

Stuttgart  
Computational  
Cognitive  
Science

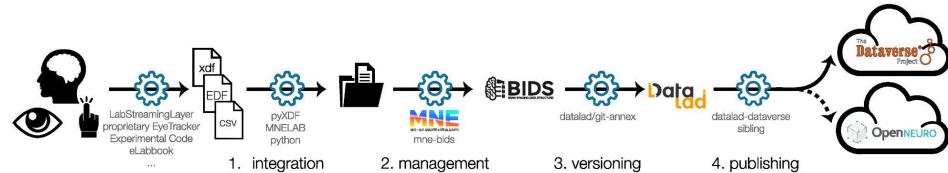


Institut für Visualisierung  
und Interaktive Systeme



Universität Stuttgart

# LSLAutoBIDS:



## Automating Data Integration and Publishing for Neuroimaging

Manpa Barman, Jan Range, Benedikt V. Ehinger  
Computational Cognitive Science Lab, University of Stuttgart

Date : 04.03.2026



# Who am I/we?

I am Manpa Barman :)

Masters Student in IT @ University of Stuttgart

I build research software and work in  
computer vision 👁

Worked with CCS as a RA over past year to  
develop this cool software 💻



Polpo



Benedikt V Ehinger



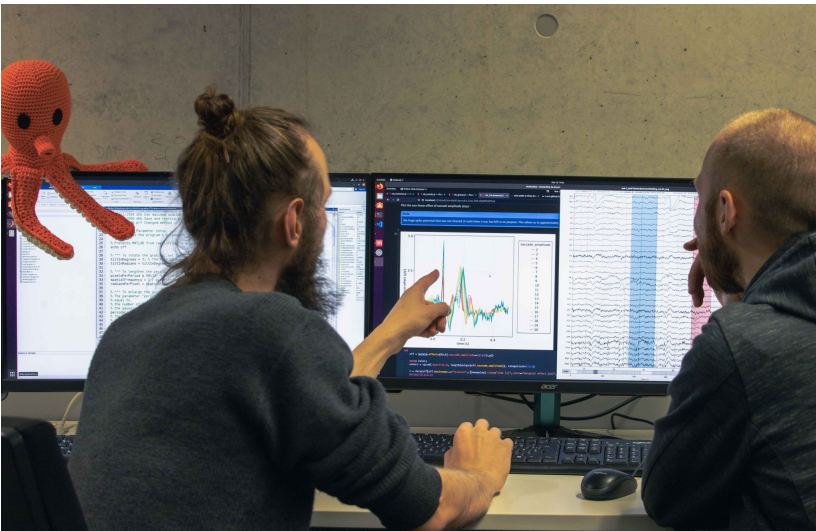
Jan Range

# Who is S-CCS?

Stuttgart  
Computational-Cognitive-Science

Understanding brain activity in combination with  
(eye-) movements

<https://www.s-ccs.de/> <https://github.com/s-ccs>

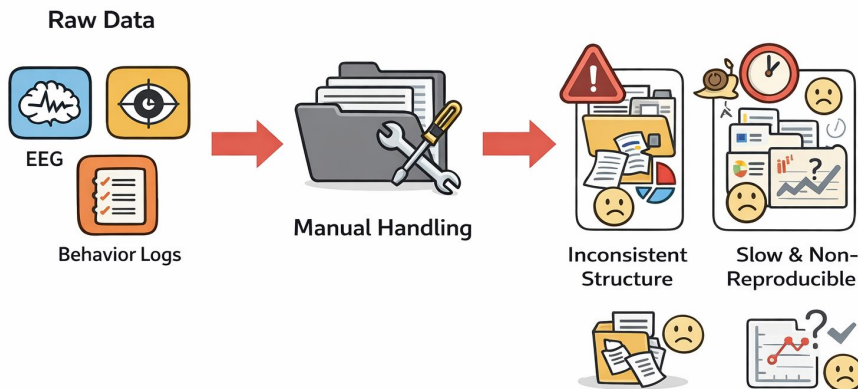


# Motivation

Many scientific domains continuously collect large, multimodal datasets.

Without standards, teams end up with manual handling of data, which is costly, error-prone, and slows progress.

This also weakens reproducibility, makes sharing difficult, and creates barriers to open science.

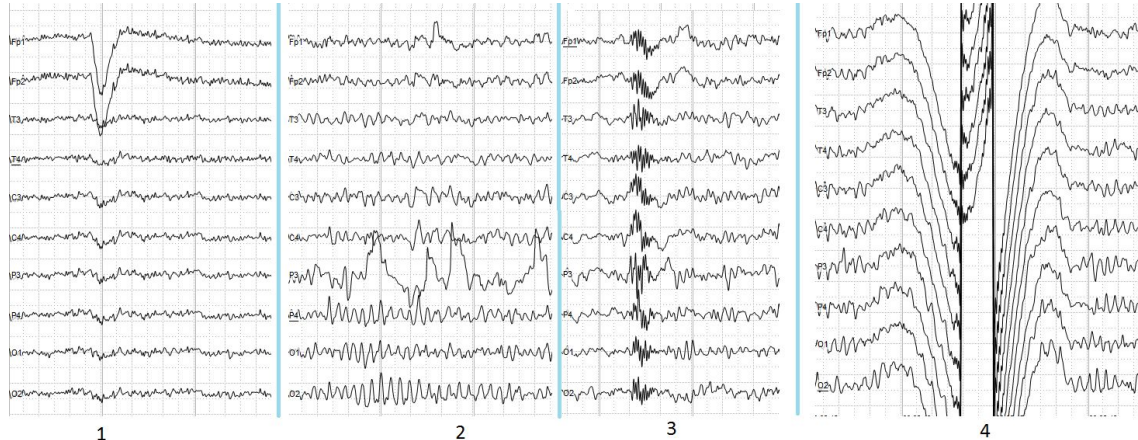




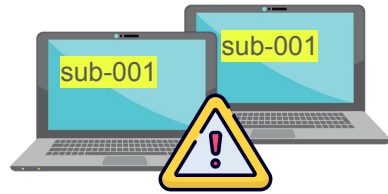
# Motivation - Neuroscience: A Representative Example

Modern neuroscience studies produce *a lot of* heterogeneous data (EEG, EMG, Eye Tracking, behavioral logs etc.).

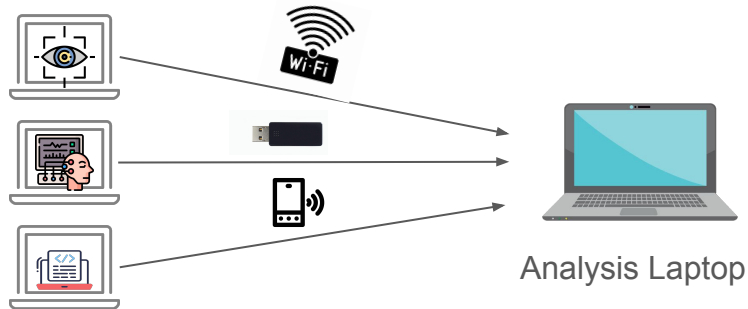
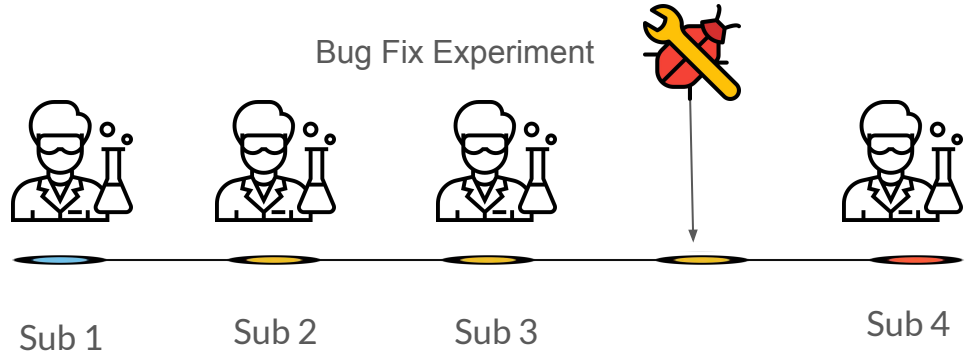
These must be synchronized, unified, and standardized — otherwise the data are hard to reuse, reproduce, or share reliably.



# What can go wrong ? - Concrete Examples

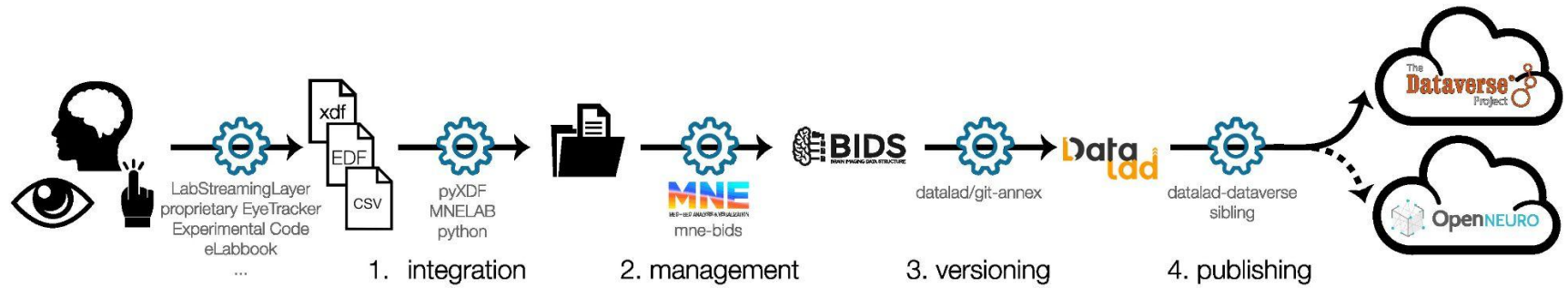


Overwritten Data



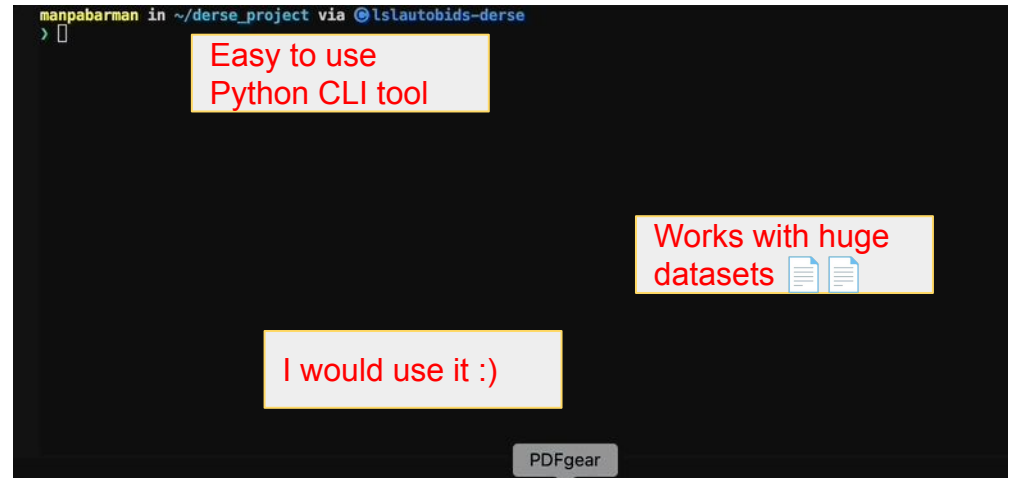
Recording Laptops



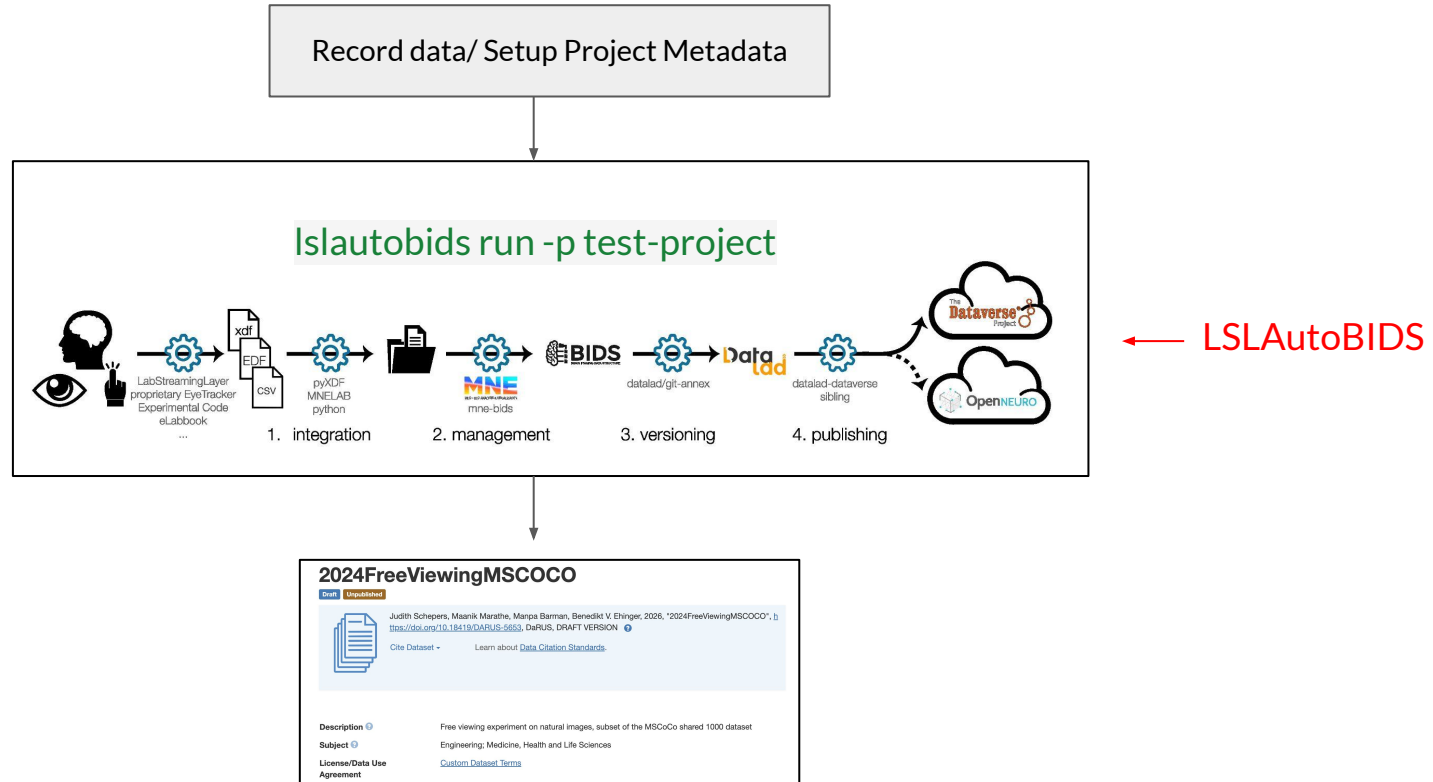


## Introducing LSLAutobids

- Simple to use Python CLI tool
- Uses popular solutions as part of the workflow -> scalable
- Works with huge datasets



# LSLAutoBIDS : How it looks for the user?

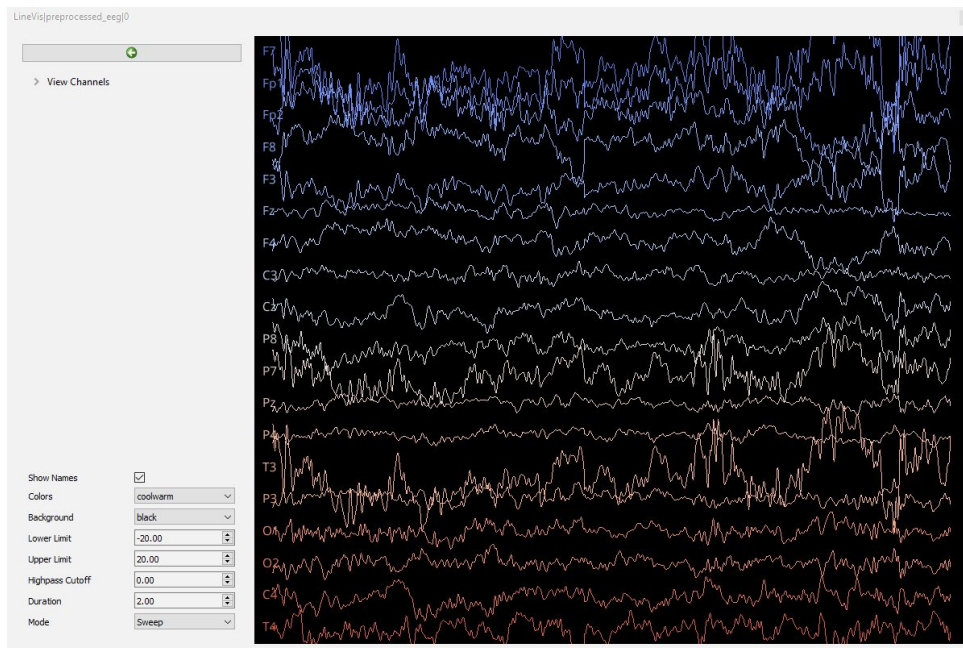






# 1. Integration

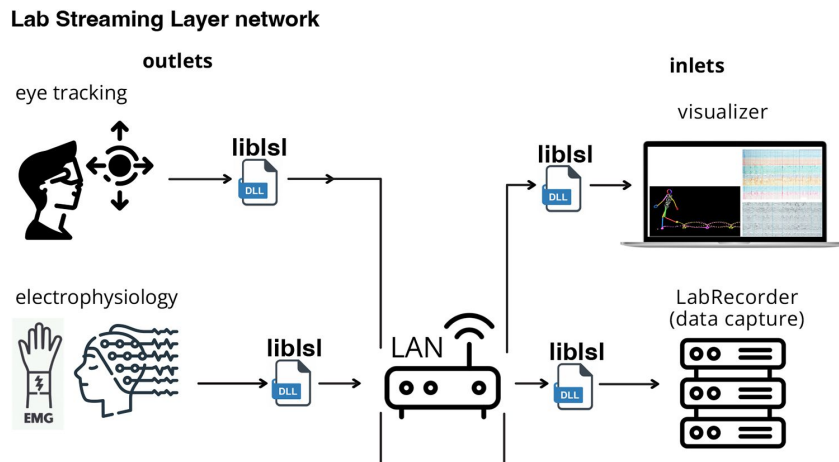
- Multiple heterogeneous data streams
- Different formats, different clocks
- Need synchronized integration



Adapted from <https://labstreaminglayer.readthedocs.io/info/viewers.html>

# Data integration - Lab Streaming Layer (LSL)

- LSL(Kothe et al., 2025) is a Middleware to stream, receive, synchronize, and record neural, physiological, and behavioral data streams
- Provides sub-millisecond timestamp synchronization across heterogeneous devices
- Recording via tools like LabRecorder, producing unified XDF container files



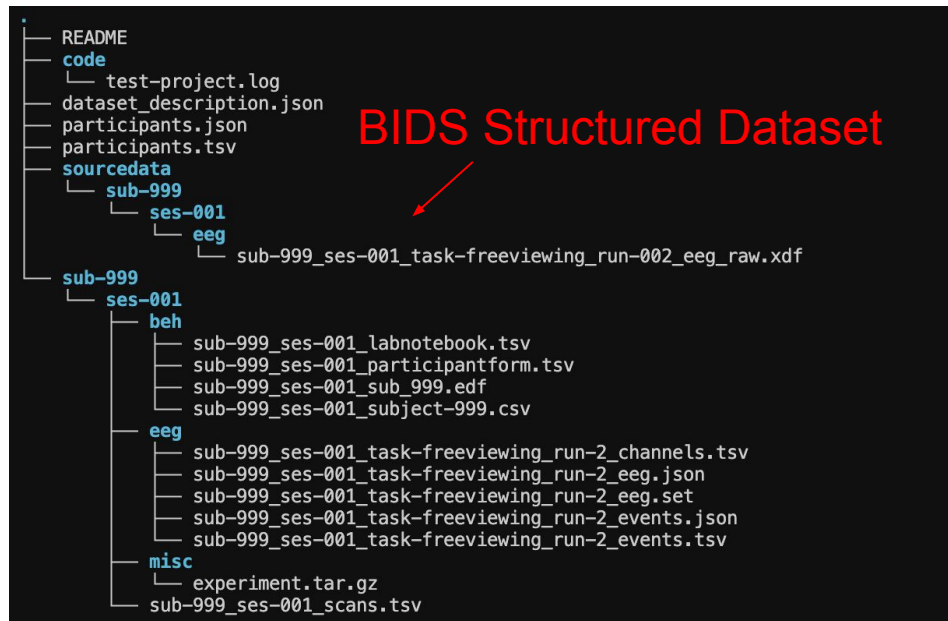
# Data integration: Our Implementation

- Read synchronized XDF recordings using `pyxdf`
- Identify which streams contain EEG data, motion tracking, or other signals
- Include **non-LSL files** (e.g. behavioral CSV) based on project configuration
- Collect all files belonging to one participant into a unified session structure
- Use a central project configuration to control metadata and publication details.



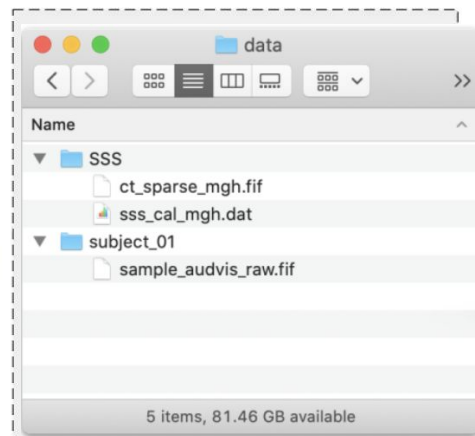
## 2. Management

- From raw files to structured datasets
- Enforce consistent metadata
- Enable validation and reuse

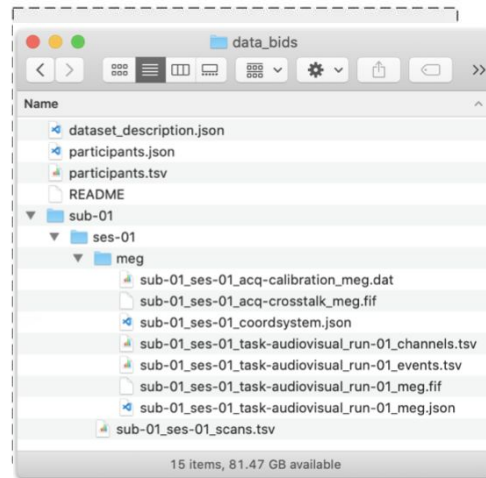


# What is BIDS?

- **BIDS = Brain Imaging Data Structure**, a community-driven standard for organizing neuroimaging data.
- Key principles:
  - Clear directory structure
  - Standard metadata (JSON/TSV etc.)
  - Supports many modalities (EEG, MRI, MEG, etc.)
- Enables tooling, validation, reproducibility.



Arbitrarily organized data



BIDS-compliant dataset

# Data Management: Our Implementation

## Conversion

- Convert synchronized EEG data to BIDS using `mne` and `mne-bids`
- Extract metadata automatically from project configuration

## Validation

- Validate output using the BIDS Validator
- Exclude auxiliary files via `.bidsignore`

## Extended Modalities

- Organize additional data (EyeLink EDF, logs, code snapshots) within the BIDS directory structure
- Maintain a single standardized dataset structure across modalities





### 3. Versioning

- Data changes over time
- Large binary files are hard to track
- Need reproducible history

Major Minor Patch

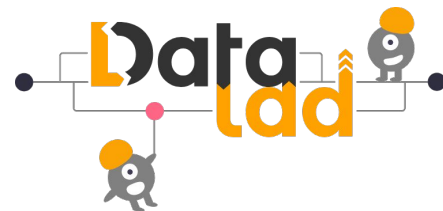
v2.21.2

New recording cohort

Adding subjects, samples, new metadata

Fixing metadata, bugs

# Versioning - Datalad to the rescue



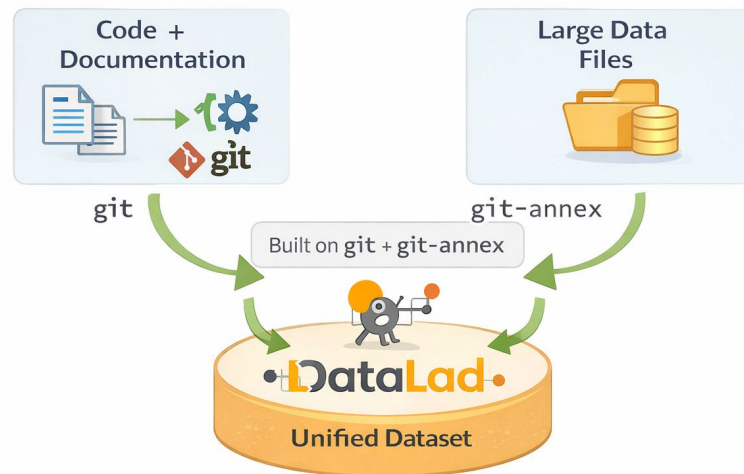
## Git

- Excellent for code and text
- Not suitable for large binary datasets

## DataLad (built on git + git-annex)

- Enables versioning of large datasets
- Tracks files via pointers instead of storing full content
- Supports distributed collaboration and remote storage

## Version Control for Research Data



# Versioning: Our Implementation

## Initialization

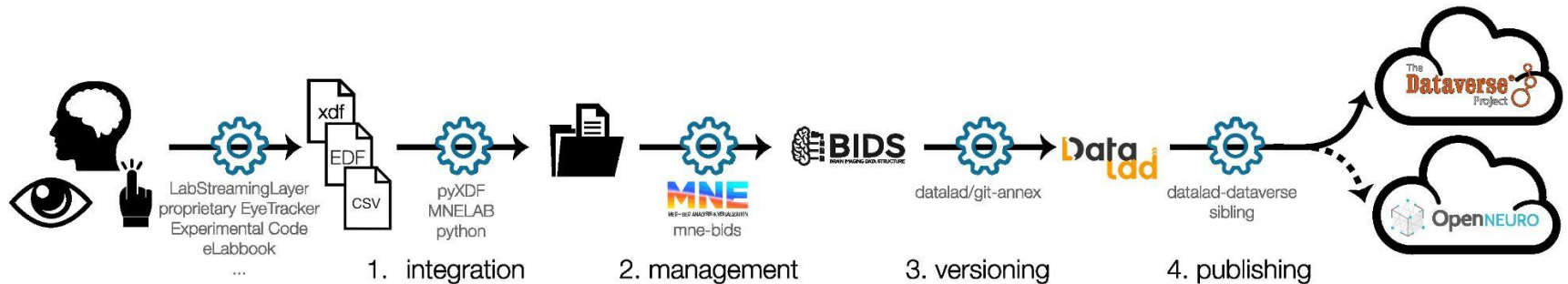
- After BIDS conversion, a DataLad dataset is initialized
- Entire project (data + code + metadata) becomes version-controlled

## Under the Hood

- Git tracks code, text, and metadata
- git-annex manages large binary files via pointers

## Minimal User Interface

- Writing: `datalad save`, `datalad push` and Reading: `datalad clone`, `datalad get`
- Enables scalable exploration of terabyte-scale datasets



## 4. Publishing

- From local dataset → persistent archive
- Assign DOI and metadata
- Enable controlled access and sharing

### 2024FreeViewingMSCOCO

Draft Unpublished



Judith Schepers, Maanik Marathe, Manpa Barman, Benedikt V. Ehinger, 2026, "2024FreeViewingMSCOCO", <https://doi.org/10.18419/DARUS-5653>, DaRUS, DRAFT VERSION ?

Cite Dataset ▾

Learn about [Data Citation Standards](#).

Description ?

Free viewing experiment on natural images, subset of the MSCoCo shared 1000 dataset

Subject ?

Engineering; Medicine, Health and Life Sciences

License/Data Use  
Agreement

[Custom Dataset Terms](#)

# Publishing or Archiving

- Open-source research data repository platform
- Provides DOI assignment
- Supports access control and long-term archival
- FAIR-aligned metadata support



**HARVARD**  
DATAVERSE



# Publishing: Our Implementation

## Automated Upload

- LSLAutoBIDS automatically deposit versioned dataset to Dataverse
- Include BIDS dataset, raw source data, and stimulus files

## Metadata Automation

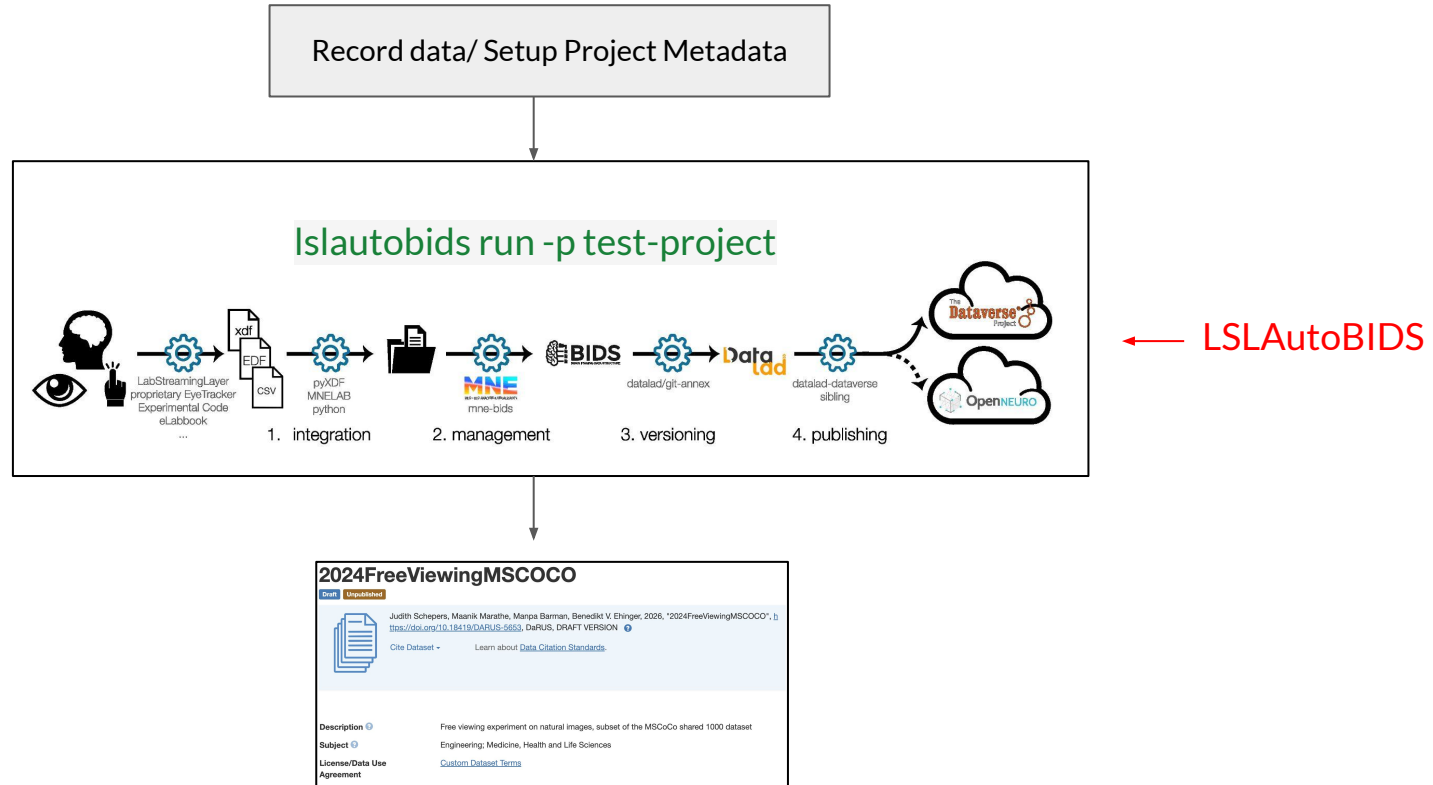
- Project-level configuration populates repository metadata
- Authors, license, description automatically included

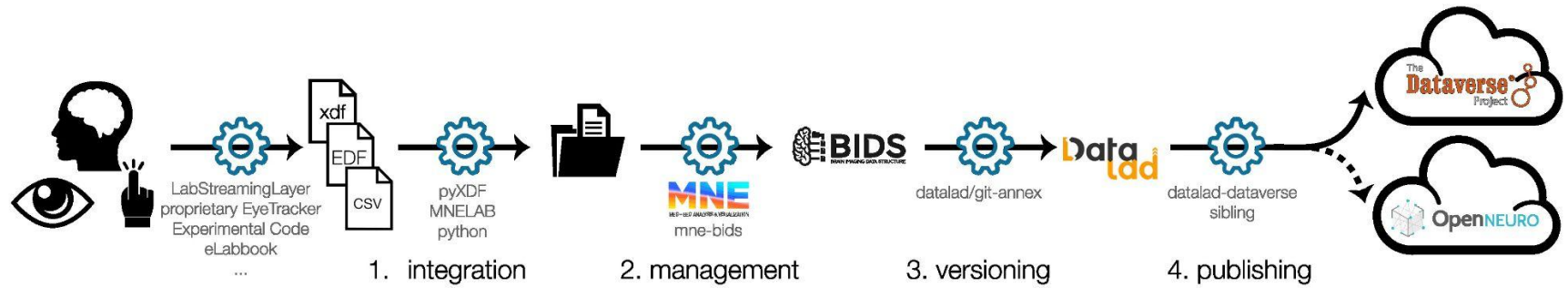
## Reproducible Archival

- Dataset version trace preserved via DataLad
- DOI assigned for citation



# LSLAutoBIDS : How it looks for the user?





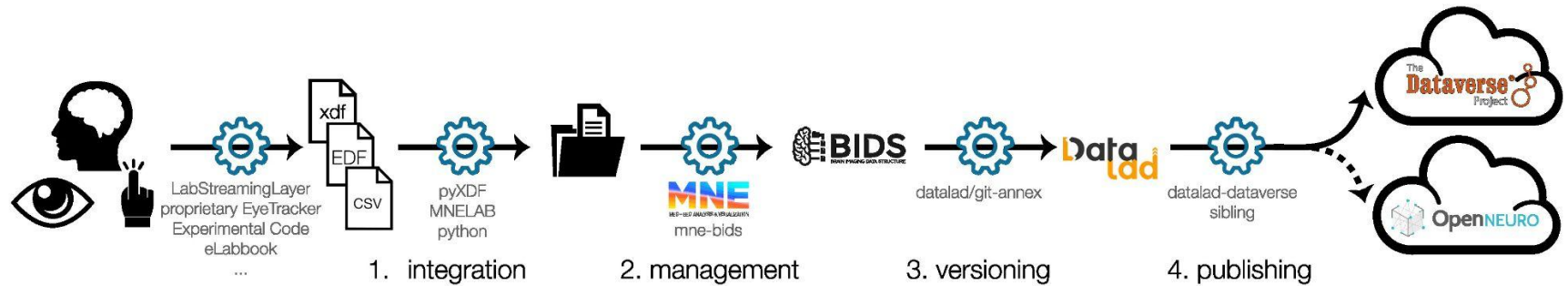
## Discussion: What Next?

Maintenance by S-CCS lab

Make data-integration more flexible

Add more metadata

Integrate BIDS-EyeTracking conversion



## Conclusion

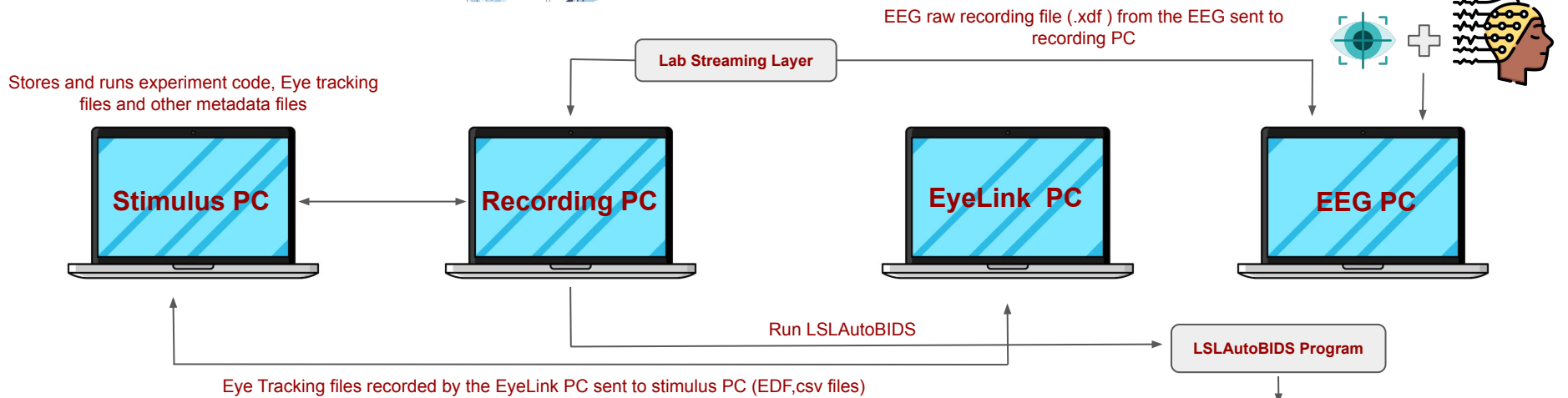
Integration, data management, versioning and publishing can be automated

Use automated pipelines like LSLAutoBIDS to simplify your lab life

Standardized dataset are much more fun to analyze afterwards.

Open-science by design!

# Thank You!



# LSLAutoBIDS

