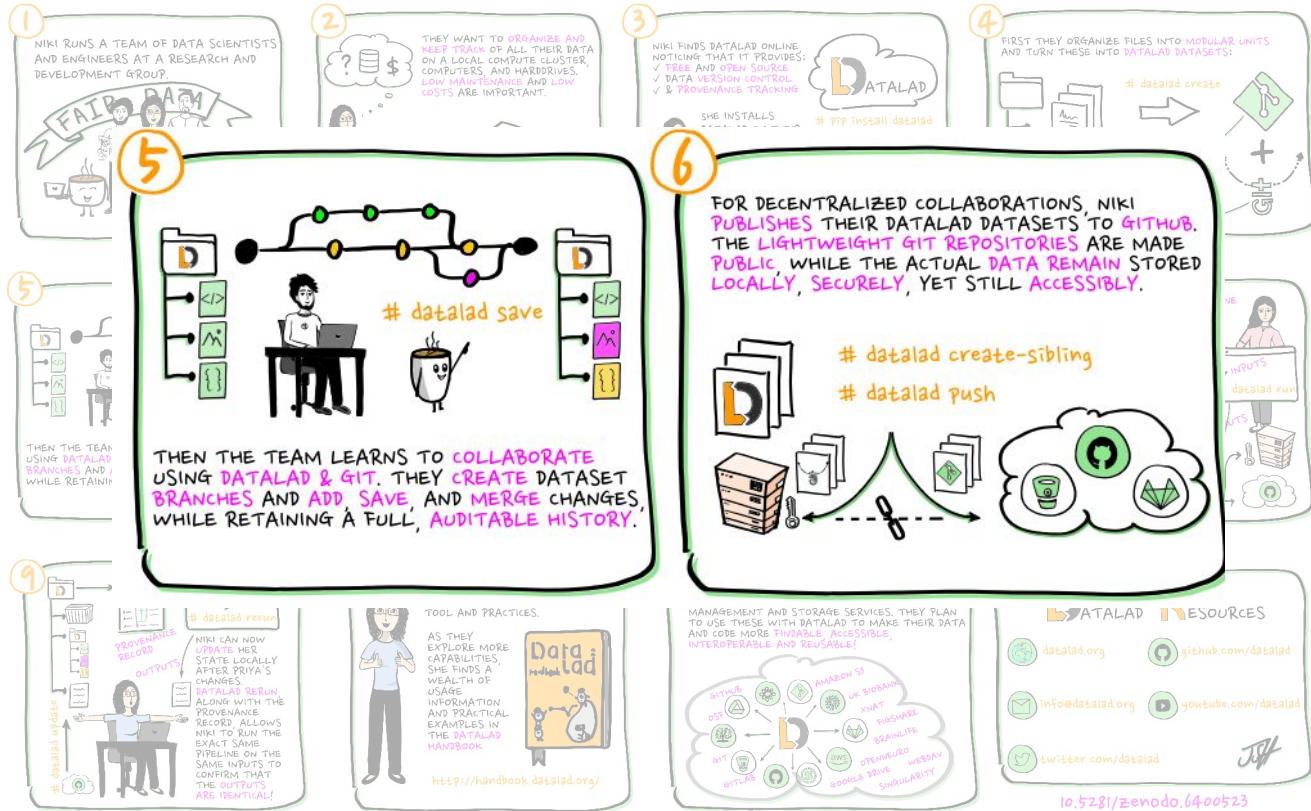
datalad.org
github.com/datalad
info@datalad.org
[youtube.com/datalad](https://www.youtube.com/datalad)
twitter.com/datalad

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>

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

1 NIKI RUNS A TEAM OF DATA SCIENTISTS AND ENGINEERS AT A RESEARCH AND DEVELOPMENT GROUP.

2 THEY WANT TO ORGANIZE AND KEEP TRACK OF ALL THEIR DATA ON A LOCAL COMPUTE CLUSTER, COMPUTERS, AND HARDDRIVES. FREE AND OPEN SOURCE, DATA VERSION CONTROL, & PROVENANCE TRACKING ARE IMPORTANT.

3 NIKI FINDS DATALAD ONLINE, NOTICING THAT IT PROVIDES: FREE AND OPEN SOURCE, DATA VERSION CONTROL, & PROVENANCE TRACKING. SHE INSTALLS DATALAD ON THEIR INFRASTRUCTURE.

4 FIRST THEY ORGANIZE FILES INTO MODULAR UNITS AND TURN THESE INTO DATALAD DATASETS.

5 NIKI'S COLLEAGUE, PRIYA, CAN CLONE THE DATASET FROM GITHUB TO GET LOCAL ACCESS TO THE FILE TREE.

6 SHE THEN USES DATALAD TOGETHER WITH HER ACCESS CREDENTIALS TO GET SPECIFIC FILES.

7 PRIYA WANTS TO RUN A REPRODUCIBLE PIPELINE ON THE DATA, TO SHARE WITH NIKI. SHE CONTAINERIZES THE PIPELINE, AND NESTS IT TOGETHER WITH THE INPUT DATASET INTO A NEW PROJECT DATASET:

8 DATASET NESTING CAPTURES THE EXACT STATE OF VERSIONED DEPENDENCIES, WHILE DATALAD RUN CAPTURES THE PROVENANCE RECORD OF TRANSFORMING INPUTS TO OUTPUTS. AFTER SAVING, THE NEW DATASET IS PUSHED:

9 NIKI CAN NOW UPDATE HER STATE LOCALLY AFTER PRIYA'S CHANGES. DATALAD RERUN ALONG WITH THE PROVENANCE RECORD, ALLOWS NIKI TO RERUN THE EXACT SAME PIPELINE ON THE SAME INPUTS TO CONFIRM THAT THE OUTPUTS ARE IDENTICAL!

AS THEY EXPLORE MORE CAPABILITIES, SHE FINDS A WEALTH OF USAGE INFORMATION AND PRACTICAL EXAMPLES IN THE DATALAD HANDBOOK.

AND CODE MORE FINDABLE, ACCESSIBLE, INTEROPERABLE AND REUSABLE!

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

1 NIKI RUNS A TEAM OF DATA SCIENTISTS AND ENGINEERS AT A RESEARCH AND DEVELOPMENT GROUP.

2 THEY WANT TO ORGANIZE AND KEEP TRACK OF ALL THEIR DATA ON A LOCAL COMPUTE CLUSTER, COMPUTERS, AND HARDDRIVES. LOW MAINTENANCE AND LOW COSTS ARE IMPORTANT.

3 NIKI FINDS DATALAD ONLINE, NOTICING THAT IT PROVIDES:
✓ FREE AND OPEN SOURCE
✓ DATA VERSION CONTROL
✓ & PROVENANCE TRACKING

SHE INSTALLS DATALAD ON THEIR # pip install datalad

4 FIRST THEY ORGANIZE FILES INTO MODULAR UNITS AND TURN THESE INTO DATALAD DATASETS:
datalad create

5 THEN USING BRIAN WHILL

6 NIKI CAN NOW UPDATE HER STATE LOCALLY AFTER PRIYA'S CHANGES. DATALAD RERUN ALONG WITH THE PROVENANCE RECORD ALLOWS NIKI TO RUN THE EXACT SAME PIPELINE ON THE SAME INPUTS TO CONFIRM THAT THE OUTPUTS ARE IDENTICAL!

7 AS THEY EXPLORE MORE CAPABILITIES, SHE FINDS A WEALTH OF USAGE INFORMATION AND PRACTICAL EXAMPLES IN THE DATALAD HANDBOOK

8

9

10 NIKI IS VERY SATISFIED WITH HER TEAM'S NEW DATA MANAGEMENT TOOL AND PRACTICES.

AS THEY EXPLORE MORE CAPABILITIES, SHE FINDS A WEALTH OF USAGE INFORMATION AND PRACTICAL EXAMPLES IN THE DATALAD HANDBOOK

<http://handbook.datalad.org/>

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

1 NIKI RUNS A TEAM OF DATA SCIENTISTS AND ENGINEERS AT A RESEARCH AND DEVELOPMENT GROUP.

2 THEY WANT TO ORGANIZE AND KEEP TRACK OF ALL THEIR DATA ON A LOCAL COMPUTE CLUSTER, COMPUTERS, AND HARDDRIVES. LOW MAINTENANCE AND LOW COSTS ARE IMPORTANT.

3 NIKI FINDS DATALAD ONLINE, NOTICING THAT IT PROVIDES:

- ✓ FREE AND OPEN SOURCE
- ✓ DATA VERSION CONTROL
- ✓ & PROVENANCE TRACKING

 SHE INSTALLS DATALAD ON THEIR INFRASTRUCTURE.


```
# pip install datalad
```

4 FIRST THEY ORGANIZE FILES INTO MODULAR UNITS AND TURN THESE INTO DATALAD DATASETS:


```
# datalad create
```

11 THEY ALSO LEARN THAT DATALAD'S INTEGRATIONS AND EXTENSIONS INCLUDE WIDELY USED DATA MANAGEMENT AND STORAGE SERVICES. THEY PLAN TO USE THESE WITH DATALAD TO MAKE THEIR DATA AND CODE MORE FINDABLE, ACCESSIBLE, INTEROPERABLE AND REUSABLE!

12 DATALAD RESOURCES

- datalad.org
- github.com/datalad
- info@datalad.org
- youtube.com/datalad
- twitter.com/datalad

[10.5281/zenodo.6400523](https://zenodo.org/record/6400523)

5 THEN USING BRANCH WHILE

9

```
# datalad update
```

 UPDATE HER STATE LOCALLY AFTER PRIVA'S CHANGES. DATALAD RE-RUN ALONG WITH THE PROVENANCE RECORD, ALLOWS NIKI TO RUN THE EXACT SAME PIPELINE ON THE SAME INPUTS TO CONFIRM THAT THE OUTPUTS ARE IDENTICAL!

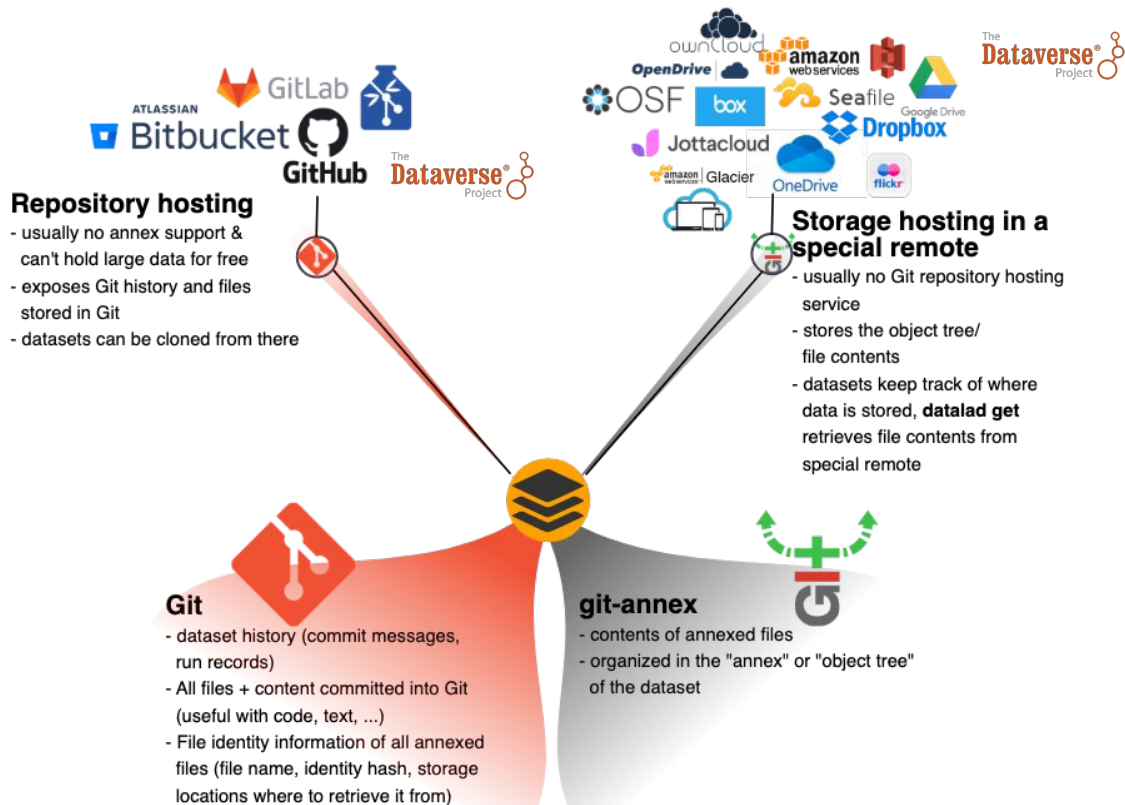
EXPLORE MORE CAPABILITIES, SHE FINDS A WEALTH OF USAGE INFORMATION AND PRACTICAL EXAMPLES IN THE DATALAD HANDBOOK.

<http://handbook.datalad.org/>

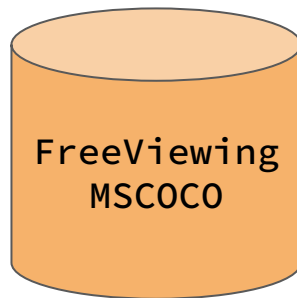
Heunis, Stephan. (2022). F.A.I.R. Data Management with DataLad – a comic strip. Zenodo.

<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.
- These 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

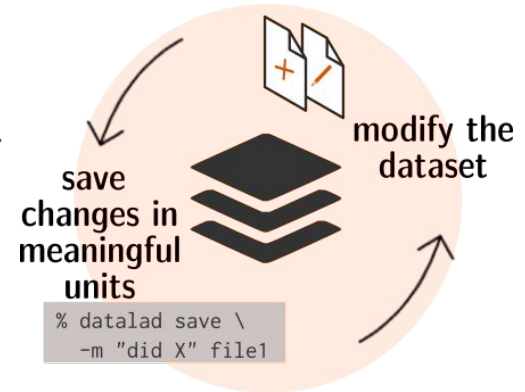
```
datalad create -c text2git datalad-test
```

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 🐙

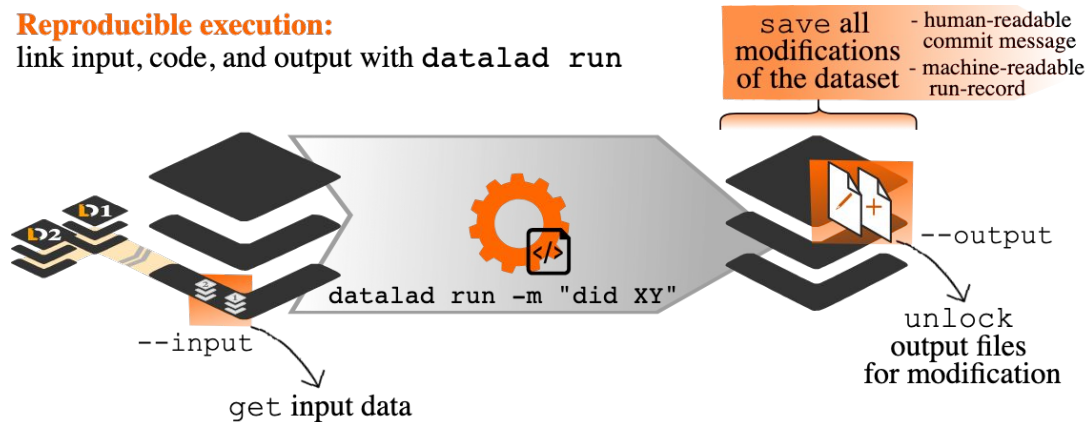
To link the current state of the datalad dataset to the current output

```
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]
```

Reproducible execution:

link input, code, and output with `datalad run`



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*)

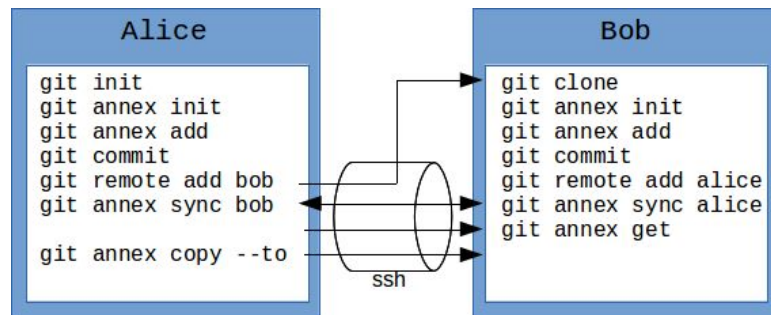
```
datalad add-sibling-dataverse <dataverse-url> <dataverse-doi>
```

- 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`.**

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

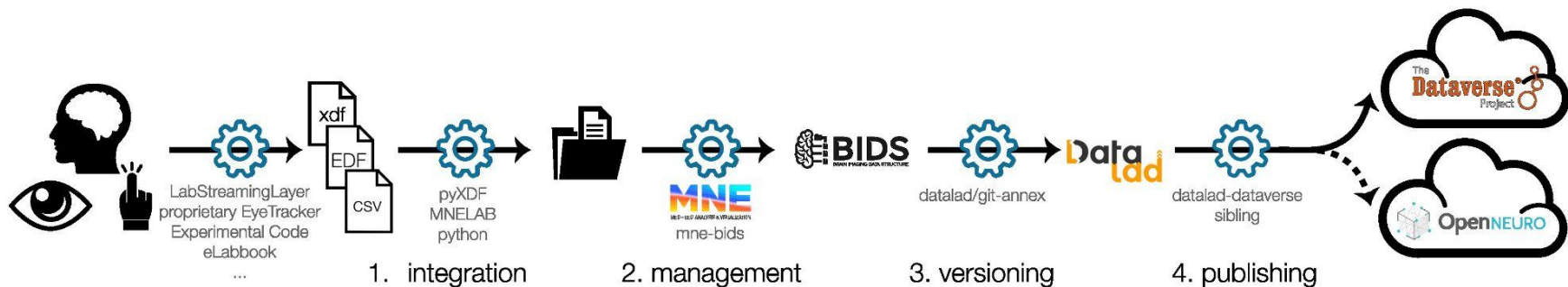
- Open source research data repository
- Platform to *share, cite, archive, download and explore* scientific research data.
- 100+ Universities/Research Institutions use Dataverse as their data repository.
- Over 65000 published datasets for open science.
- Obtain DOI upon publishing.



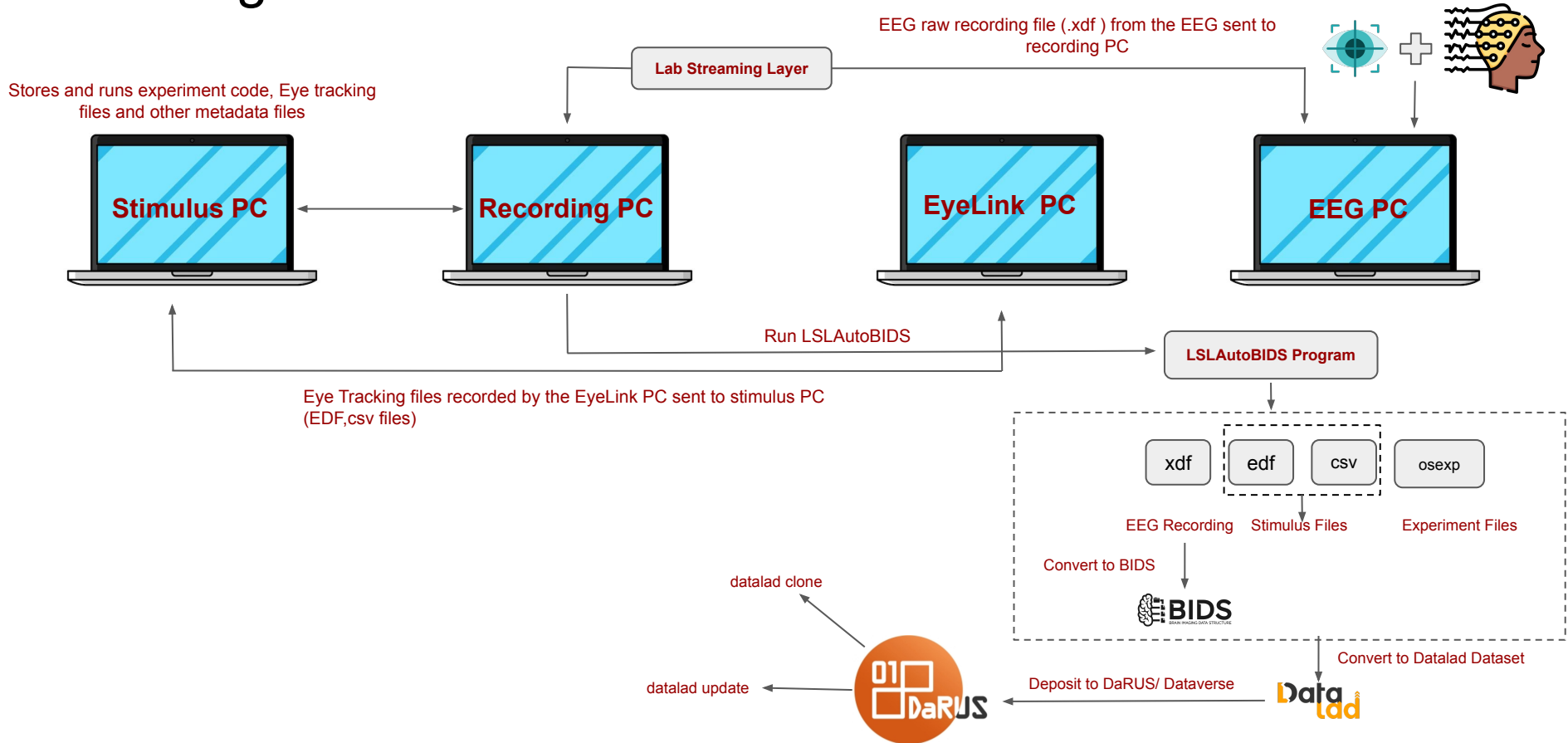
HARVARD
DATAVERSE



LSLAutoBIDS using Datalad and Dataverse in our Lab



Working of LSLAutoBIDS



Demo Time 🐙