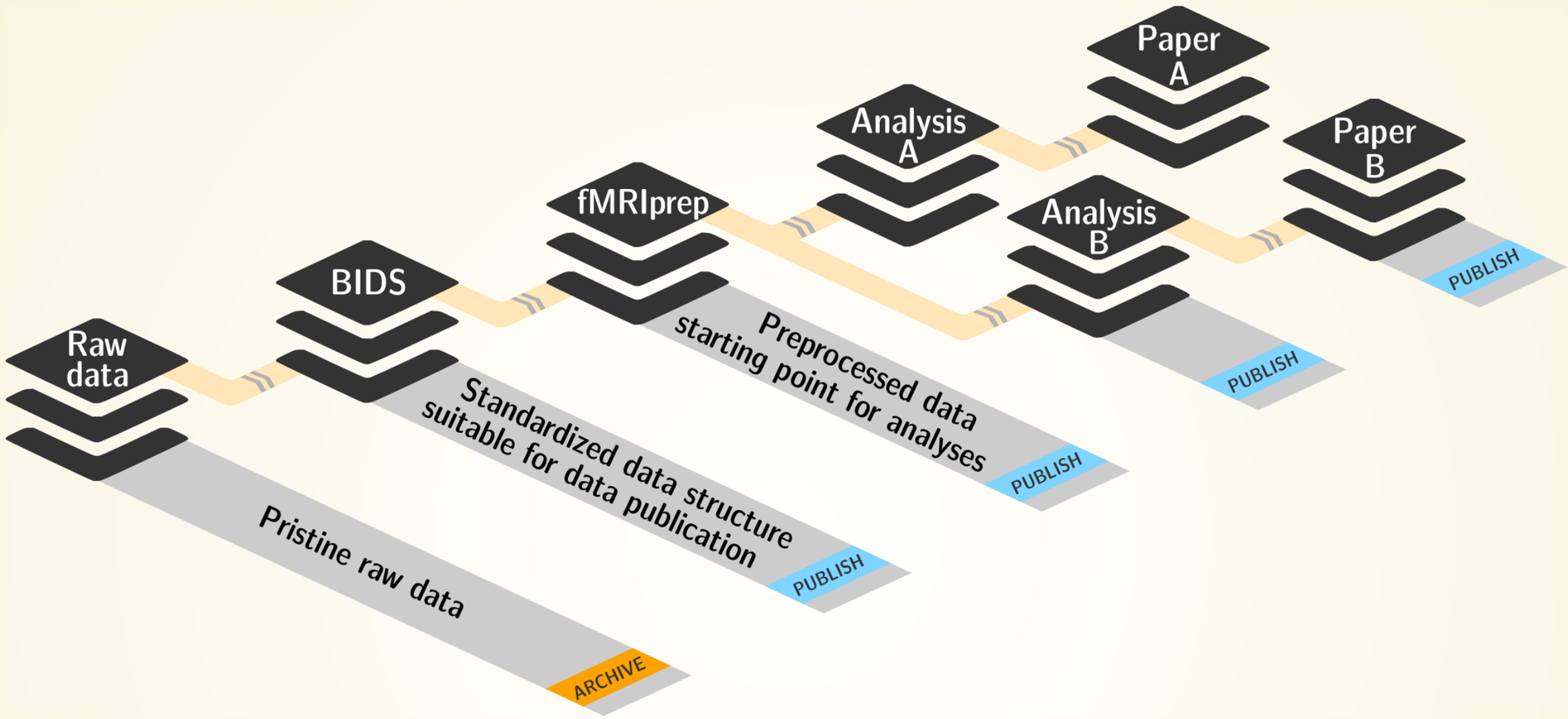
DATASET MANAGEMENT FOR REPRODUCIBILITY & REUSABILITY

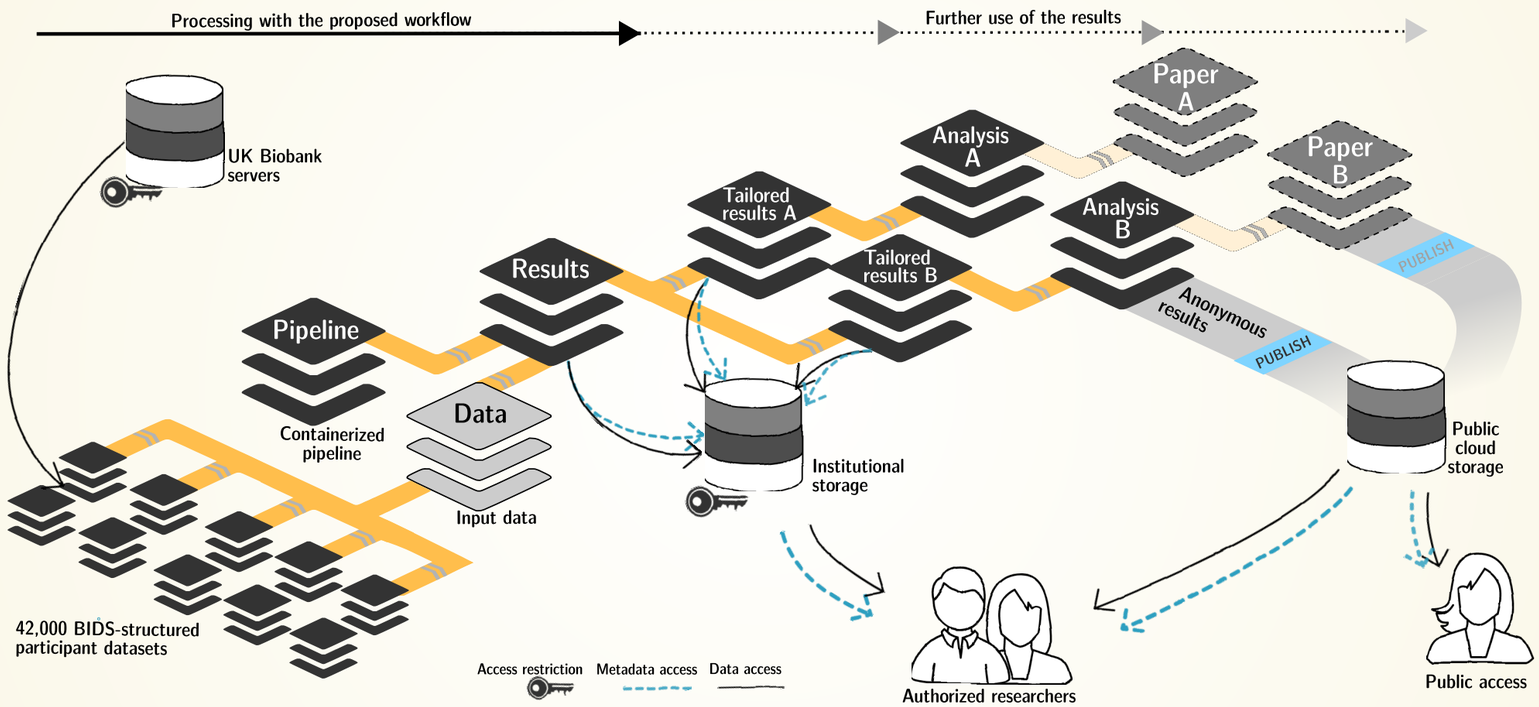
Read more at psychoinformatics-de.github.io/rdm-course/04-dataset-management



When setting up data analyses...

WHY MODULARITY?

- 1. Reuse and access management



- 2. Scalability

```
adina@bulk1 in /ds/hcp/super on git:master❱ datalad status --annex -r
  15530572 annex'd files (77.9 TB recorded total size)
  nothing to save, working tree clean
```

(github.com/datalad-datasets/human-connectome-project-openaccess)

WHY MODULARITY?

- 3. Transparency

  Original:

```
/dataset
├── sample1
│   └── a001.dat
├── sample2
│   └── a001.dat
...
```

Without modularity, after applied transform (preprocessing, analysis, ...):

```
/dataset
├── sample1
│   ├── ps34t.dat
│   └── a001.dat
├── sample2
│   ├── ps34t.dat
│   └── a001.dat
...
```

Without expert/domain knowledge, no distinction between original and derived data possible.

WHY MODULARITY?

- 3. Transparency

  Original:

```
/raw_dataset
├── sample1
│   └── a001.dat
├── sample2
│   └── a001.dat
...
```

With modularity after applied transform (preprocessing, analysis, ...)

```
/derived_dataset
├── sample1
│   └── ps34t.dat
├── sample2
│   └── ps34t.dat
├── ...
└── inputs
    └── raw
        ├── sample1
        │   └── a001.dat
        ├── sample2
        │   └── a001.dat
        ...
```

Clearer separation of semantics, through use of pristine version of original dataset within a *new*, *additional* dataset holding the outputs.

A MACHINE-LEARNING EXAMPLE

Code along or try it later at

handbook.datalad.org/usecases/ml-analysis.html

ANALYSIS LAYOUT

- Prepare an input data set
- Configure and setup an analysis dataset
- Prepare data
- Train models and evaluate them
- Compare different models, repeat with updated data

Imagenette dataset

PREPARE AN INPUT DATASET

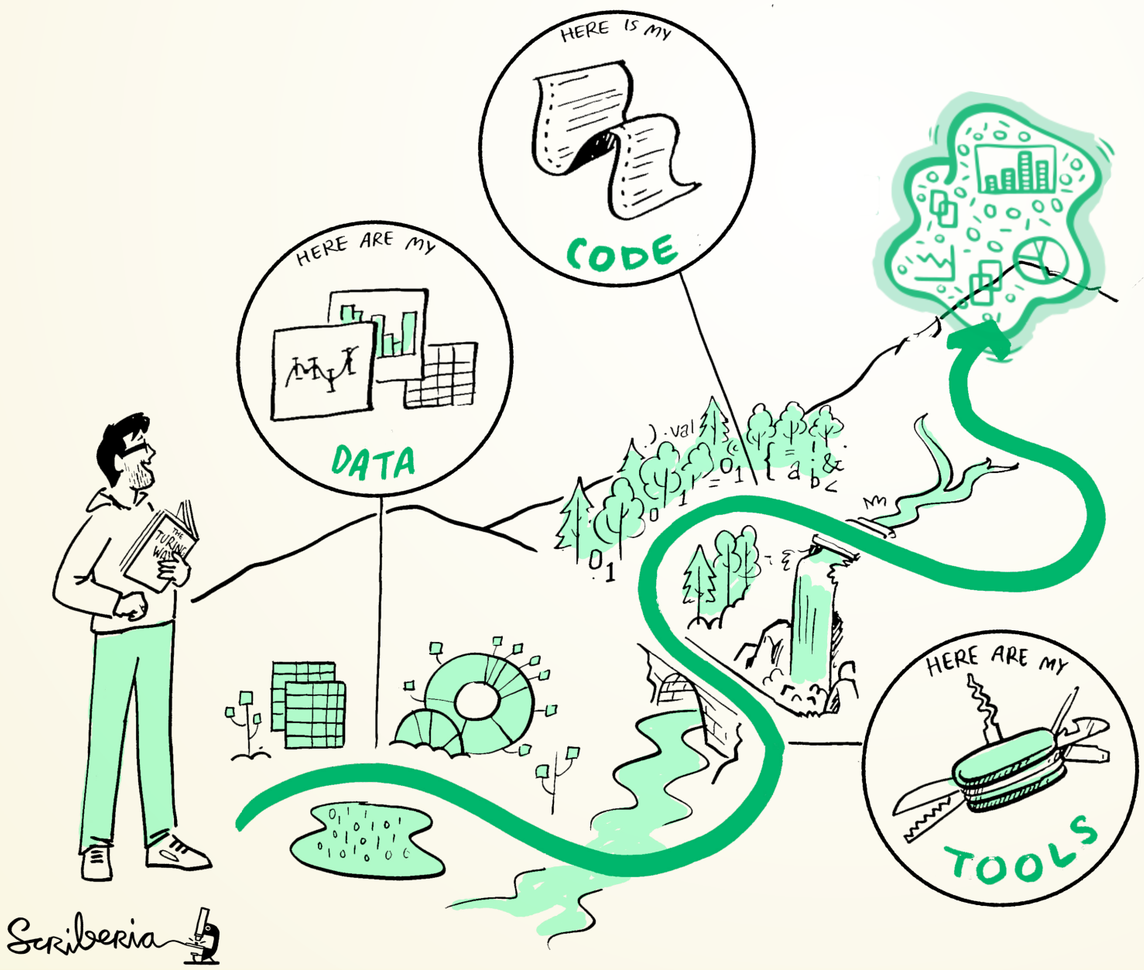
- Create a stand-alone input dataset
- Either add data and `datalad save` it, or use commands such as `datalad download-url` or `datalad add-urls` to retrieve it from web-sources

CONFIGURE AND SETUP AN ANALYSIS DATASET

- Given the purpose of an analysis dataset, configurations can make it easier to use:
  - `-c yoda` prepares a useful structure
  - `-c text2git` keeps text files such as scripts in Git
- The input dataset is installed as a subdataset
- Required software is containerized and added to the dataset

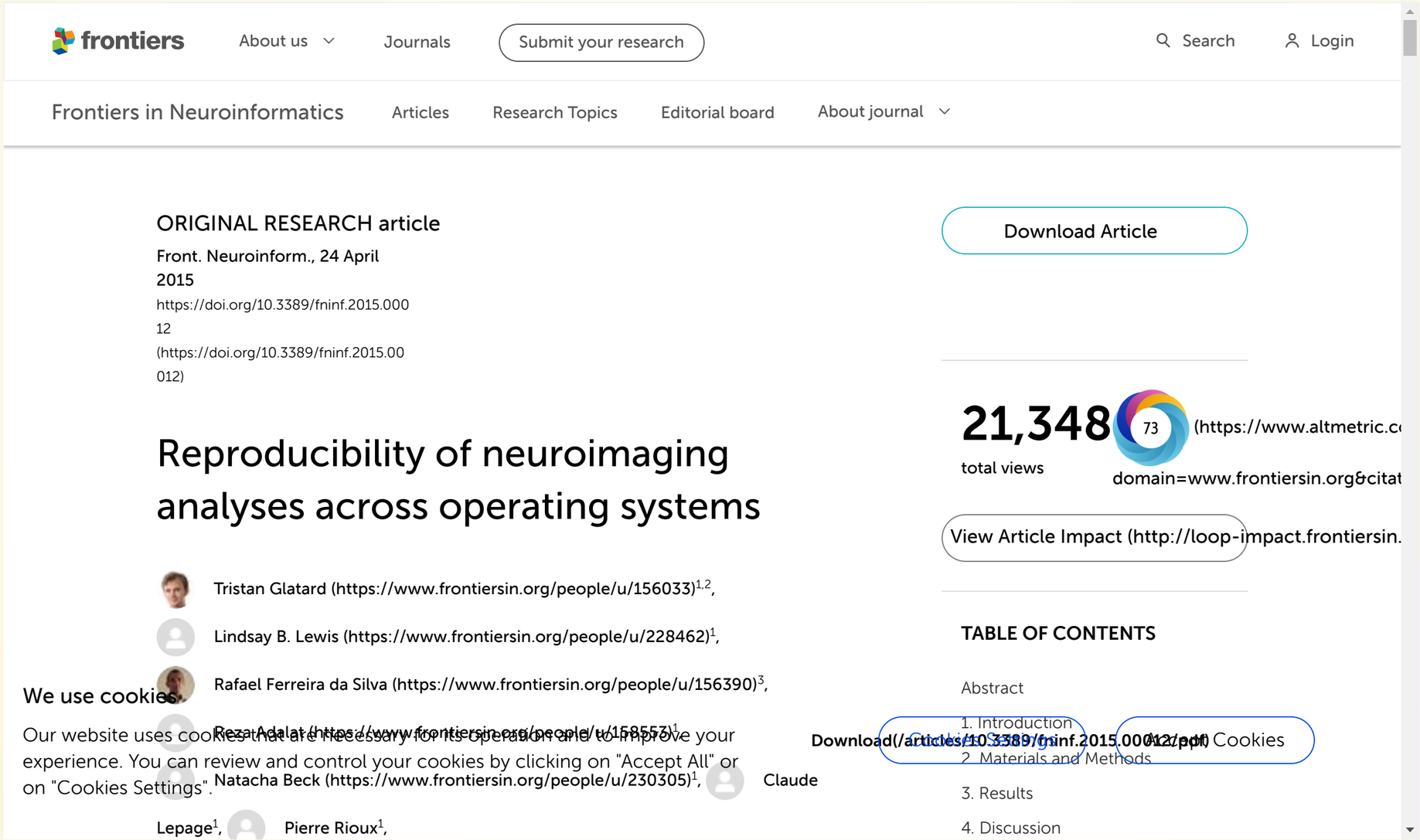
SHARING SOFTWARE ENVIRONMENTS: WHY AND HOW

Science has many different building blocks: Code, software, and data produce research outputs. The more you share, the more likely can others reproduce your results



SHARING SOFTWARE ENVIRONMENTS: WHY AND HOW

- Software can be difficult or impossible to install (e.g. conflicts with existing software, or on HPC) for you or your collaborators
- Different software versions/operating systems can produce different results: Glatard et al., doi.org/10.3389/fninf.2015.00012



SOFTWARE CONTAINERS

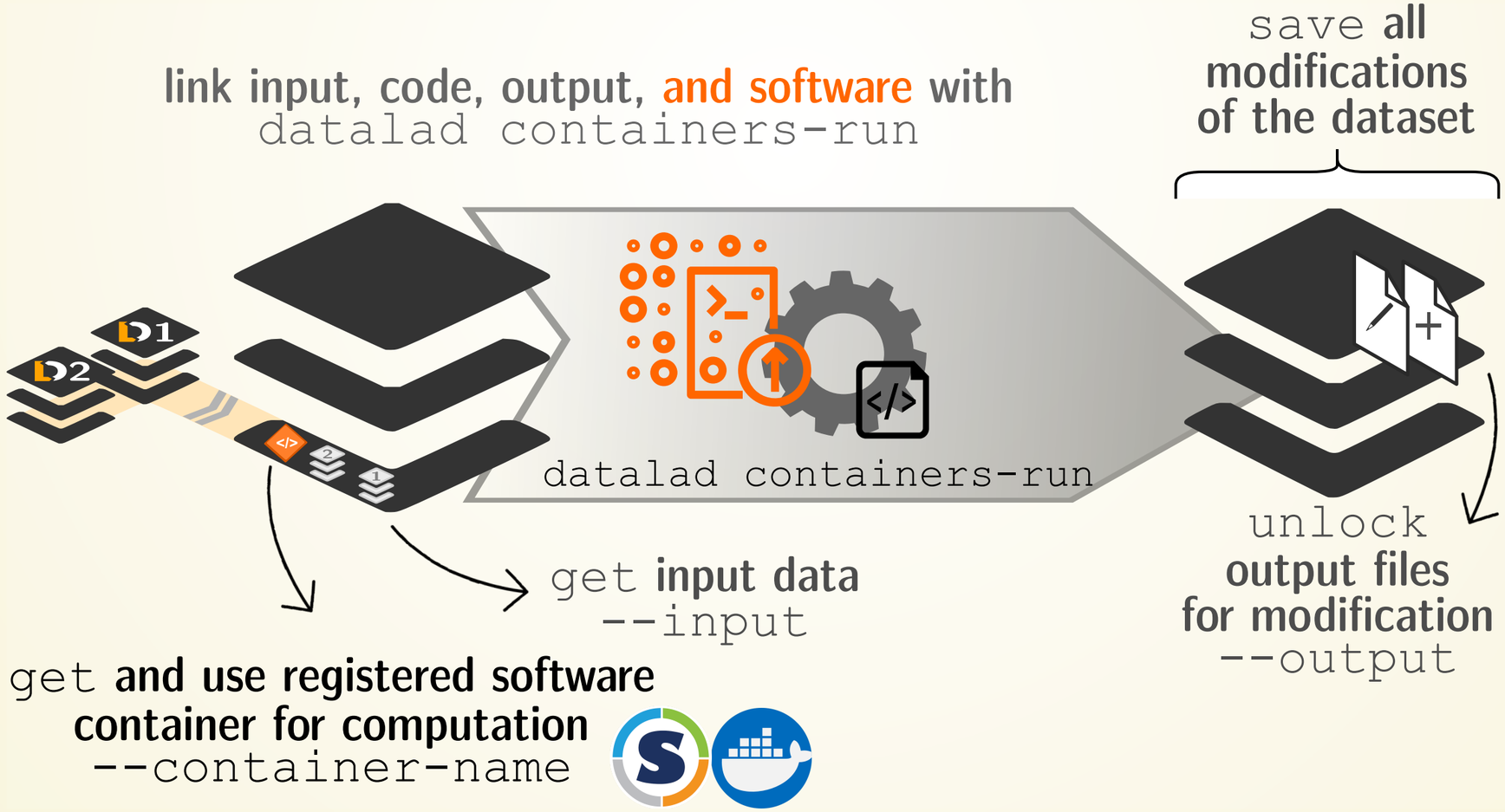
- Put simple, a cut-down virtual machine that is a portable and shareable bundle of software libraries and their dependencies
- **Docker** runs on all operating systems, but requires "sudo" (i.e., admin) privileges
- **Singularity** can run on computational clusters (no "sudo") but is not (well) on non-Linux
- Their containers are different, but interoperable - e.g., Singularity can use and build Docker Images

THE DATALAD-CONTAINER EXTENSION

- The `datalad-container` extension gives DataLad commands to add, track, retrieve, and execute Docker or Singularity containers.

```
pip/conda install datalad-container
```

- If this extension is installed, DataLad can register software containers as "just another file" to your dataset, and **datalad containers-run** analysis inside the container, capturing software as additional provenance



DID YOU KNOW...

Helpful resources for working with software containers:

- repo2docker can fetch a Git repository/DataLad dataset and builds a container image from configuration files
- neurodocker can generate custom Dockerfiles and Singularity recipes for neuroimaging.
- The ReproNim container collection, a DataLad dataset that includes common neuroimaging software as configured singularity containers.
- rocker - Docker container for R users

PREPARE DATA

- Add a script for data preparation (labels train and validation images)
- Execute it using `datalad containers-run`

TRAIN MODELS AND EVALUATE THEM

- Add scripts for training and evaluation. This dataset state can be tagged to identify it easily at a later point
- Execute the scripts using `datalad containers-run`
- By dumping a trained model as a joblib object the trained classifier stays reusable

AND NOW WHAT?

WHEN EVERYTHING IS TRACKED: A REPRODUCIBLE PAPER

- Peer-reviewed paper published in Behavior Research Methods [DOI 10.3758/s13428-020-01428-x]
- Free to reproduce at https://github.com/psychoinformatics-de/paper-remodnav more details in the DataLad handbook http://handbook.datalad.org/r.html?reproducible-paper.
- Full video: https://youtube.com/datalad

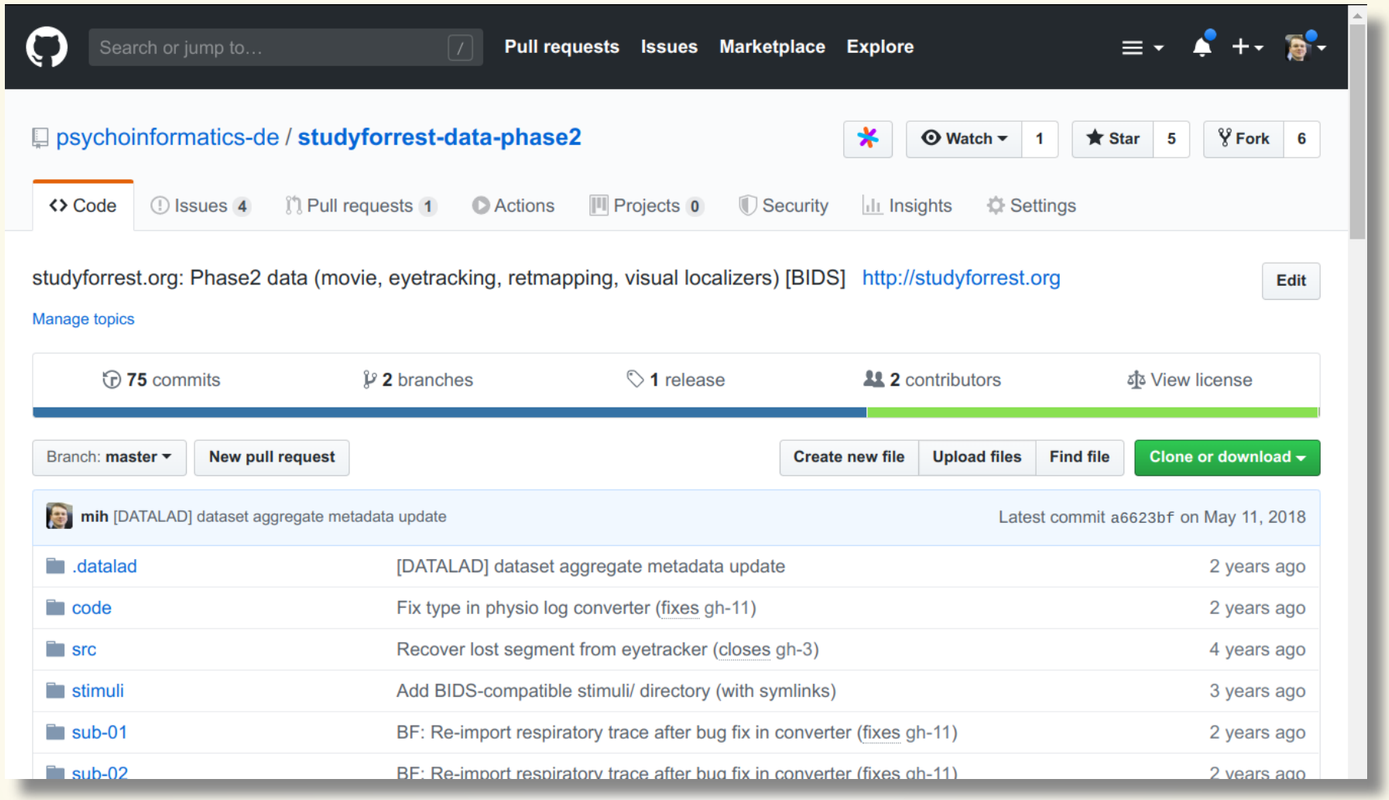
ANTICIPATE CHANGE!

EXHAUSTIVE CAPTURE ENABLES PORTABILITY

Precise identification of data and computational environments, combined for provenance records form a comprehensive and portable data structure, capturing all aspects of an investigation.

Easily take your stuff with you, whereever and whenever you move on!

SERVICES

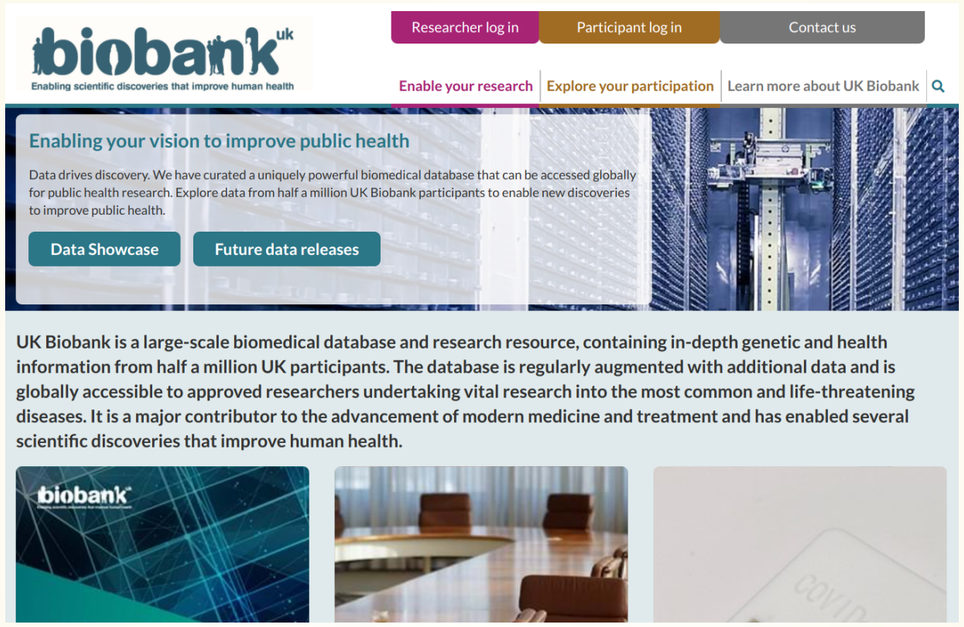


- make *the* difference for advertisment, discovery, convenience
- but imply gigantic dependencies
- often impossible to "take over"

Make sure data/metadata are self-contained
to facilitate/enable transition to another service

IS IT REALLY WORTH THE INVESTMENT?

FAIRLY BIG: PROCESS THE UK BIOBANK (IMAGING DATA)

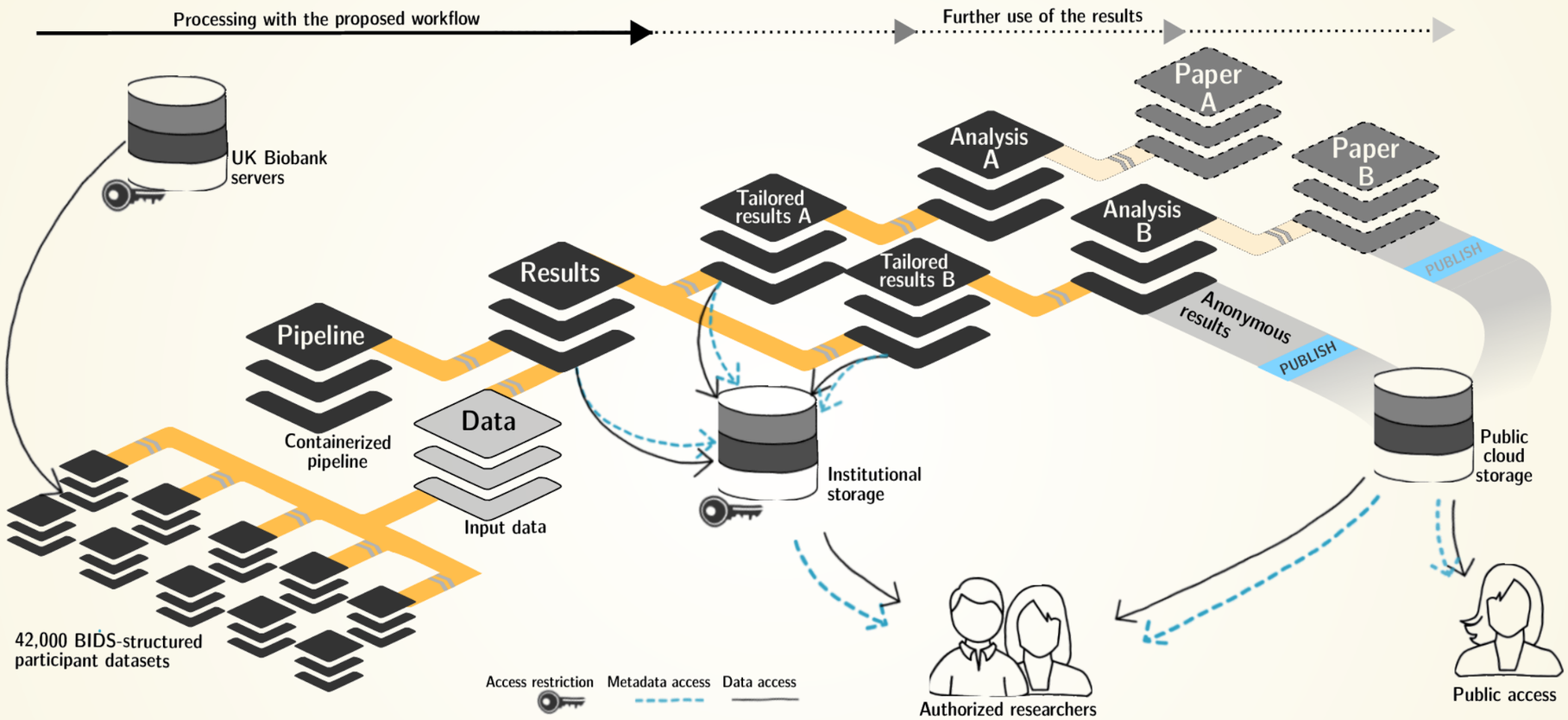


- 76 TB in 43 million files in total
- 42,715 participants contributed personal health data
- Strict DUA
- Custom binary-only downloader
- Most data records offered as (unversioned) ZIP files

CHALLENGES

- Process data such that
  - Results are computationally reproducible (without the original compute infrastructure)
  - There is complete linkage from results to an individual data record download
  - It scales with the amount of available compute resources
- Data processing pipeline
  - Compiled MATLAB blob
  - 1h processing time per image, with 41k images to process
  - 1.2 M output files (30 output files per input file)
  - 1.2 TB total size of outputs

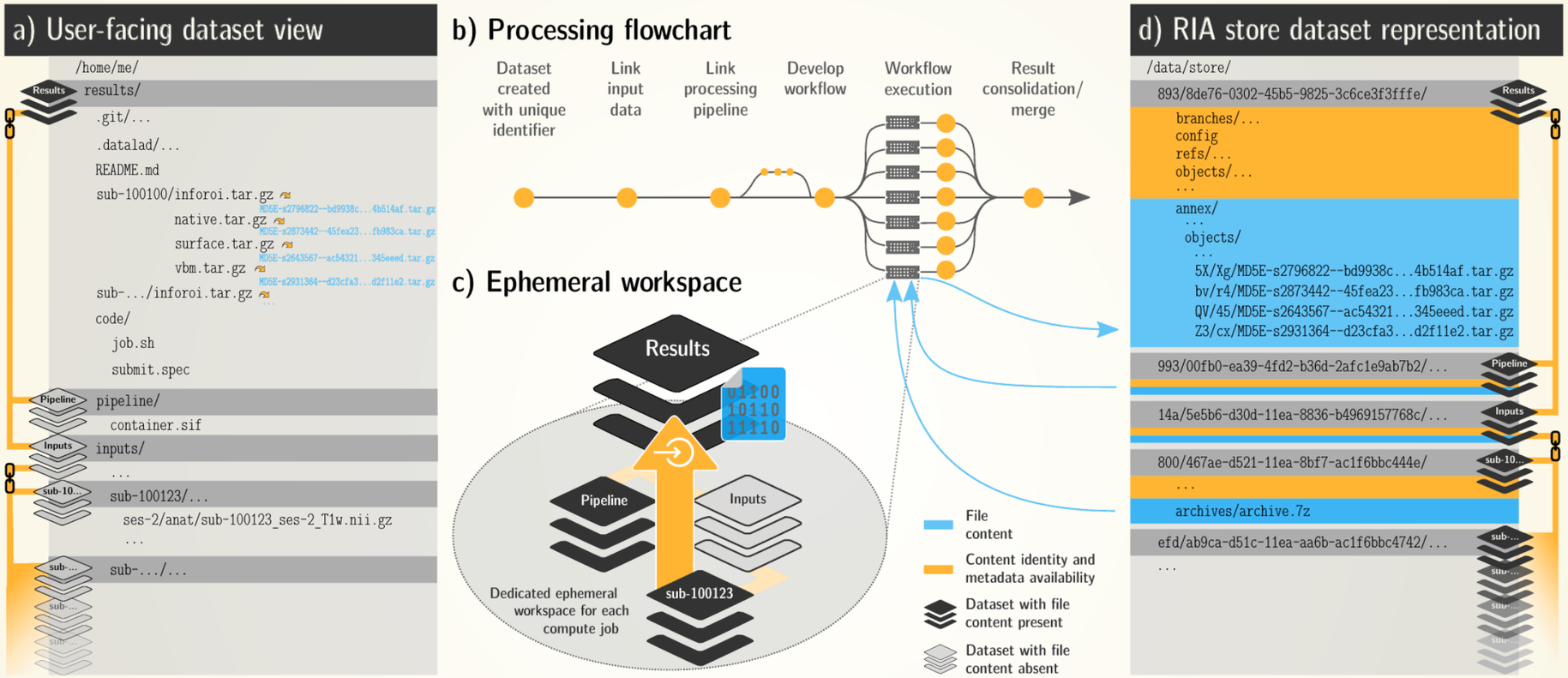
FAIRLY BIG SETUP



- UKB DataLad extension can track the evolution of the complete data release in DataLad datasets
- Full version history
- Native and BIDSified data layout

Wagner, Waite, Wierzba, Hoffstaedter, Waite, Poldrack, Eickhoff, Hanke (2021). FAIRly big: A framework for computationally reproducible processing of large-scale data.

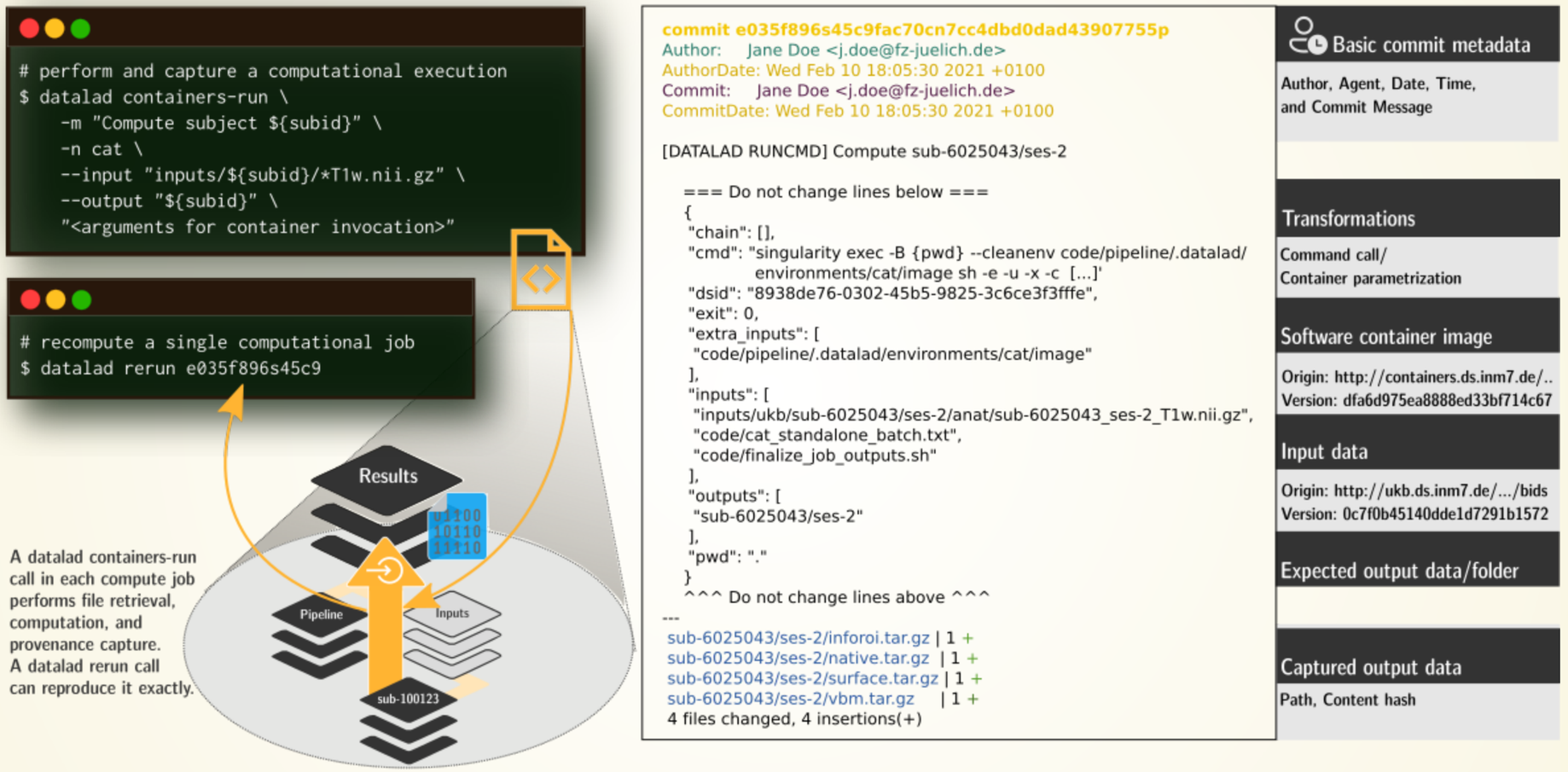
FAIRLY BIG WORKFLOW



- Common data representation in secure environments
- Content-agnostic persistent (encrypted) storage
- All computations in freshly bootstrapped emphemeral environments, only using information from a fully self-contained DataLad dataset

Wagner et al. (2021). FAIRly big: A framework for computationally reproducible processing of large-scale data.

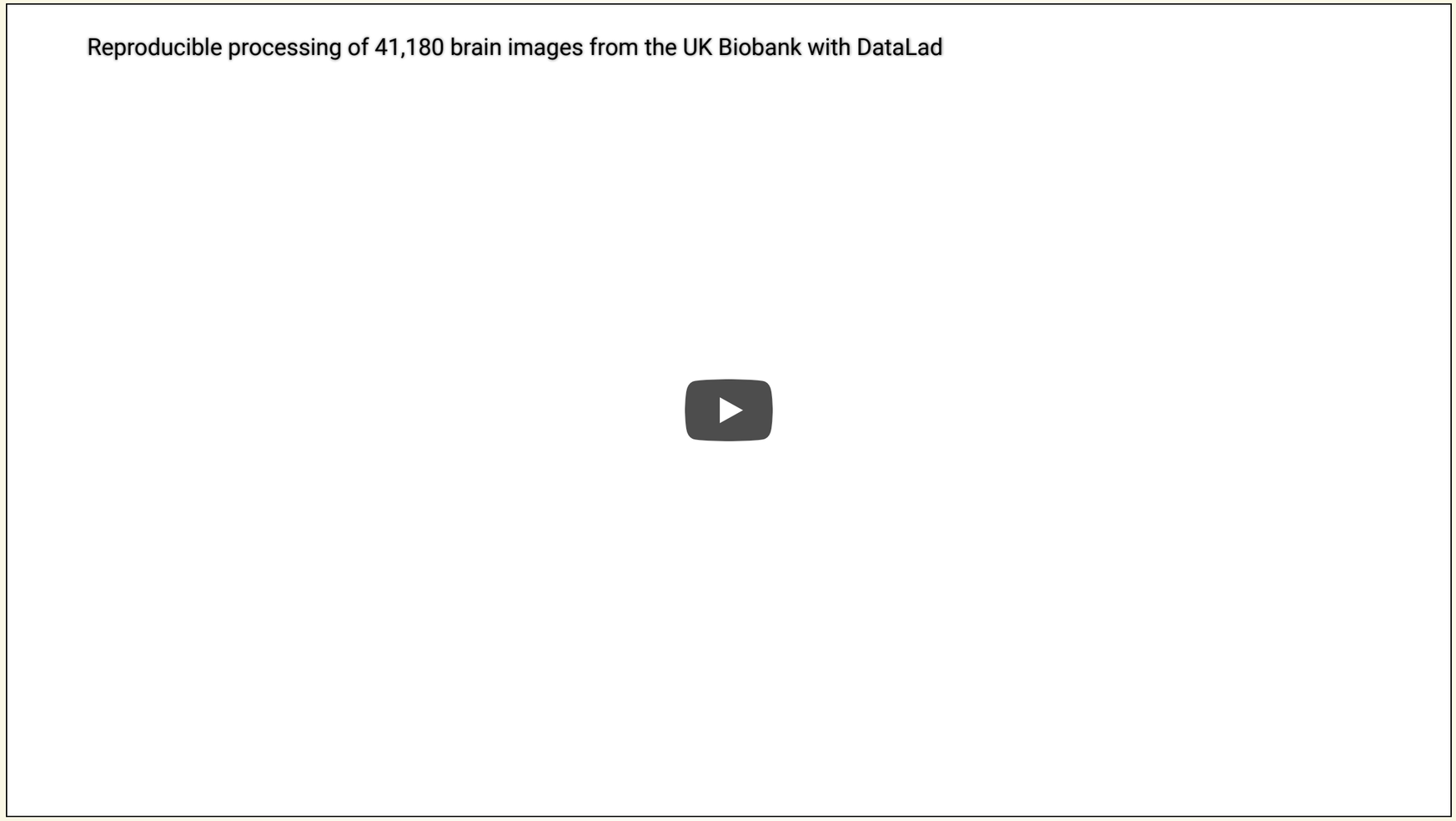
FAIRLY BIG PROVENANCE CAPTURE



- Every single pipeline execution is tracked
- Each execution individually reproducible without HPC access

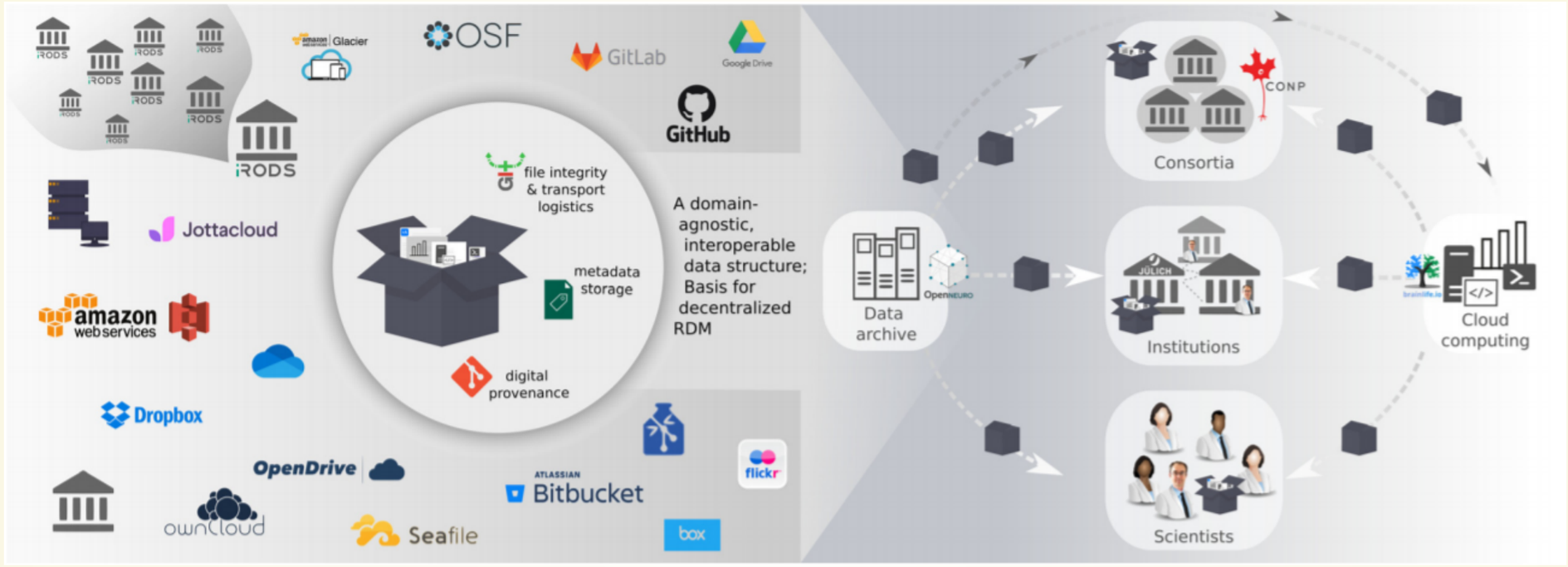
Wagner et al. (2021). FAIRly big: A framework for computationally reproducible processing of large-scale data.

FAIRLY BIG MOVIE



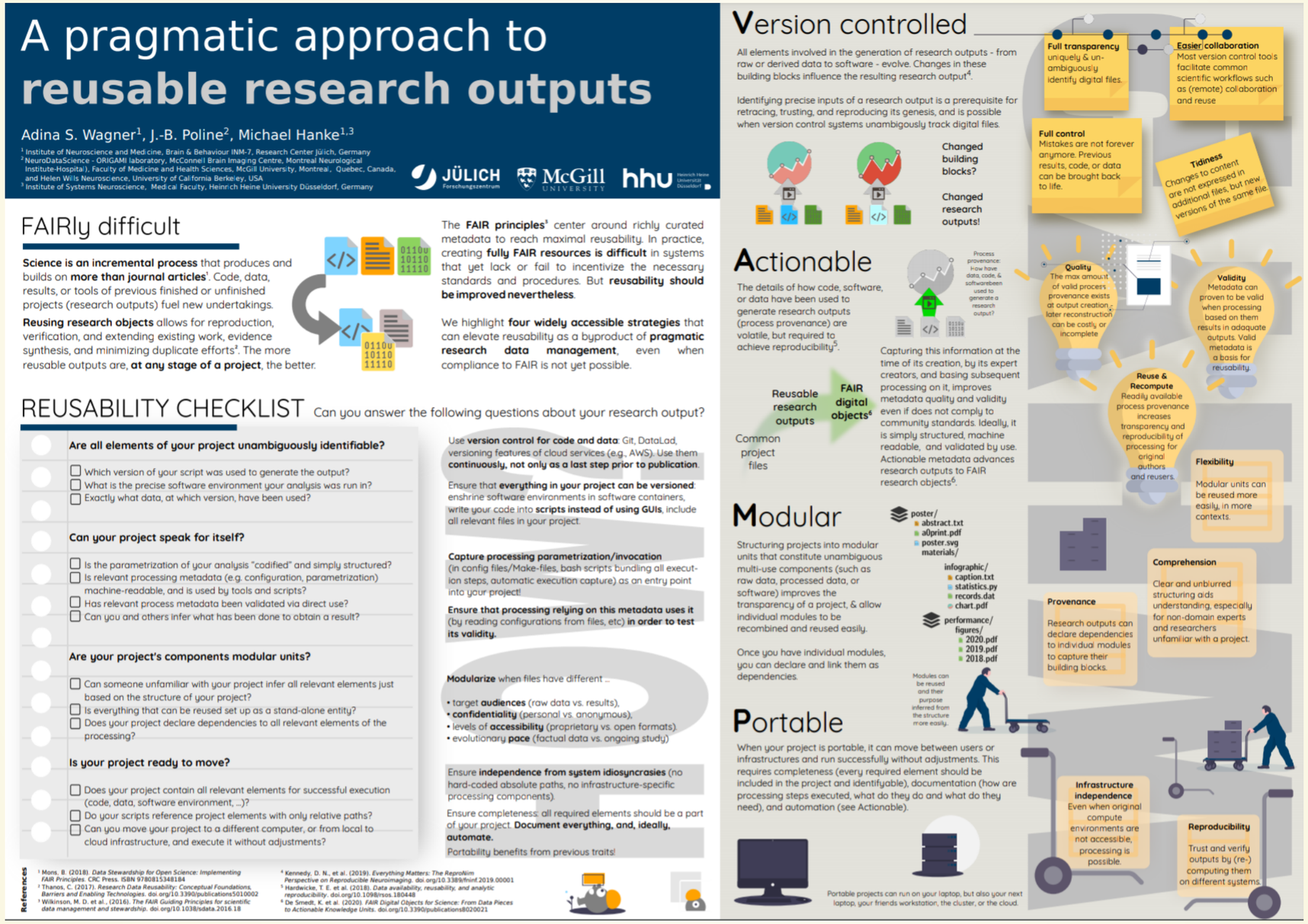
- Rendered exclusively from information captured by DataLad in the output dataset. Full video: https://youtube.com/datalad
- Two full (re-)computations, programmatically comparable, verifiable, reproducible -- on any system with data access

INTEROPERABLE DIGITAL RESEARCH ECOSYSTEM



Hanke, Pestilli, Wagner, Markiewicz, Poline & Halchenko (2021). In defense of decentralized research data management. Neuroforum, 72, 17-25.

TRAINING AND GUIDELINE DEVELOPMENT



Adina S. Wagner, Jean-Baptiste Poline, Michael Hanke. *A pragmatic approach to reusable research outputs.* 10.7490/f1000research.1118575.1 More hands-on details in the DataLad handbook at http://handbook.datalad.org

AFTER THE WORKSHOP

If you have a question after the workshop, you can reach out for help:

Reach out to to the DataLad team via

- Matrix (free, decentralized communication app, no app needed). We run a weekly Zoom office hour (Thursday, 4pm Berlin time) from this room as well.
- the development repository on GitHub

Reach out to the user community with

- A question on neurostars.org with a `datalad` tag

Find more user tutorials or workshop recordings

- On DataLad's YouTube channel
- In the DataLad Handbook
- In the DataLad RDM course
- In the Official API documentation