Have a ☕!

# BEFORE WE BEGIN...

Any left-over questions from yesterday?

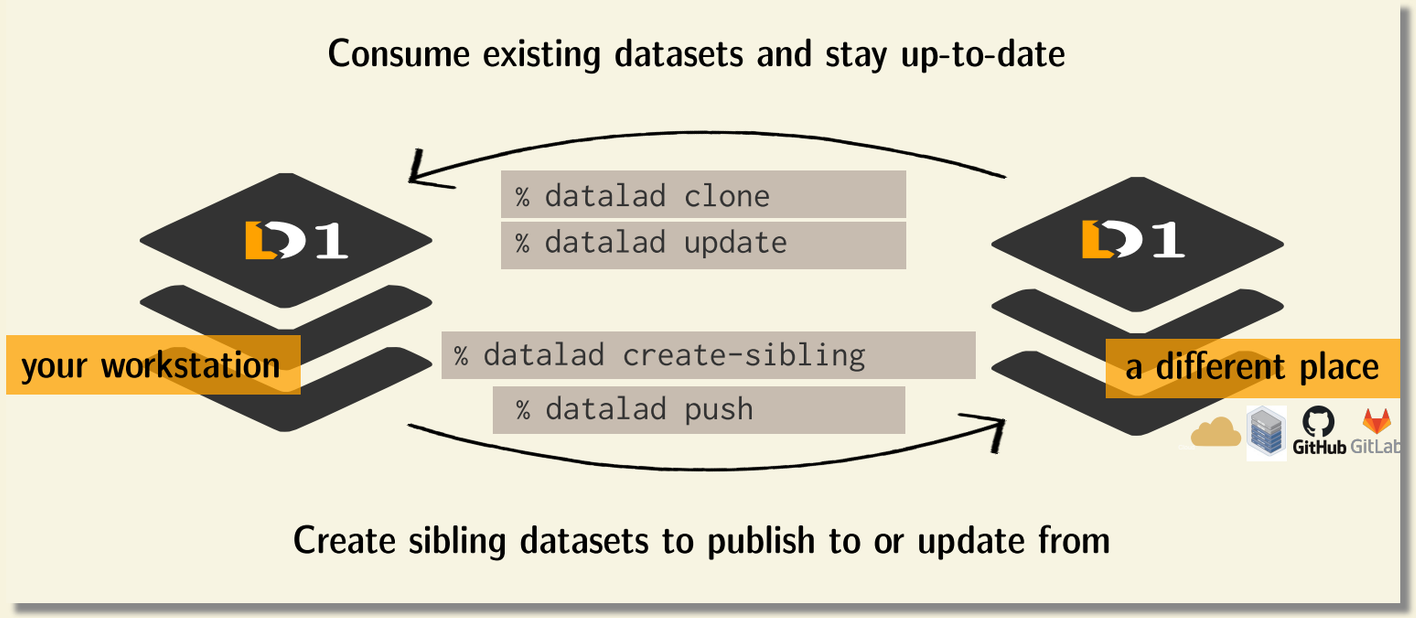
# PUBLISHING DATASETS

## How to share your work with others

Repository hosting services, siblings, and `datalad push`

# "SHARE DATA LIKE SOURCE CODE"

- Datasets can be cloned, pushed, and updated from and to **local** and **remote** paths, remote **hosting services**, external **special remotes**




- Examples:
  Local path

```
../my-projects/experiment_data
```

  Remote path

```
myuser@myinstitutes.hcp.system:/home/myuser/my-projects/experiment_data
```

  Hosting service

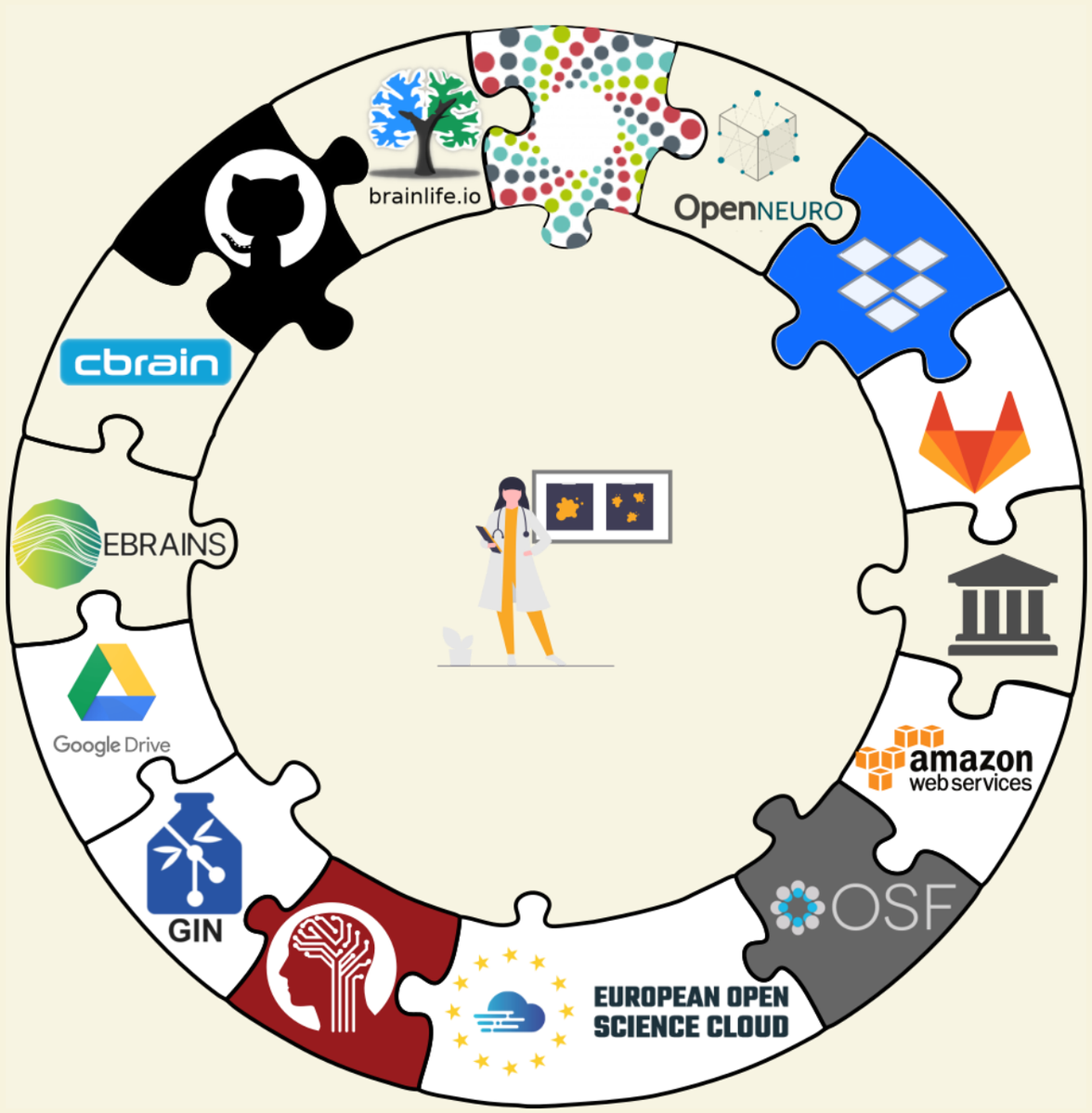
```
git.github.com:myuser/experiment_data.git
```

  External special remotes

```
osf://my-osf-project-id
```

# INTEROPERABILITY

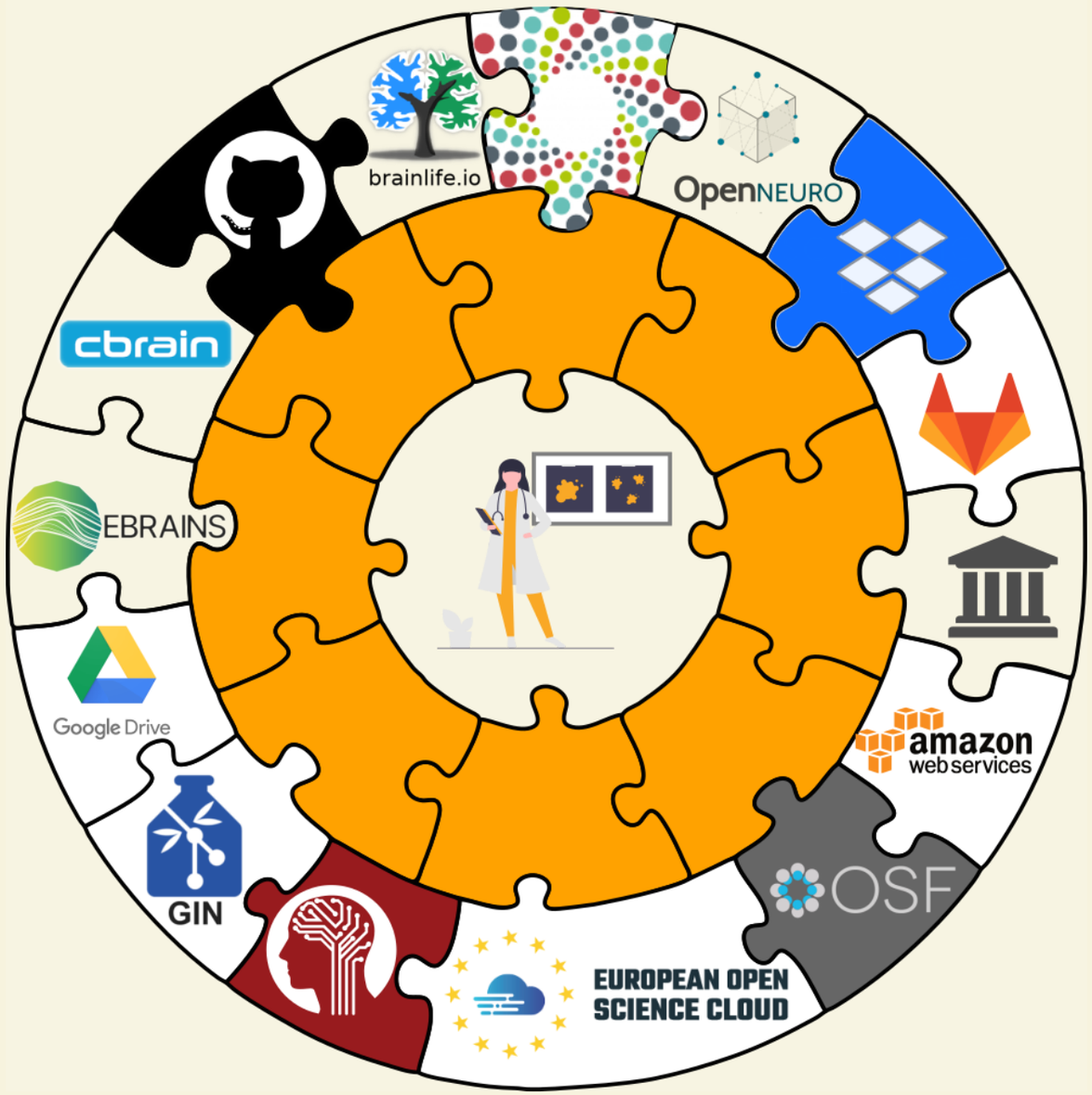
- DataLad is built to maximize interoperability and use with hosting and storage technology




See the chapter Third party infrastructure for walk-throughs for different services

# INTEROPERABILITY

- DataLad is built to maximize interoperability and use with hosting and storage technology




See the chapter Third party infrastructure for walk-throughs for different services

# PUBLISHING DATASETS

# GLOSSARY

**Sibling (remote)**
: Linked clones of a dataset. You can usually update (from) siblings to keep all your siblings in sync (e.g., ongoing data acquisition stored on experiment compute and backed up on cluster and external hard-drive)

**Repository hosting service**
: Webservices to host Git repositories, such as GitHub, GitLab, Bitbucket, Gin, ...

**Third-party storage**
: Infrastructure (private/commercial/free/...) that can host data. A "special remote" protocol is used to publish or pull data to and from it

**Publishing datasets**
: *Pushing* dataset contents (Git and/or annex) to a sibling using **datalad push**

**Updating datasets**
: *Pulling* new changes from a sibling using **datalad update --merge**

# PUBLISHING DATASETS

# PUBLISHING DATASETS



## Repository hosting

- usually no annex support & can't hold large data for free
- exposes Git history and files stored in Git
- datasets can be cloned from there

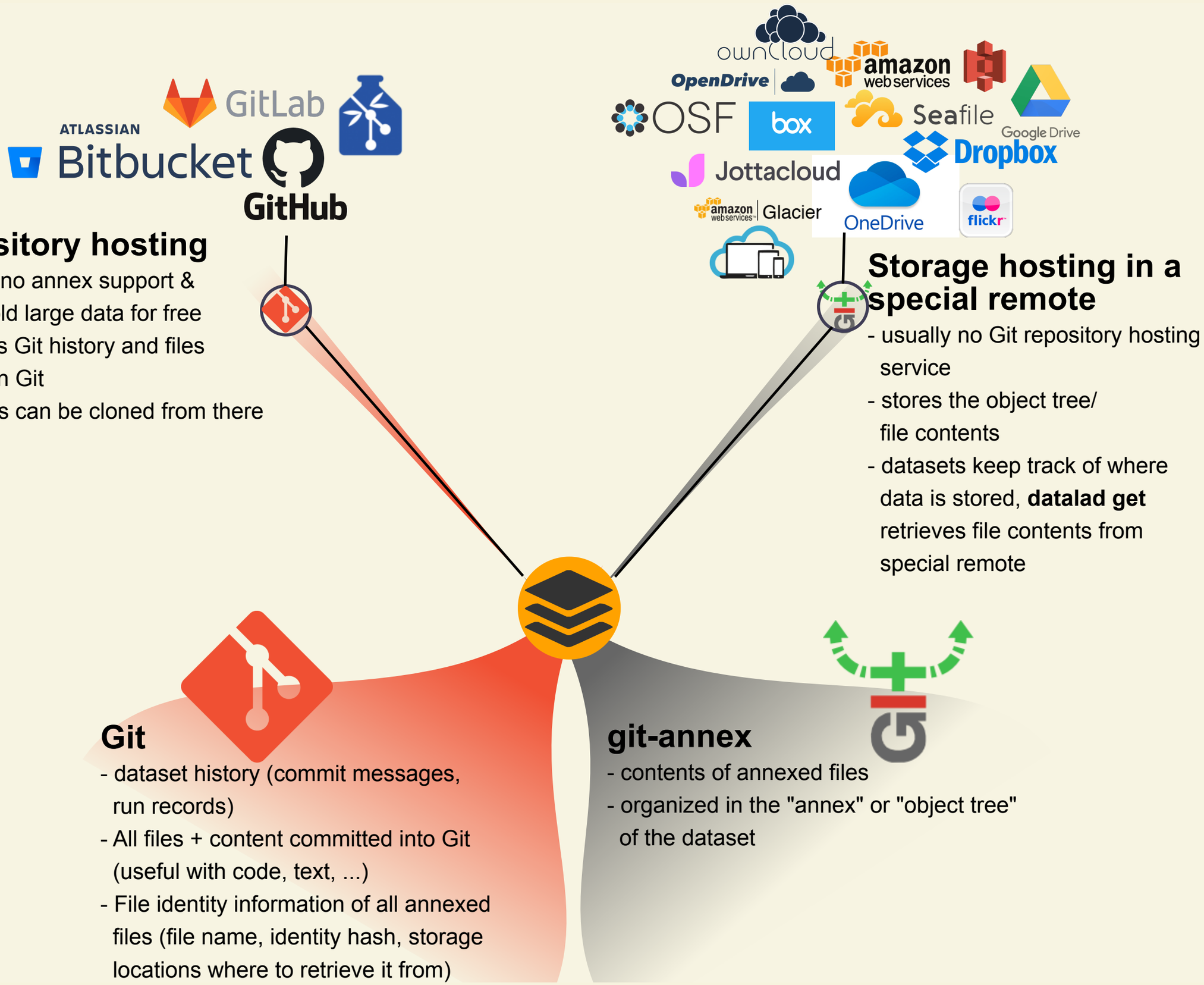
## Git

- dataset history (commit messages, run records)
- All files + content committed into Git (useful with code, text, ...)
- File identity information of all annexed files (file name, identity hash, storage locations where to retrieve it from)

## git-annex

- contents of annexed files
- organized in the "annex" or "object tree" of the dataset

# PUBLISHING DATASETS




## Repository hosting

- usually no annex support & can't hold large data for free
- exposes Git history and files stored in Git
- datasets can be cloned from there

## Storage hosting in a special remote

- usually no Git repository hosting service
- stores the object tree/ file contents
- datasets keep track of where data is stored, **datalad get** retrieves file contents from special remote

## Git

- dataset history (commit messages, run records)
- All files + content committed into Git (useful with code, text, ...)
- File identity information of all annexed files (file name, identity hash, storage locations where to retrieve it from)

## git-annex

- contents of annexed files
- organized in the "annex" or "object tree" of the dataset

# PUBLISHING DATASETS

Typical case:

- Datasets are exposed via a private or public repository on a repository hosting service
- Data can't be stored in the repository hosting service, but can be kept in almost any third party storage
- Publication dependencies automate pushing to the correct place, e.g.,

```
$ git config --local remote.github.datalad-publish-depends gdrive
# or
$ datalad siblings add --name origin --url git@git.jugit.fzj.de:adswa/experiment-data.git --publish-depends s3
```

# PUBLISHING DATASETS

# PUBLISHING DATASETS

# PUBLISHING DATASETS

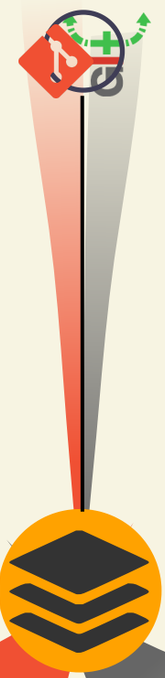
# PUBLISHING DATASETS

Special case 1: repositories with annex support



## Repositories with annex support

- examples: GIN (gin.g-node.org), GitLab instances with enabled annex support
- can hold large data for free
- exposes Git history and all files + content
- datasets can be cloned from there



## Git

- dataset history (commit messages, run records)
- All files + content committed into Git (useful with code, text, ...)
- File identity information of all annexed files (file name, identity hash, storage locations where to retrieve it from)

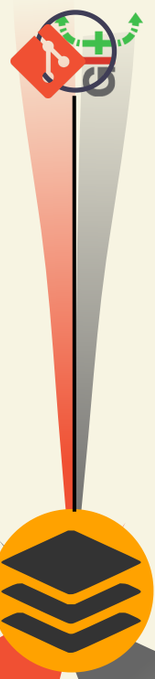
## git-annex

- contents of annexed files
- organized in the "annex" or "object tree" of the dataset

# PUBLISHING DATASETS

Special case 2: Special remotes with repositories





**Publishing both components to a special remote**

- performed with DataLad extensions (e.g., datalad-osf), or helpers (e.g., git-remote-rclone)
- publishes the Git repository (in two files) in conjunction with annexed data
- allows cloning from a special remote

**Git**

- dataset history (commit messages, run records)
- All files + content committed into Git (useful with code, text, ...)
- File identity information of all annexed files (file name, identity hash, storage locations where to retrieve it from)

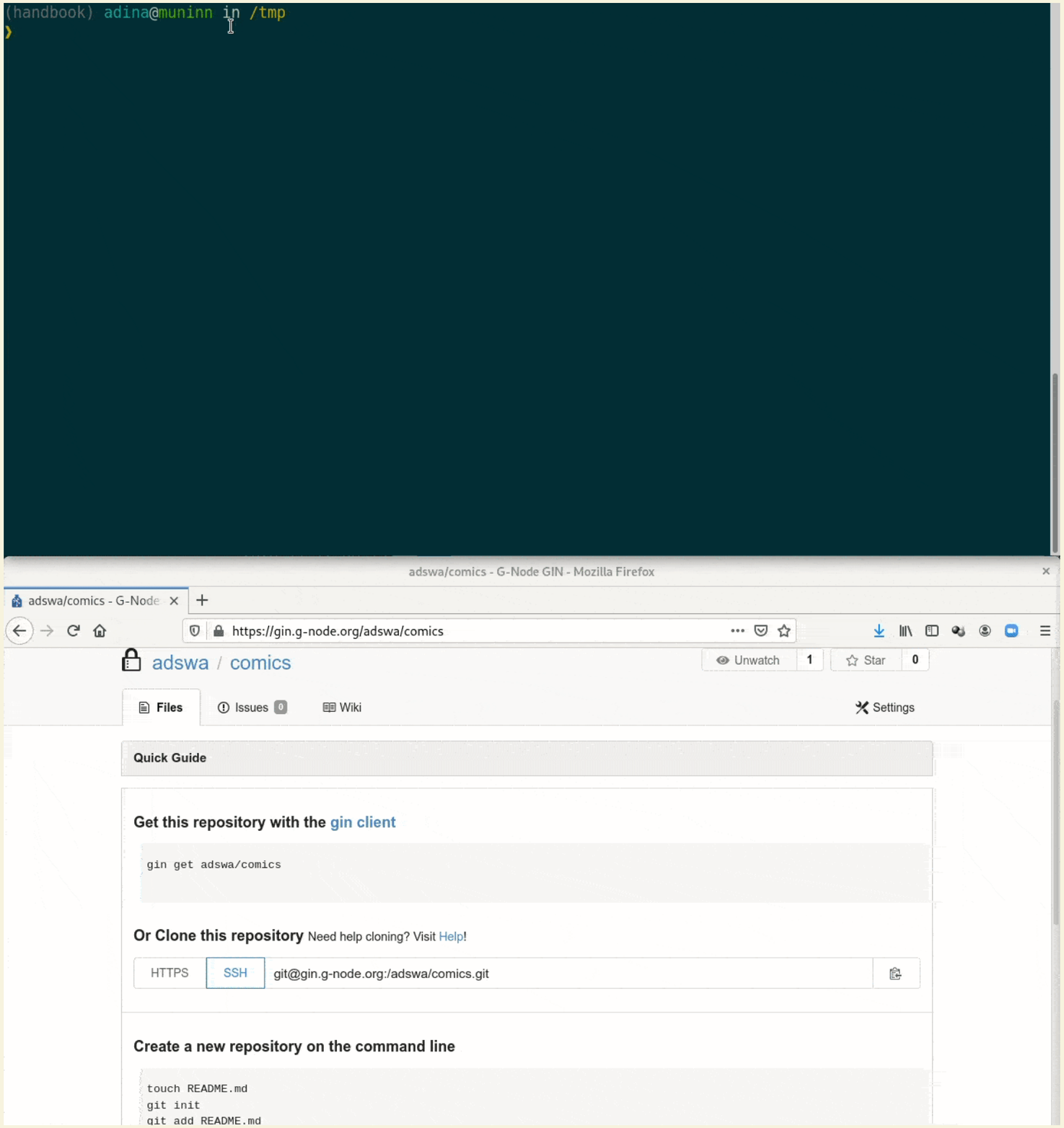


**git-annex**

- contents of annexed files
- organized in the "annex" or "object tree" of the dataset

# PUBLISHING DATASETS

Special case 1: repositories with annex support



# PUBLISHING DATASETS

Special case 2: Special remotes with repositories

# PUBLISHING DATASETS

Special case 3: RIA stores for dataset hosting/backup

# PUBLISHING DATASETS

DataLad can create siblings from the command line for the following services:

**GitHub**

```
datalad create-sibling-github
```

**GitLab**

```
datalad create-sibling-gitlab
```

**Gin**

```
datalad create-sibling-gin
```

**Gogs**

```
datalad create-sibling-gogs
```

**local or remote paths**

```
datalad create-sibling
```

**RIA stores**

```
datalad create-sibling-ria
```

**Open Science Framework (needs datalad-osf)**

```
datalad create-sibling-osf
```

(Additional services being worked on: webdav-based services such as Sciebo, ebrains; if you need something else, get in touch)

# CLONING DATALAD DATASETS

# CLONING DATALAD DATASETS

# CLONING DATALAD DATASETS

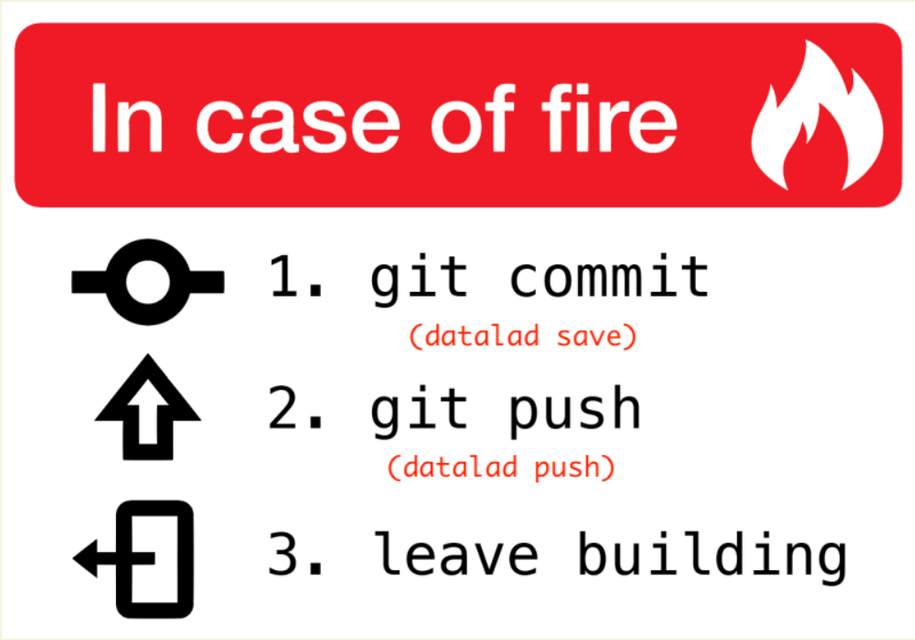
# CLONING DATALAD DATASETS

# CLONING DATALAD DATASETS

Let's take a look at the special cases:

Requires the DataLad extension datalad-osf

# SUMMARY: DATA PUBLICATION




**datasets can have "siblings", linked clones in other places**

Those can be local or remote, on commercial, free, or personal infrastructure

**Typical repository hosting services do not host annexed contents**

A notable exception is Gin

**Typical storage providers do not host Git repositories**

but datalad extensions can make it possible for certain services, such as the OSF

**Despite the different possible services, operations are streamlined**

**clone** installs datasets, **get** retrieves data, **push** publishes (new changes in) datasets, **update** pulls dataset updates. This remains the case even if underlying data hosting changes.

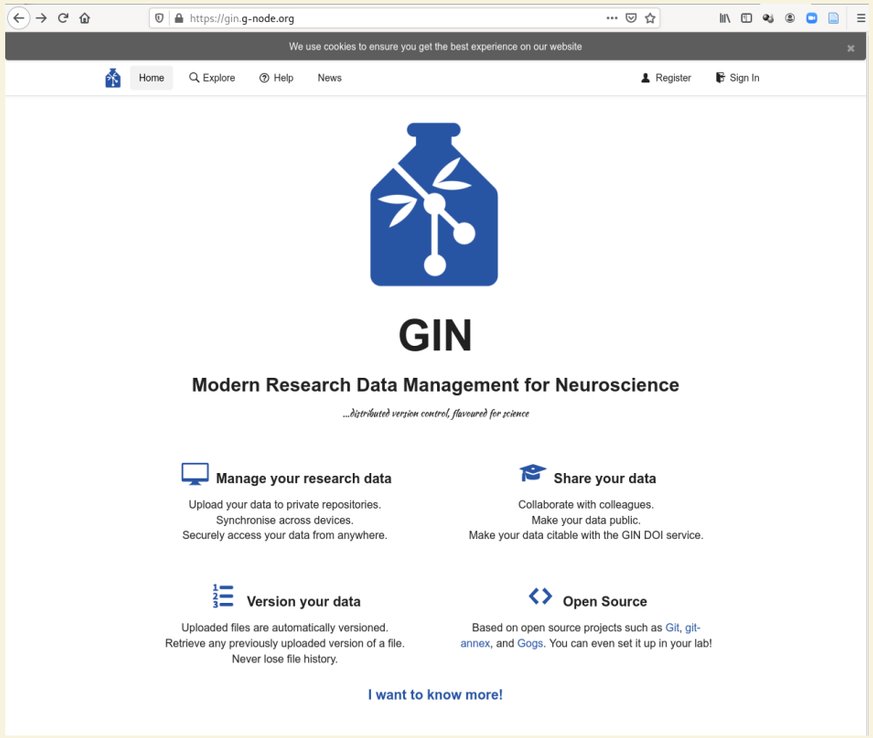
**Siblings serve multiple purposes:**

Personal back-up that's easy to sync; Publicly or privately exposed files to share with (selected) others; Entrypoints for collaborations or others' contributions; ...

# PUBLISH YOUR OWN DATASET

Code: psychoinformatics-de.github.io/rdm-course/03-remote-collaboration/index.html#publishing-datasets-to-gin

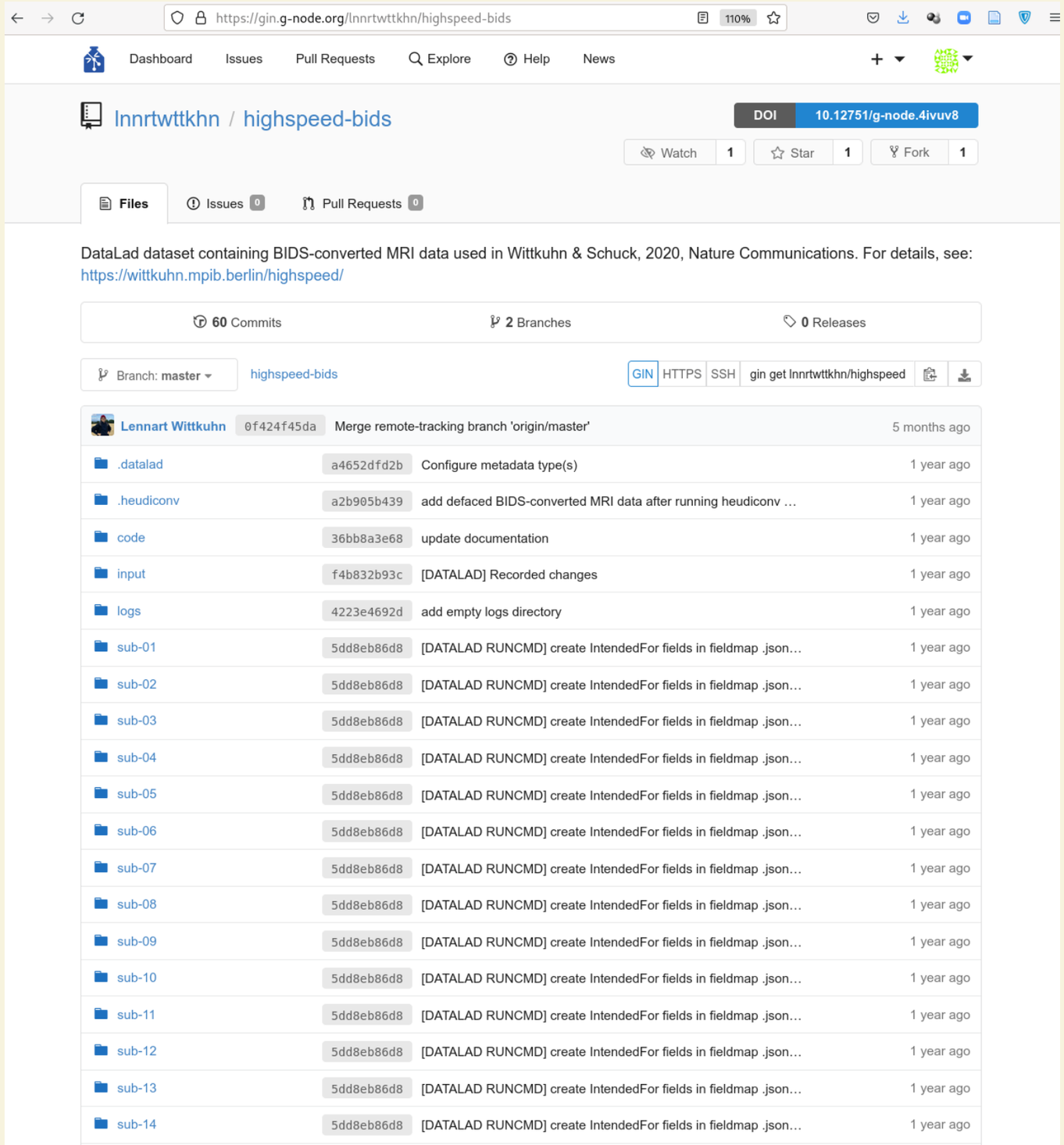
# USING GIN FOR DATA PUBLICATION




**Gin has a few advantages for publishing data**

- DataLad Integration: Convenience commands to create siblings
- Annex support: Easiest possible publication, preview and individual download of annexed contents in the webinterface
- Open Science support: Archive datasets to obtain a DOI; ensures minimal metadata and a license
- Private or Public repositories
- Runs on European infrastructure (some data protection officers like this)
- Free, and with yet unlimited storage

# USING GIN FOR DATA PUBLICATION



# USING GIN FOR DATA PUBLICATION

- Step 1: Create a Gin account (requires an email address)
- Step 2: Generate and upload an SSH key
- Step 3: Create and register a sibling repository
- Step 4: Publish your dataset
- Step 5: Update your dataset

# SUMMARY: PUBLISHING AND UPDATING DATA (GIN)

**Gin is a free repository hosting service**

To publish datasets to Gin, you need an account and an SSH key

**DataLad has built-in integration with datalad create-sibling-gin**

This requires generating an access token

**Gin has annex support**

**datalad push** published all dataset contents and the Git history

**The dataset can be cloned from Gin by others**

If the dataset is public, this does not even require a Gin account

**You can still publish your dataset to (your lab's) GitHub/GitLab/other places**

and use Gin only for data hosting. Walkthrough: handbook.datalad.org/basics/101-139-gin.html#ginbts

**Next: Let's collaborate!**