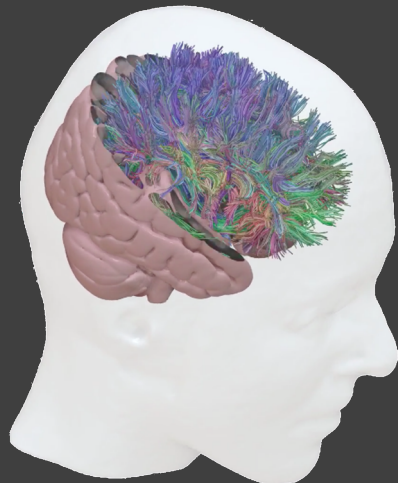
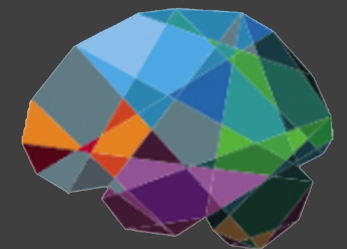




GDPR compliant TVB image processing container workflows with HPC on EBRAINS



Dr. Michael Schirner
Brain Simulation Section (PI: Prof. Petra Ritter)
Charité—Universitätsmedizin Berlin
November 25th, 2021

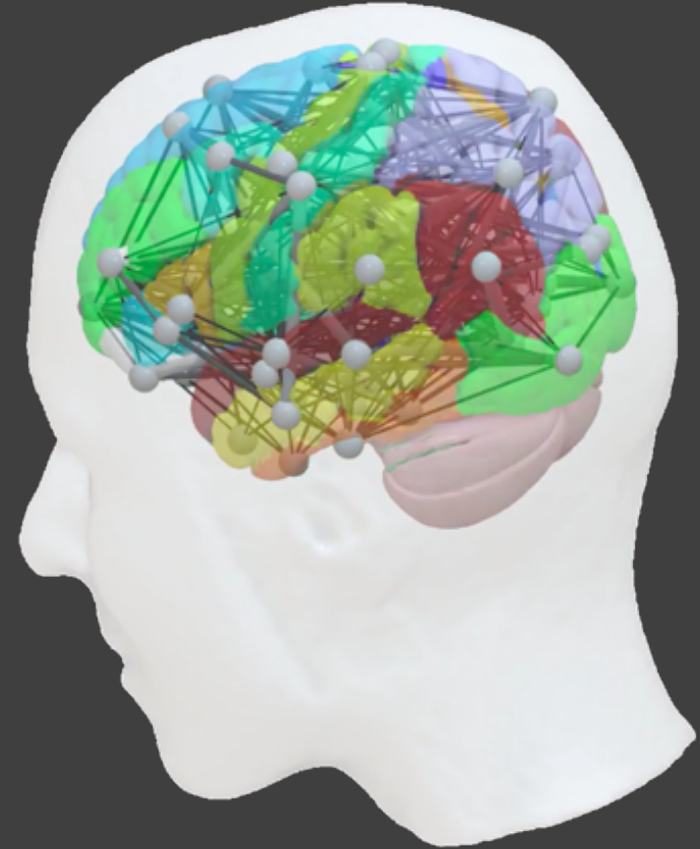
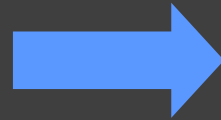


THEVIRTUALBRAIN.

TVB Image Processing Pipeline



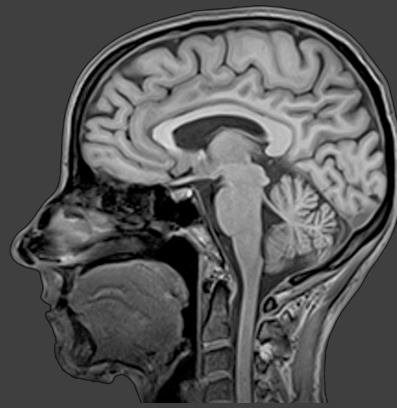
magnetic resonance imaging



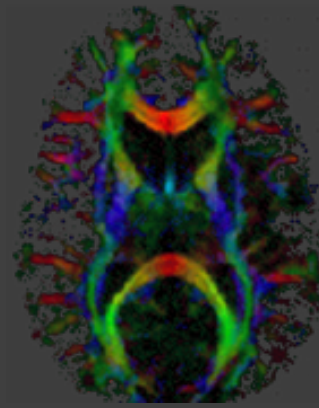
brain network model



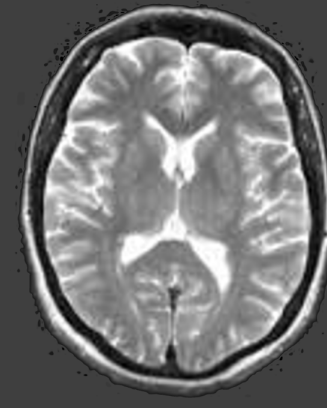
Pipeline input



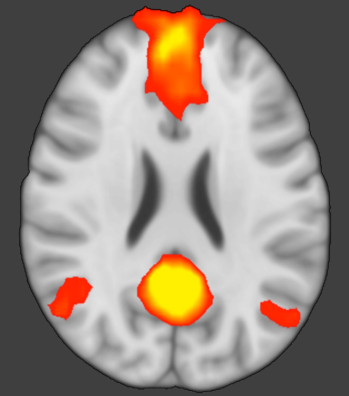
T1-weighted MRI



Diffusion-weighted MRI

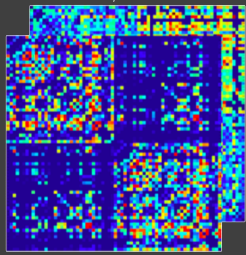


T2-weighted MRI

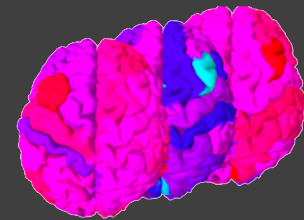
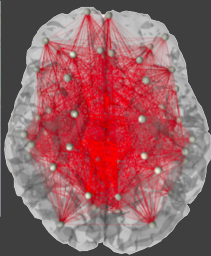


functional MRI

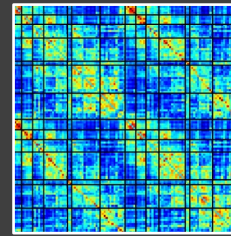
Pipeline output



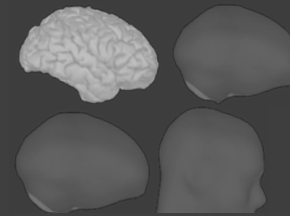
Structural connectivity



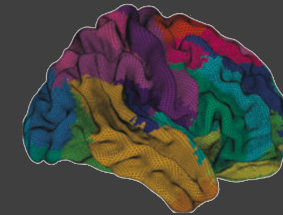
Region-wise fMRI



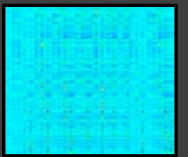
Functional connectivity



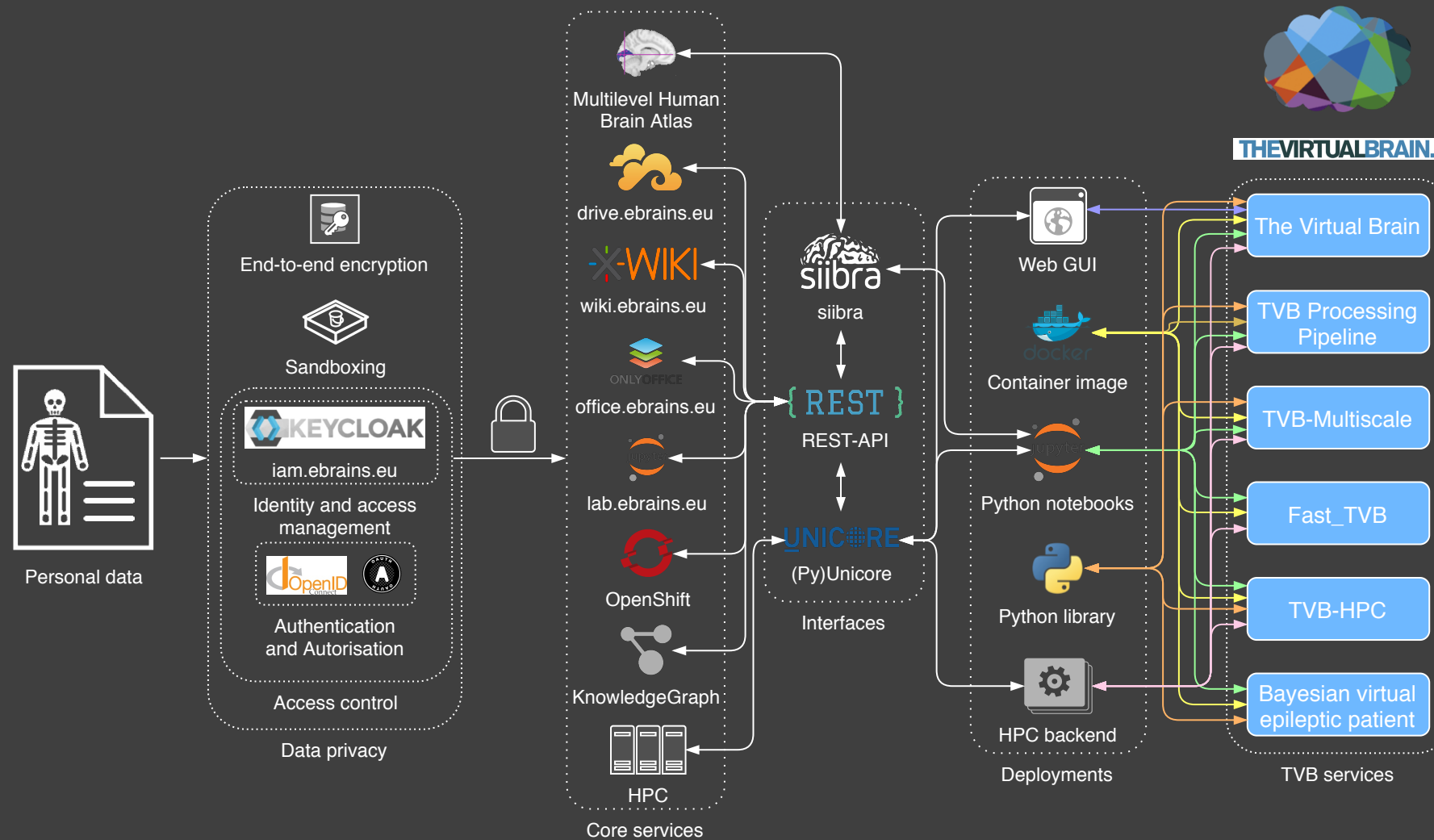
Surface triangulations



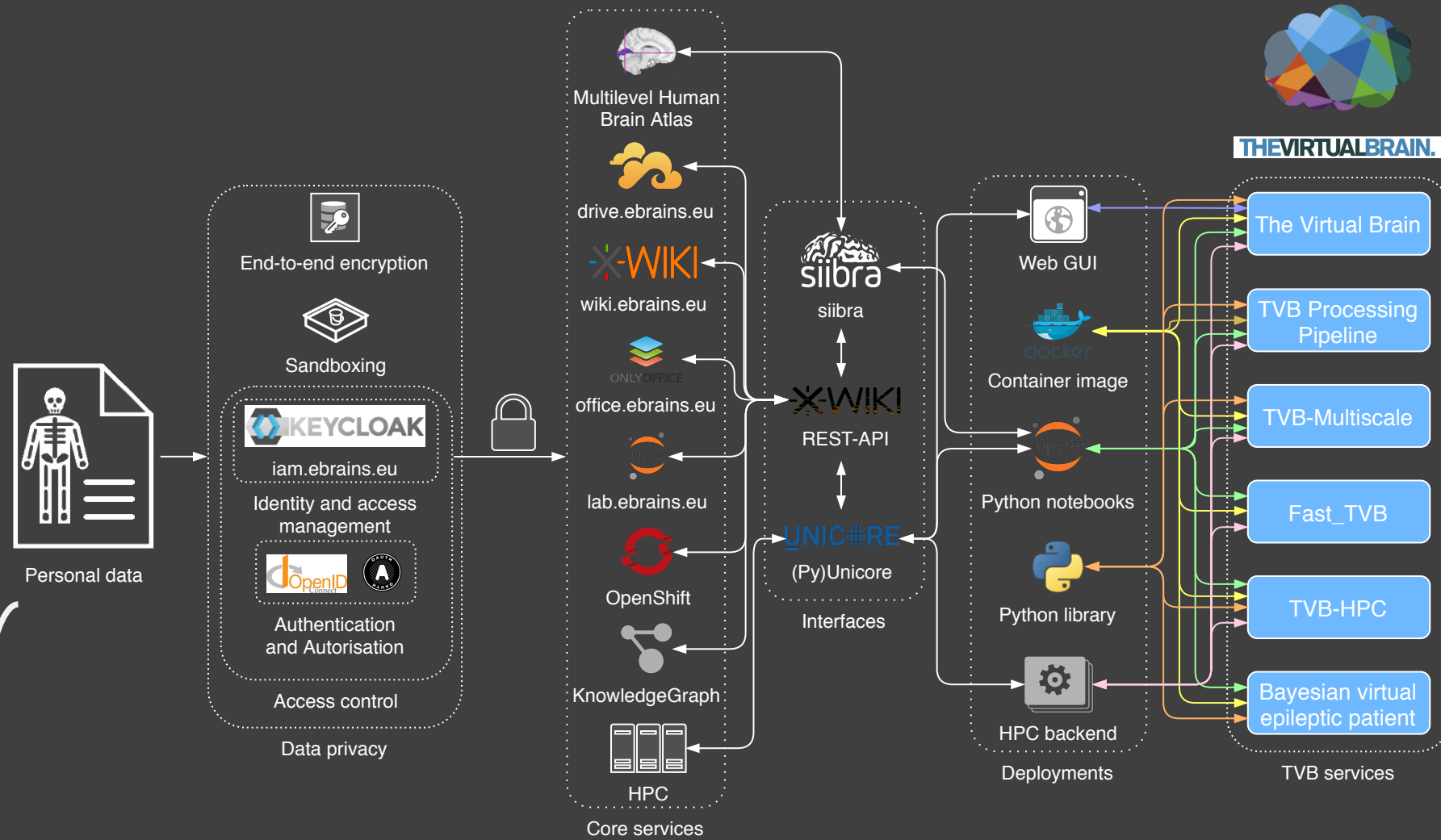
Brain maps



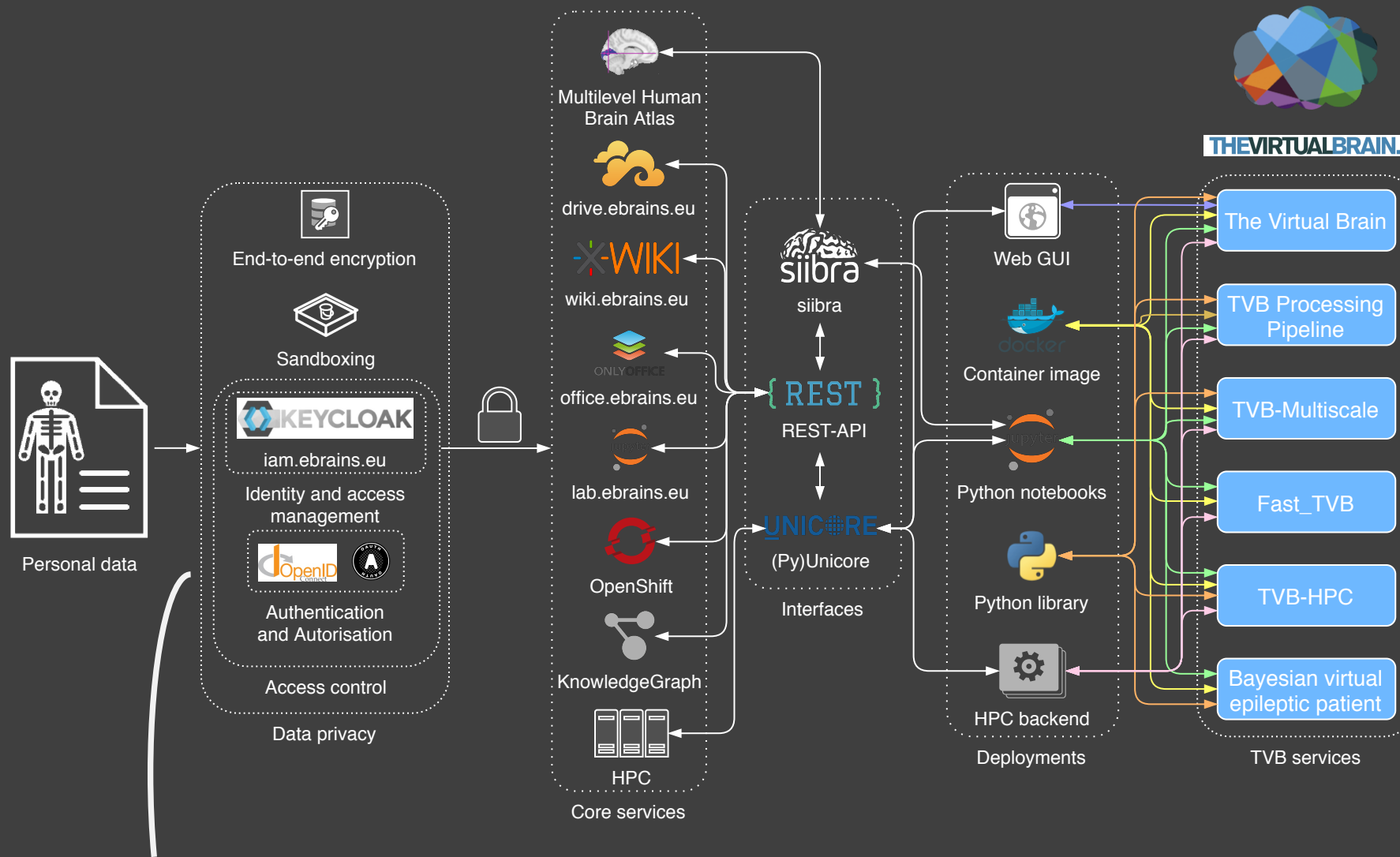
Projection matrices



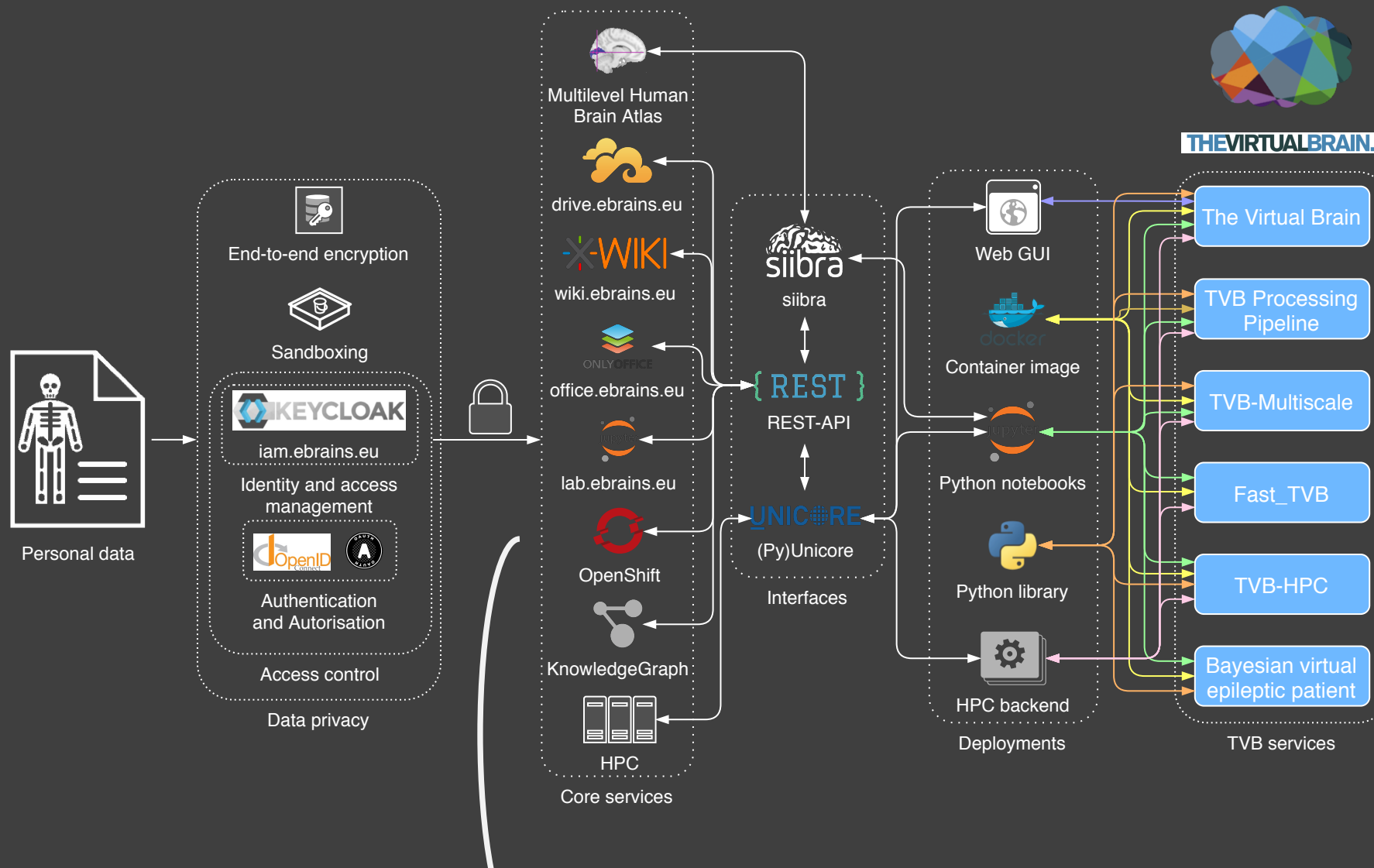
TVB-on-EBRAINS cloud infrastructure



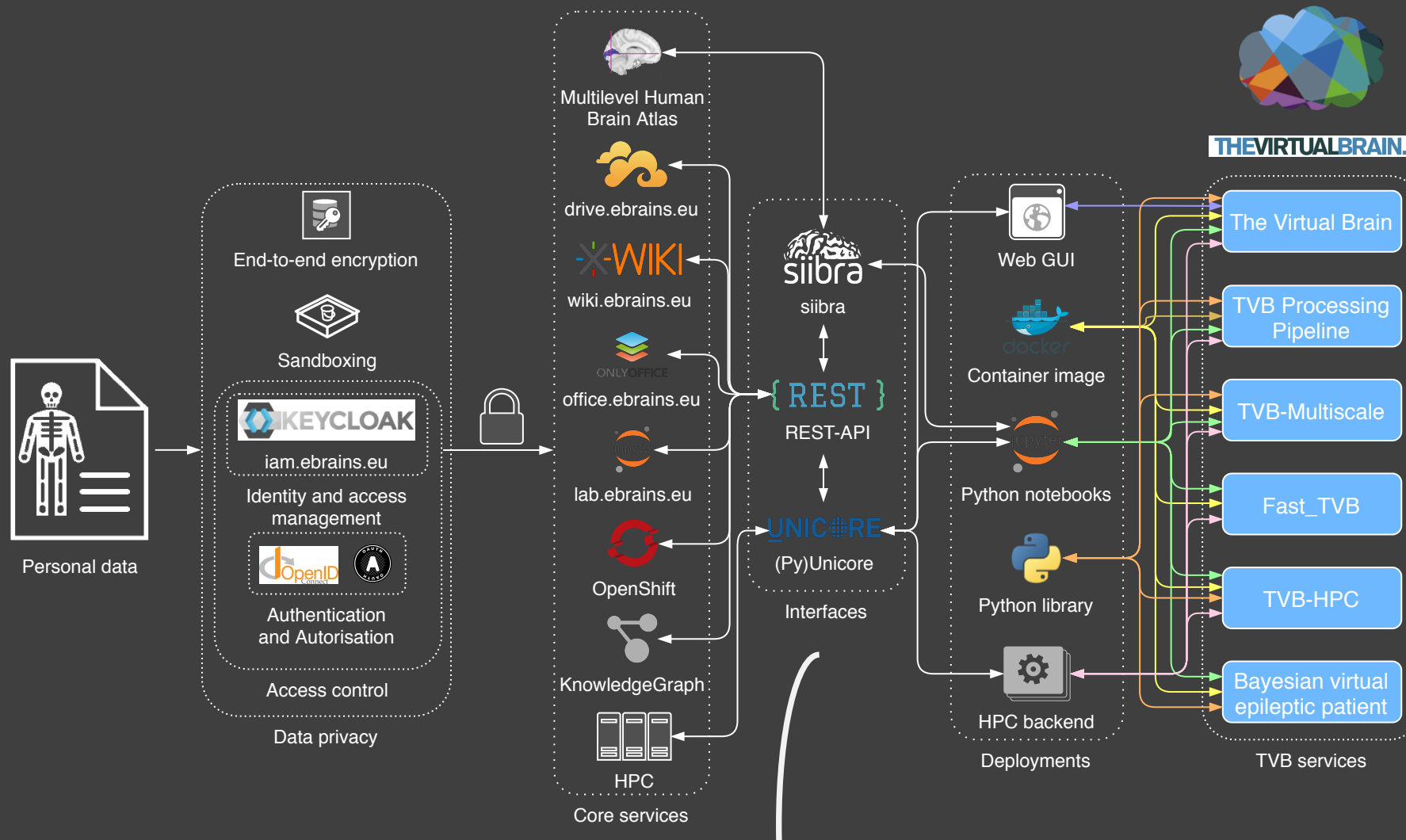
Brain modeling requires personal data subject to GDPR



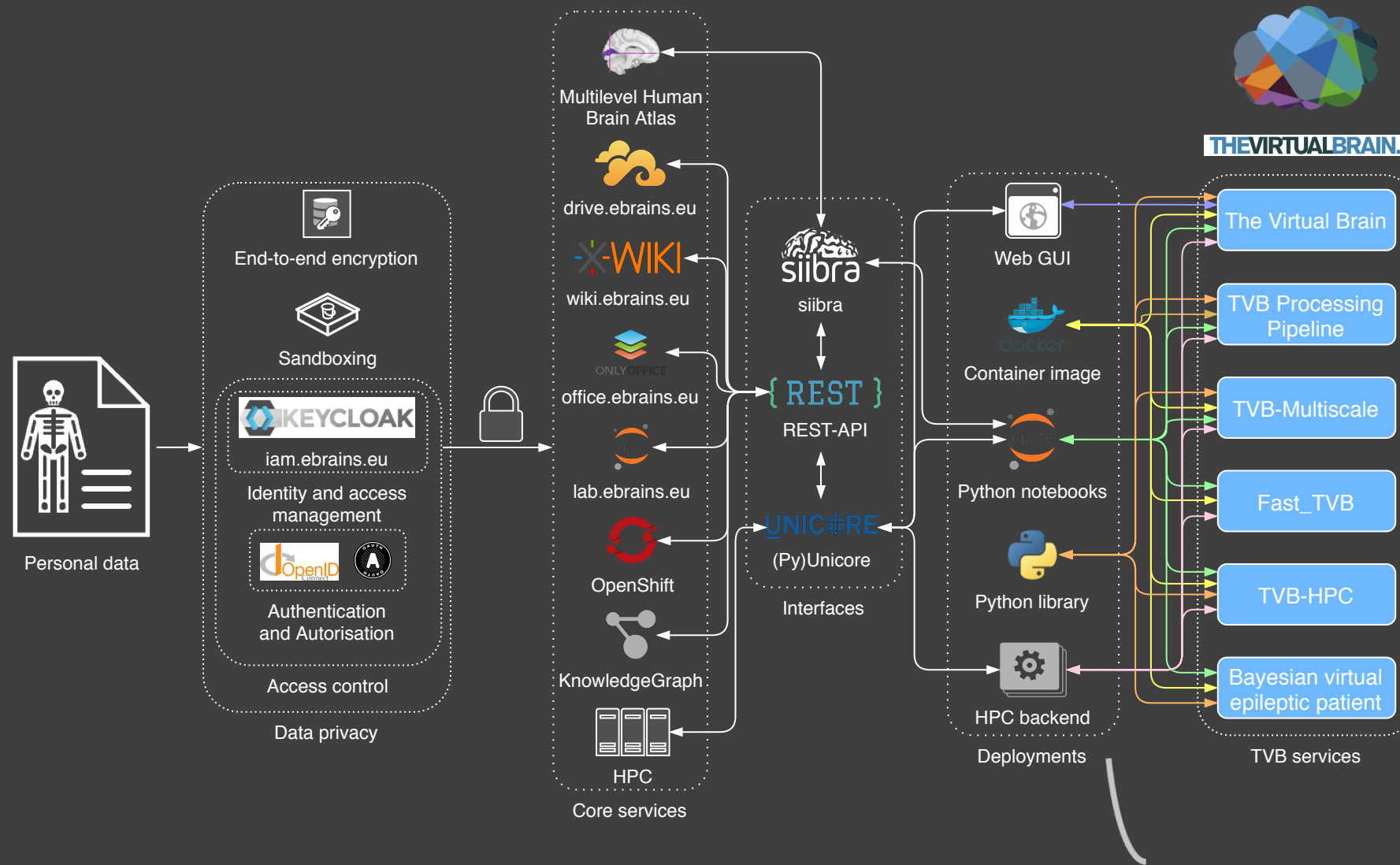
Access control, encryption and sandboxing protect personal data



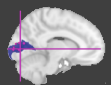
EBRAINS provides core services



Services are connected via RESTful API



TVB services can be used through a variety of deployments and interfaces



Multilevel Human Brain Atlas



drive.ebrains.eu



wiki.ebrains.eu



office.ebrains.eu



lab.ebrains.eu



OpenShift

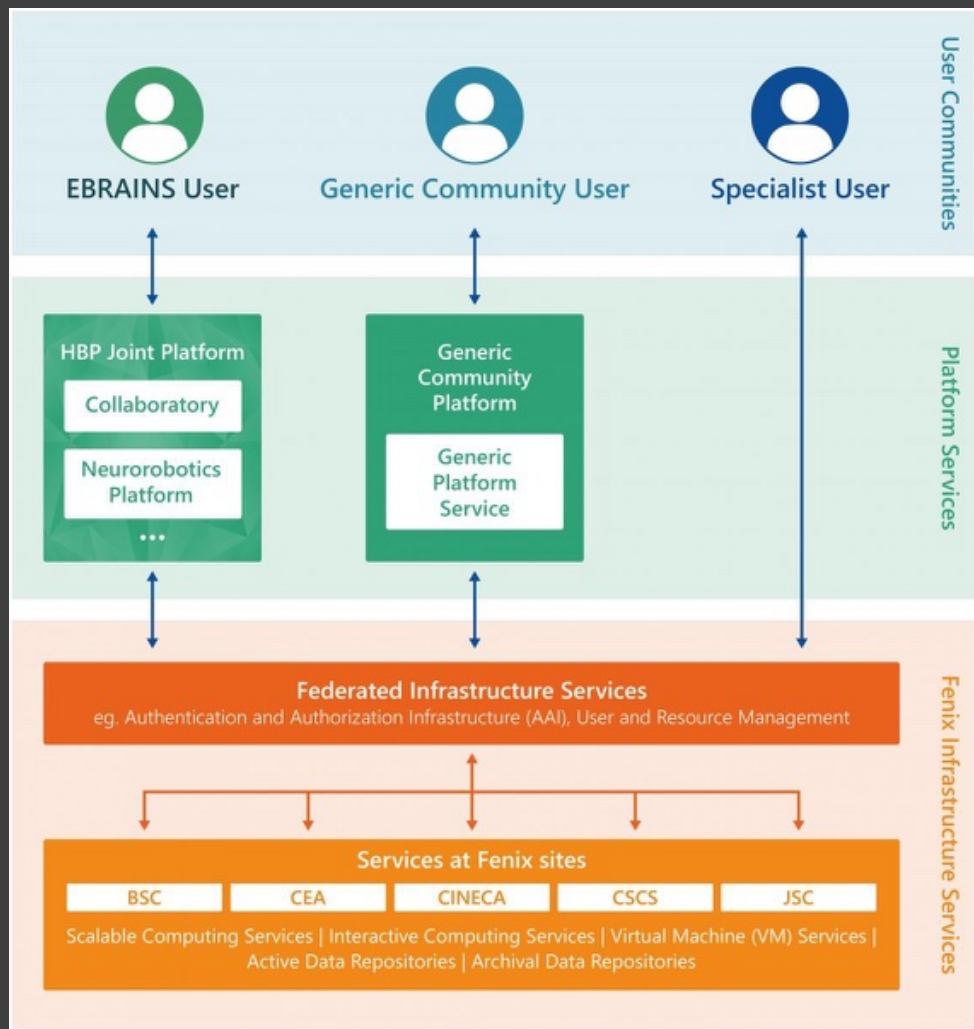


KnowledgeGraph



HPC

Core services

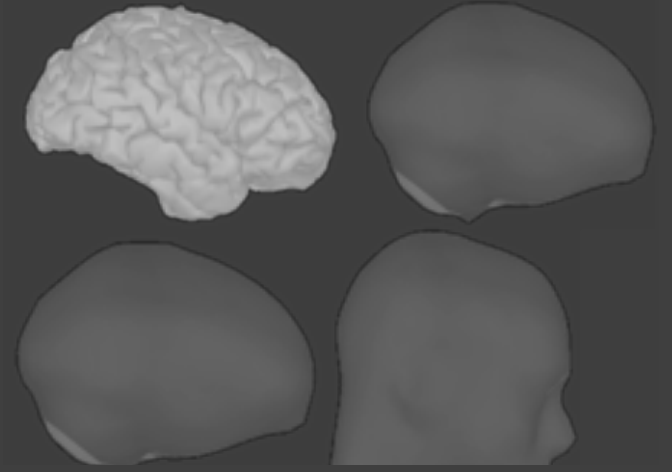
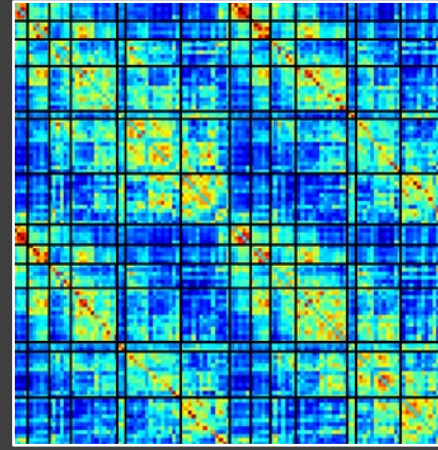
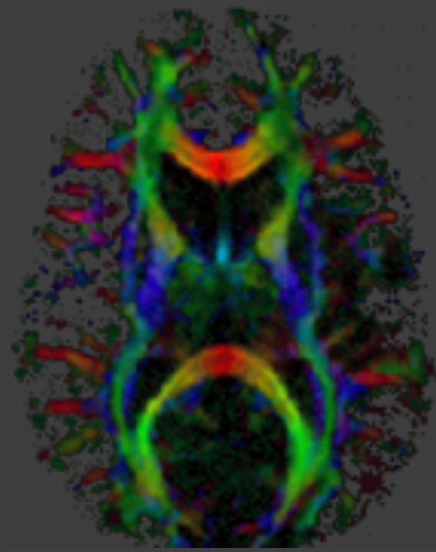
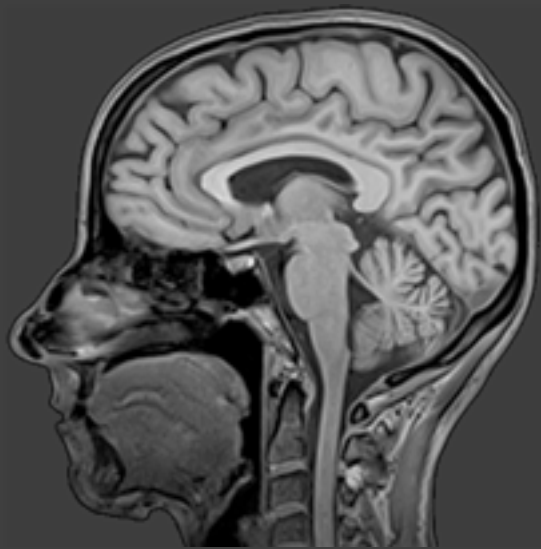


Data privacy

- Petabytes of valuable brain imaging data are generated every year.
- Openly sharing/processing human data on our shared infrastructures can compromise the privacy of the personal data of research participants.

BIG DATA





We have "fingerprints" all over our bodies



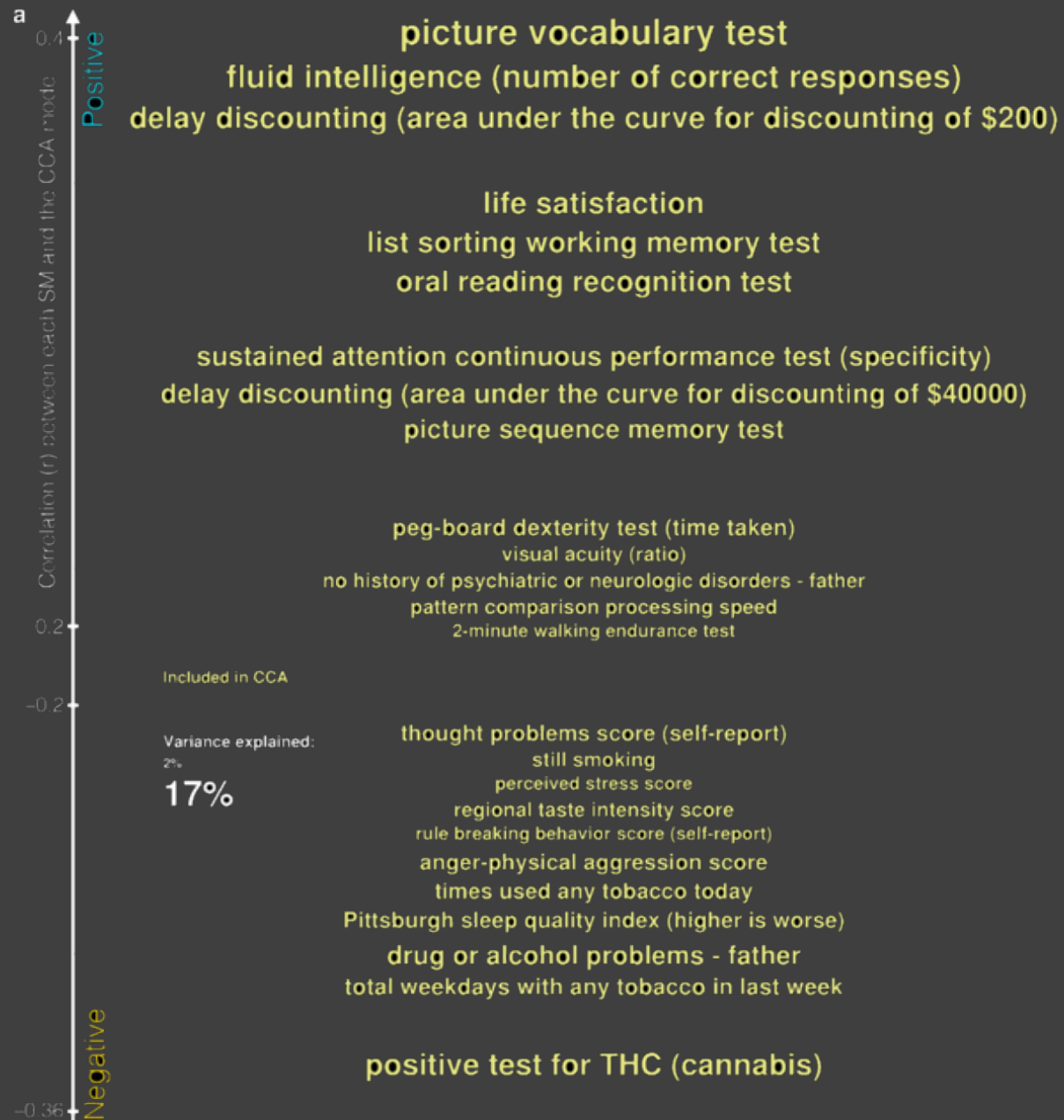
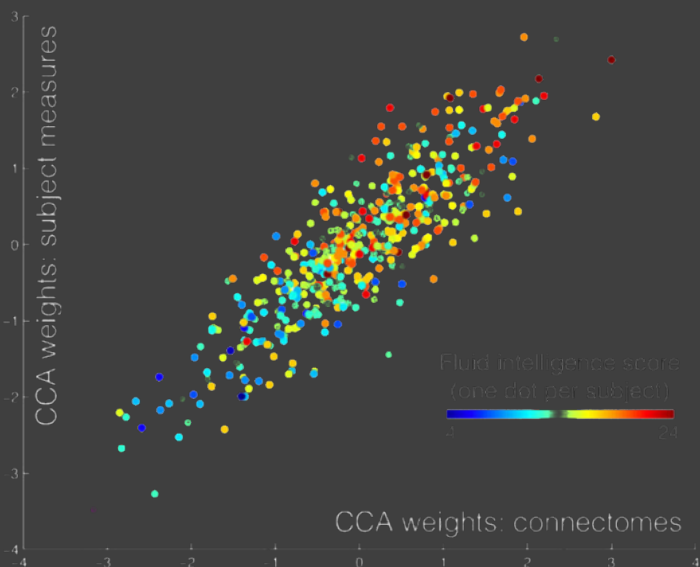
Rich personal data like MRI or connectomes cannot be easily pseudonymized and therefore require thorough protection.

MRI contains highly personal information

A positive-negative mode of population covariation links brain connectivity, demographics and behavior

Stephen M Smith¹, Thomas E Nichols², Diego Vidaurre³, Anderson M Winkler¹, Timothy E J Behrens¹, Matthew F Glasser⁴, Kamil Ugurbil⁵, Deanna M Barch⁴, David C Van Essen⁴ & Karla L Miller¹

We investigated the relationship between individual subjects' functional connectomes and 280 behavioral and demographic measures in a single holistic multivariate analysis relating imaging to non-imaging data from 461 subjects in the Human Connectome Project. We identified one strong mode of population co-variation: subjects were predominantly spread along a single 'positive-negative' axis linking lifestyle, demographic and psychometric measures to each other and to a specific pattern of brain connectivity.





NETWORK
NEURO
SCIENCE

High-accuracy individual identification using a “thin slice” of the functional connectome

Lisa Byrge and Daniel P. Kennedy

Department of Psychological and Brain Sciences, Indiana University, Bloomington, IN, USA

Keywords: Functional connectivity MRI, Individual differences, Single-subject fMRI, Resting state, Within-subject reliability

ABSTRACT

nature
COMMUNICATIONS

ARTICLE

<https://doi.org/10.1038/s41467-019-10933-3>

OPEN

Estimating the success of re-identifications in incomplete datasets using generative models

Luc Rocher ^{1,2,3}, Julien M. Hendrickx¹ & Yves-Alexandre de M.

Science

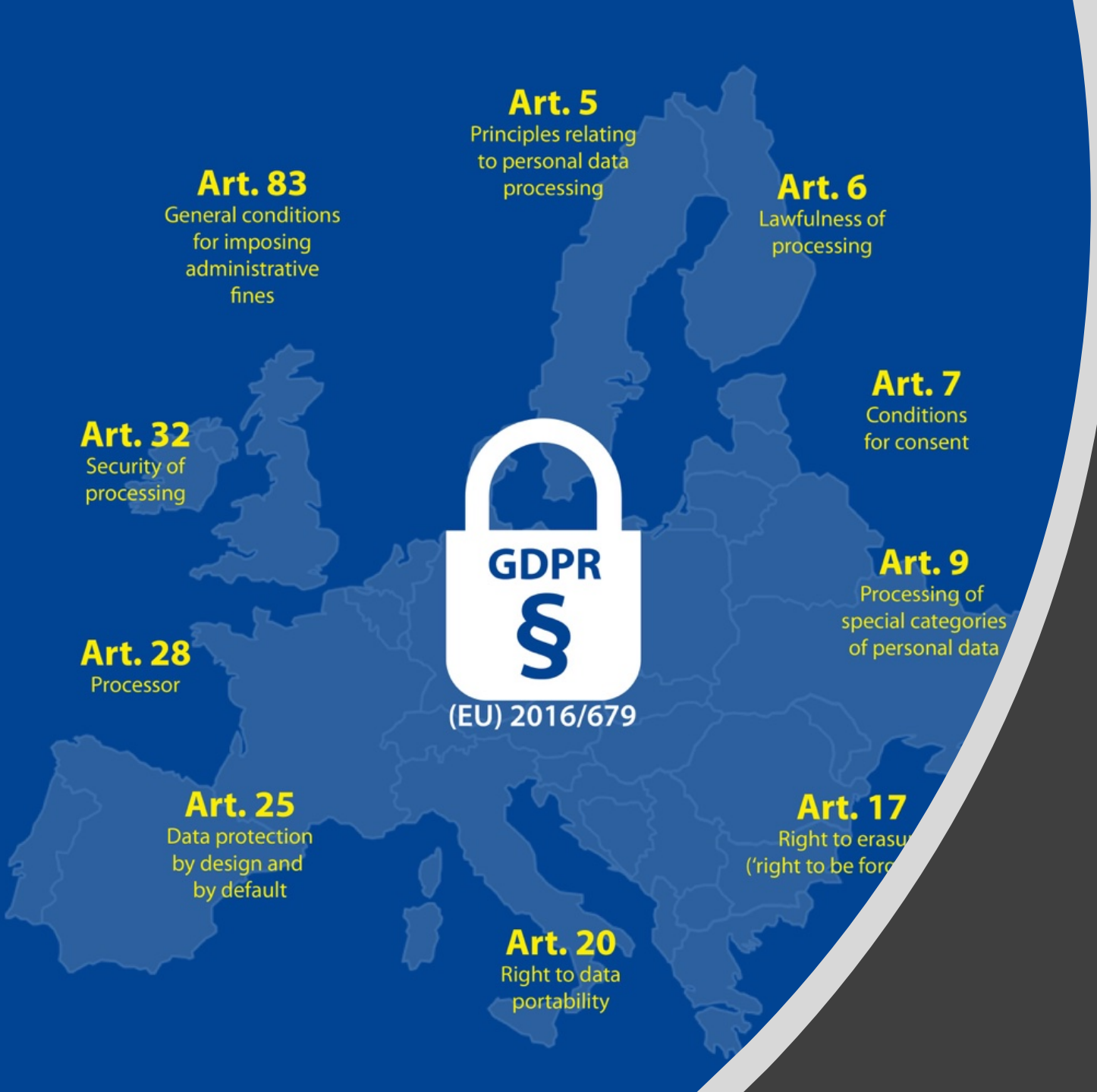


Identifying Personal Genomes by Surname Inference

Melissa Gymrek^{1,2,3,4}, Amy L. McGuire⁵, David Golan⁶, Eran Halperin^{7,8,9} et al.

+ See all authors and affiliations

It's often possible to re-identify us based on very little information. At the same time more and more records of us are accessible through open infrastructures (“clouds”).



Art. 5
Principles relating
to personal data
processing

Art. 6
Lawfulness of
processing

Art. 7
Conditions
for consent

Art. 9
Processing of
special categories
of personal data

Art. 17
Right to erasure
(‘right to be forgotten’)

Art. 20
Right to data
portability

Art. 83
General conditions
for imposing
administrative
fines

Art. 32
Security of
processing

Art. 28
Processor

Art. 25
Data protection
by design and
by default

GDPR
§
(EU) 2016/679

European Union General Data Protection Regulation (GDPR)



Art. 4 GDPR Definitions



Data subject gives **consent** that data is processed for scientific purposes



Controller determines **alone or jointly with others** the purposes and means of processing



Processor processes personal data **on behalf** of the data controller.



Art. 24 GDPR

Responsibility of the controller

“...shall implement appropriate technical and organisational measures to ensure and...demonstrate that processing is performed in accordance with this Regulation”



Controller determines alone or jointly with others the purposes and means of processing

Responsible for protecting the personal data!



- Art. 25 (Data protection by design and by default)
 - **“by default personal data are not made accessible without the individual’s intervention to an indefinite number of natural persons.”**
- Art. 32 (Security of processing):
 - Controller and processor shall implement T+O measures to mitigate risks
 - Pseudonymisation, encryption, resilience of systems, restore availability, regular testing
 - Adherence to code of conduct and certification may used to demonstrate compliance
 - Prevent unlawful processing by people (staff, etc.) who have access to the data

Anonymity



Pseudonymity



Real Identity



Information about race, ethnic origin, sexual orientation, political views, religious beliefs, genes, biometrics, health, etc. are sensitive personal data processing of which is generally prohibited.



Art. 9 GDPR

Processing of special categories of personal data



1. Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited.
2. Paragraph 1 shall not apply if one of the following applies:
 - (a) the data subject has given explicit consent to the processing of those personal data for one or more specified purposes, except where Union or Member State law provide that the prohibition referred to in paragraph 1 may not be lifted by the data subject;



Art. 9 GDPR

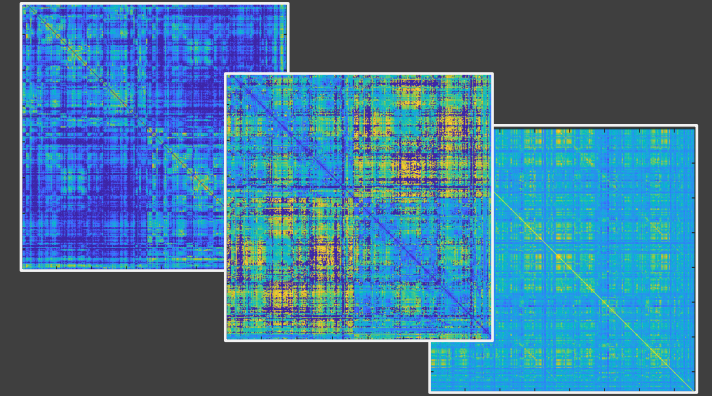
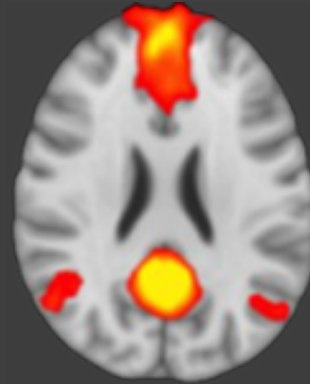
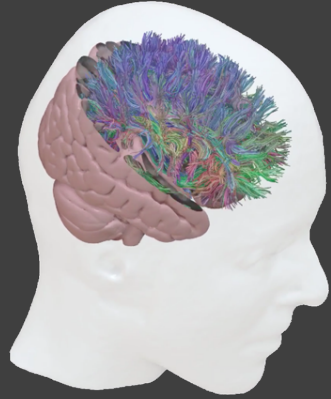
Processing of special categories of personal data



(j) processing is necessary for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) based on Union or Member State law which shall be proportionate to the aim pursued, respect the essence of the right to data protection and provide for suitable and specific measures to safeguard the fundamental rights and the interests of the data subject.

“Processing for...scientific...research...shall be subject to appropriate safeguards...technical and organisational measures...ensure respect for the principle of data minimisation.”

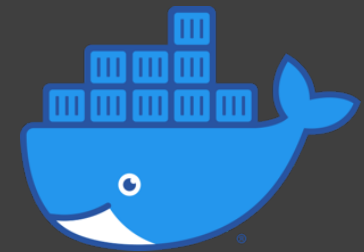
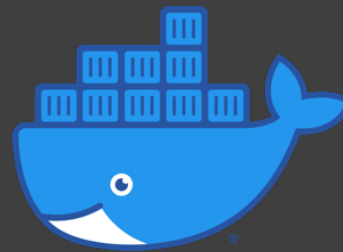
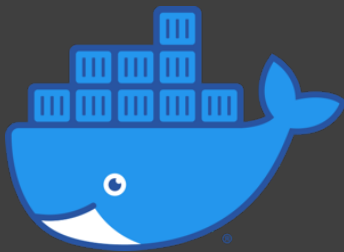
TVB Image Processing Pipeline: container workflows



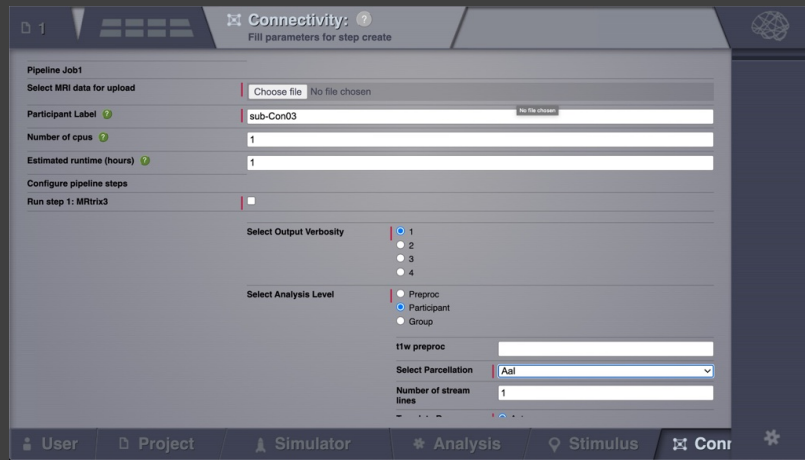
`bids/mrtrix3_connectome`

`nipreps/fmriprep`

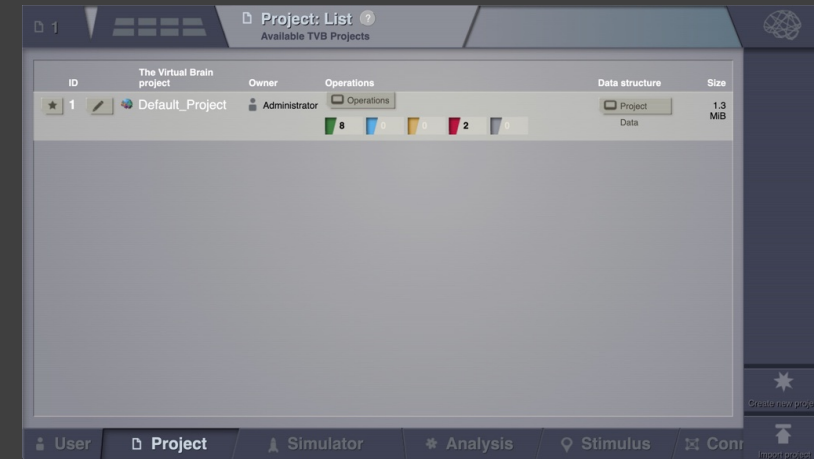
`thevirtualbrain/tvb-
pipeline-converter`



Configure processing with GUI



Download results



Orchestrator runs containers on supercomputer, stores provenance information with DataLad

Data privacy risks



Intercept during transit



Break-in on device



User impersonation

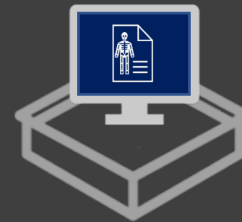
Data protection



encryption



access control

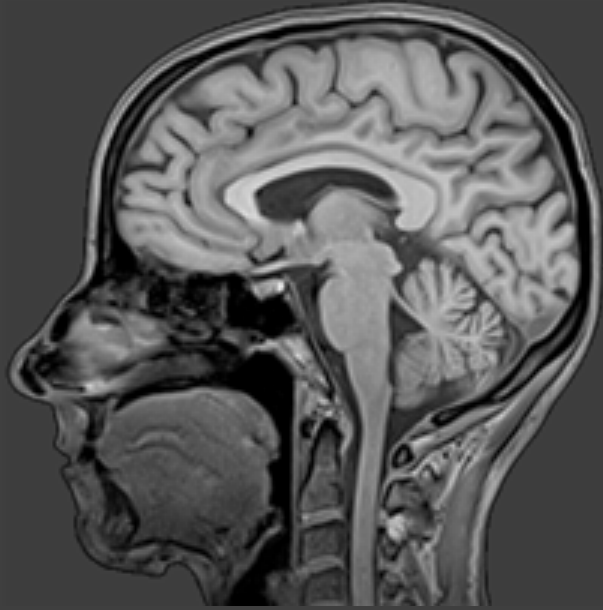


sandboxing



automation

Protecting cloud workflows: Encryption



2BpfrRRC%2FtkuF01V%2F35n
mKB%2F7UmTrKuWaR9%2Fu9
84jcWvPUPy%2B2V%2B6%2B
NaJ%2FI0%2FI0TKvwLyi3TtP36
a7i4ogdm%2Fd1iv84T%2F529f
zzyUozbf%2BYE5NcJR9Lvv%

- Sensitive data is always encrypted by default and only decrypted for the actual processing
- In shared environments unencrypted data is sandboxed during processing


Protecting workflows in shared environments: Encryption



- Personal data is encrypted before it leaves the computer of the data controller
- Results are encrypted before being returned to the data controller
- The keys for decryption are created ad-hoc shortly before the processing and stay always in a safe location at the destination site of the data
- All (intermediate) processing results are only written inside sandboxed file systems and securely deleted before termination.

Protecting cloud workflows: Access control

iam.ebrains.eu/auth/realms/hbp/protocol/openid-connect/auth?client_id=tvb-web&redirect_uri=https%3A%2F%2Fthevirtualbrain.apps.hbp.eu%2Fuser%2F&state=2a441653-1fde-44ef-8028-56...




EBRAINS

Log In

Username or email

Password

[Log In](#) [Forgot Password?](#) [Register](#)



Human Brain Project

©2020 Human Brain Project. All rights reserved.



LEAD
INSTITUTIONS

Aix*Marseille
université

Baycrest

CHARITÉ
UNIVERSITÄTSMEDIZIN BERLIN

USER DETAILS

DISPLAY NAME	Michael Schirner
USERNAME	michaels
ROLE	administrator manage other users
DATA	0.0 KiB
ACCOUNT	Account management console

ABOUT THIS VERSION

- ★ You are running TVB version **2.4.1-16969**
- ★ You may use this version on your **personal computer**.
- ★ You may also install it on a **server** and make it accessible to an unlimited number of users.









Please make sure that your server has appropriate hardware resources like a decent multi-core CPU and at least 16 GB of RAM.
- ★ You may also install it on a **cluster** (similar to a server installation but with parallelization support).


Please note that for cluster installations, OAR is expected to be configured separately from TVB.




Logout



ID	The Virtual Brain project	Owner	Operations	Data structure	Size
 1	  Default_Project	 Administrator	Operations  8  0  0  2  0	 Project Data	1.3 MiB


Create new project


Import project structure



Large Scale Connectivity

View Connectivity Regions. Perform Connectivity lesions



Allen Connectome Builder

Download data from Allen dataset and create a mouse connectome



Local Connectivity

Create or view existent Local Connectivity entities.



Image Preprocessing Pipeline

Launch Image Preprocessing pipeline

Image Preprocessing Pipeline



Pipeline Job1

Select MRI data for upload

Choose file No file chosen

Participant Label ?

sub-Con03

Number of cpus ?

1

Estimated runtime (hours) ?

1

Configure pipeline steps

Run step 1: MRtrix3

Select Output Verbosity

- 1
- 2
- 3
- 4

Select Analysis Level

- Preproc
- Participant
- Group

tlw preproc

Run step 2: fmriprep

Skip Bids Validation

Anat Only

No Reconall

6 DoF

MNI normalization

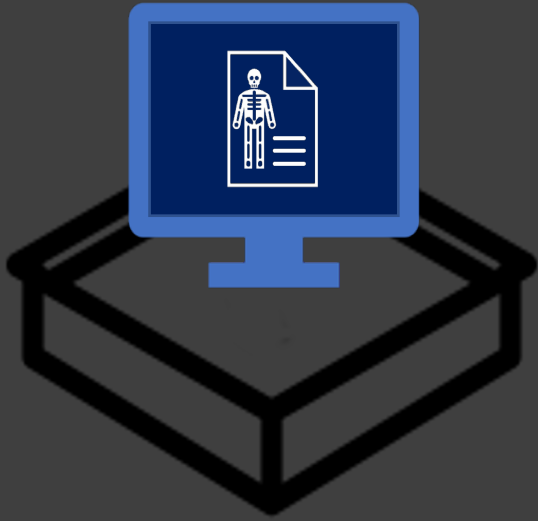
Run step 3: freesurfer

Run step 4: tvb-pipeline-converter



Launch

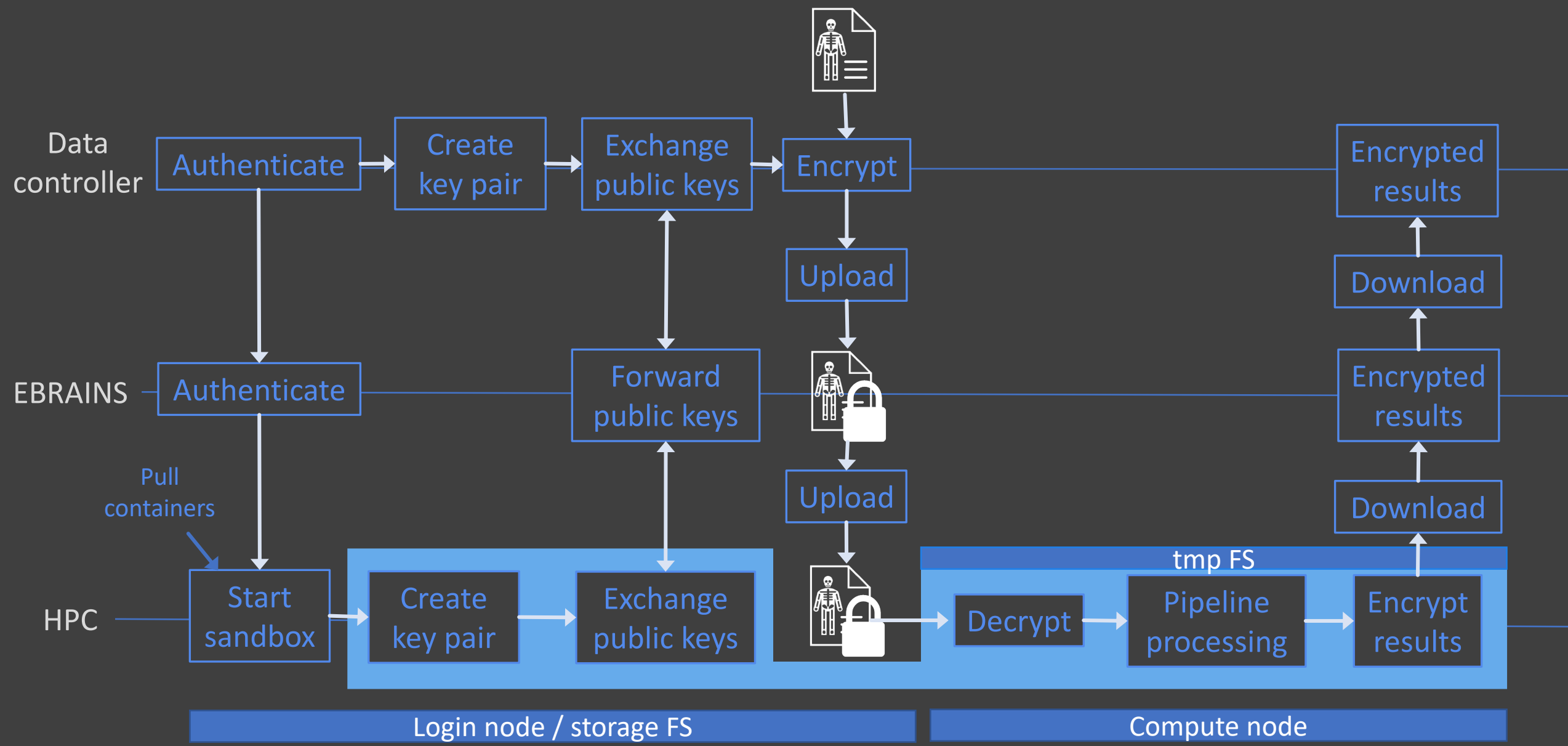
Protecting cloud workflows: Sandboxing



- Provide only a controlled set of resources (storage, network, memory, CPU, devices) to a process.
- Often used to isolate potentially malicious software from the system to avoid harm.

- In shared environments unencrypted personal data and secrets may only be written into a temporary filesystem namespace that is entirely invisible from the host and which is fully and securely cleaned up when the last process exits.

Automation: cryptography & sandboxing workflow



168 lines (155 sloc) | 8.52 KB

```
1 FROM ubuntu:18.04
2 MAINTAINER Robert E. Smith <robert.smith@florey.edu.au>
3
4 # Core system capabilities required
5 RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get
6     bc \
7     build-essential \
8     curl \
9     dc \
10    git \
11    libegl1-mesa-dev \
12    libopenblas-dev \
13    nano \
14    perl-modules-5.26 \
15    python2.7 \
16    python3 \
17    tar \
18    tcsh \
19    tzdata \
20    unzip \
21    wget
22
23 # PPA for newer version of nodejs, which is required for bids
24 RUN curl -sL https://deb.nodesource.com/setup_12.x -o nodesou
25     bash nodesource_setup.sh && \
26     rm -f nodesource_setup.sh && \
27     apt-get install -y nodejs
28
29 # NeuroDebian setup
```

```
137 SUBJECTS_DIR=/opt/freesurfer/subjects \
138 FSLDIR=/opt/fsl \
139 FSLOUTPUTTYPE=NIFTI \
140 FSLMULTIFILEQUIT=TRUE \
141 FSLTCLSH=/opt/fsl/bin/fsltcclsh \
142 FSLWISH=/opt/fsl/bin/fslwish \
143 LD_LIBRARY_PATH=/opt/fsl/lib:$LD_LIBRARY_PATH \
144 PATH=/opt/mrtrix3/bin:/usr/lib/ants:/opt/freesurfer/bin:/opt/freesurfer/mn
145 PYTHONPATH=/opt/mrtrix3/lib:$PYTHONPATH
146
147 # MRtrix3 setup
148 # Commitish is 3.0.2 plus relevant hotfix
149 RUN git clone https://github.com/MRtrix3/mrtrix3.git /opt/mrtrix3 && \
150     cd /opt/mrtrix3 && \
151     git checkout 4ab54489f40997f7da1e1915c2adde3373cf6039 && \
152     python3 configure -nogui && \
153     python3 build -persistent -nopaginate && \
154     git describe --tags > /mrtrix3_version && \
155     rm -rf .git/ cmd/ core/ src/ testing/ tmp/ && \
156     cd /
157
158 # Acquire extra MRtrix3 data
159 RUN wget -q "https://osf.io/v8n5g/download" -O /opt/mrtrix3/share/mrtrix3/labe
160     wget -q "https://osf.io/ug2ef/download" -O /opt/mrtrix3/share/mrtrix3/labe
161
162 # Acquire script to be executed
163 COPY mrtrix3_connectome.py /mrtrix3_connectome.py
164 RUN chmod 775 /mrtrix3_connectome.py
165
166 COPY version /version
167
168 ENTRYPOINT ["/mrtrix3_connectome.py"]
```



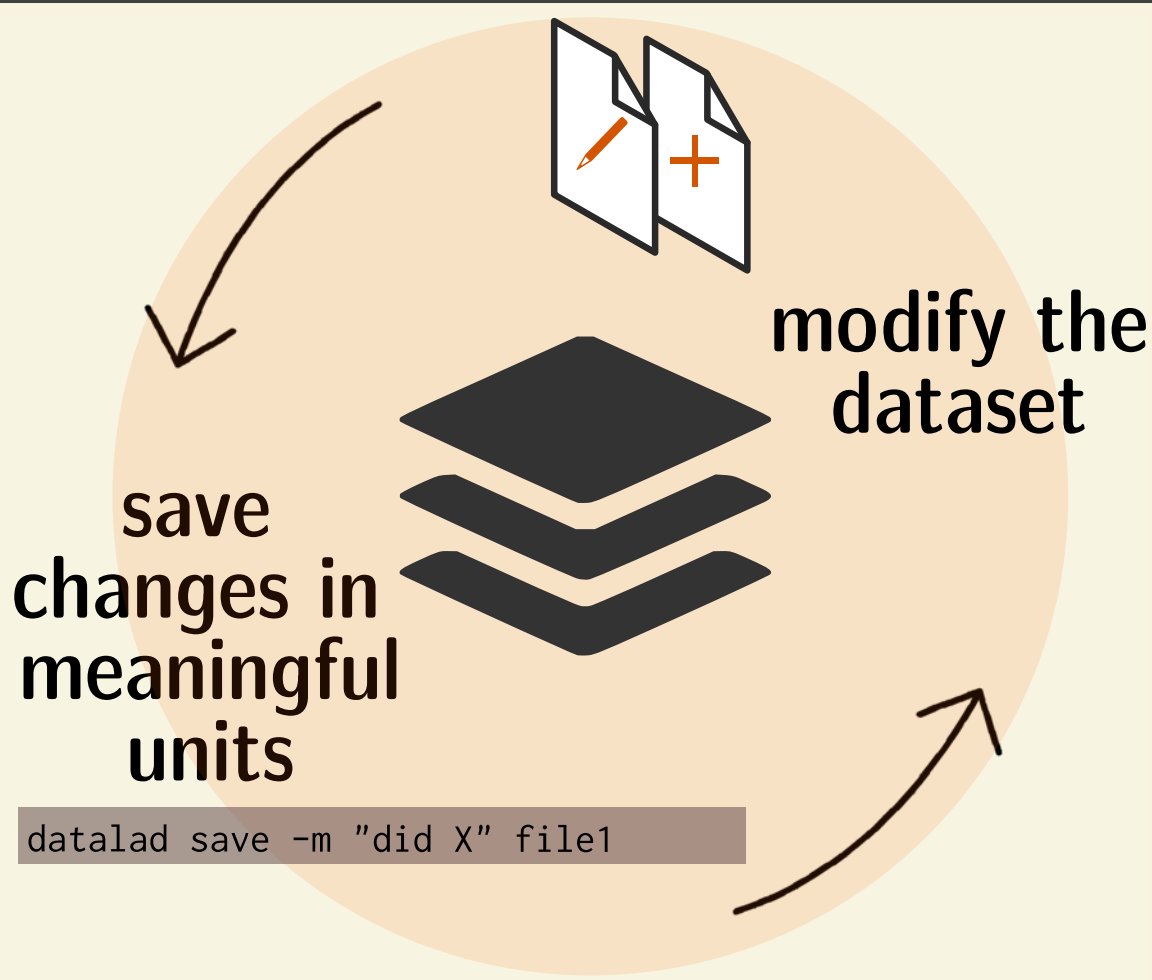

- “Git for data”
- based on **Git** and **git-annex**
- command-line tool for Linux, OS X, Windows
- **version control** arbitrarily large files
- **provenance tracking**
- **share and obtain data**
- **reproducible workflows**
- **domain-agnostic**



- DataLad datasets
 - are directories managed by DataLad
 - are Git repositories
 - can be nested: linked subdirectories
 - can be created or installed
 - can branch and merge



Local version control



datalad create

Create empty dataset

datalad save

record state of dataset to history incl. commit message

datalad download-url

obtain web content and record origin

datalad status

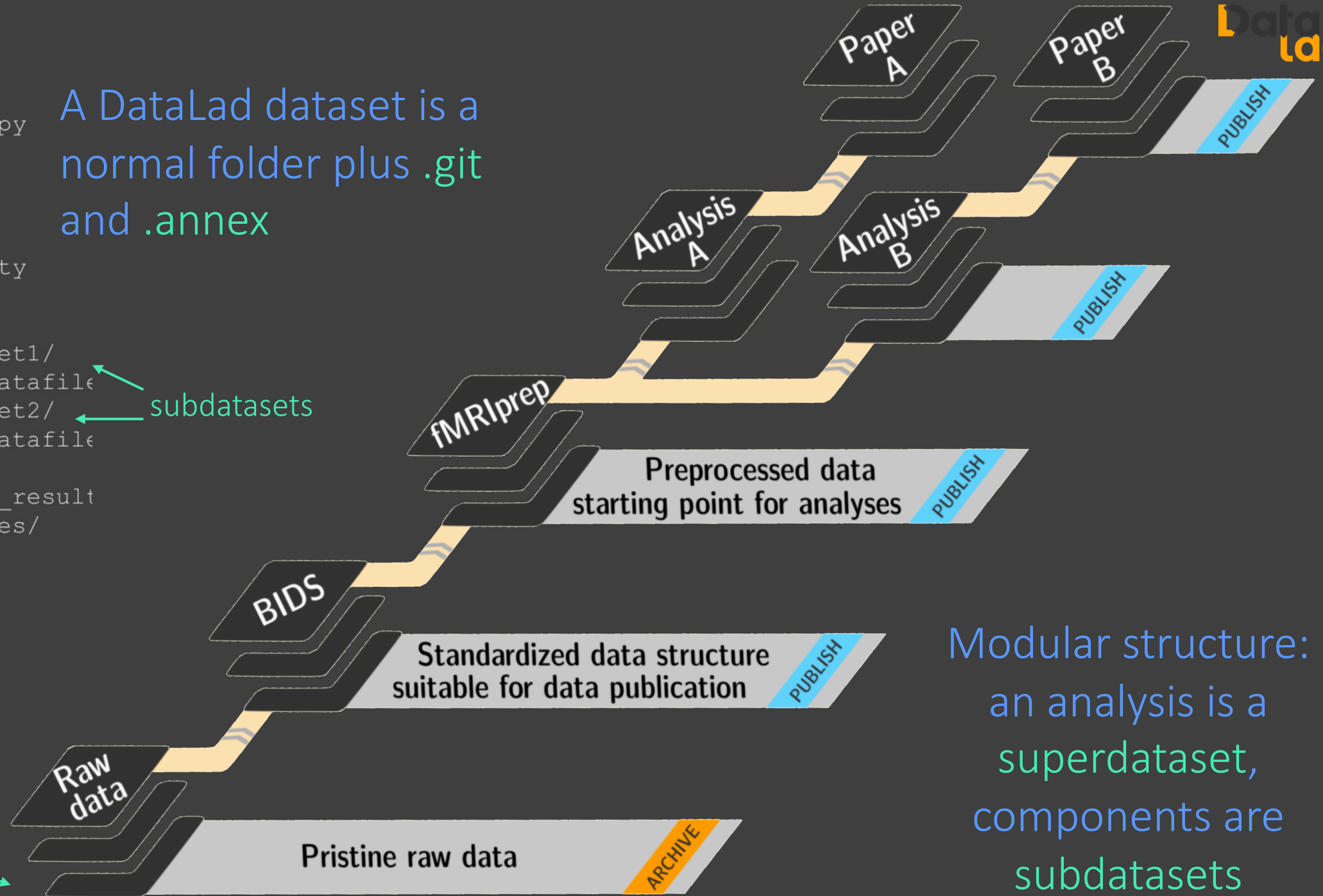
report tracking state

```
superdataset  
├── code/  
│   ├── tests/  
│   └── myscript.py  
├── docs  
│   ├── build/  
│   └── source/  
├── envs  
│   └── Singularity  
├── inputs/  
│   └── data/  
│       ├── dataset1/  
│       │   └── datafile  
│       └── dataset2/  
│           └── datafile  
├── outputs/  
│   ├── important_result  
│   └── figures/  
├── README.md  
├── .annex  
└── .git
```

A DataLad dataset is a normal folder plus `.git` and `.annex`

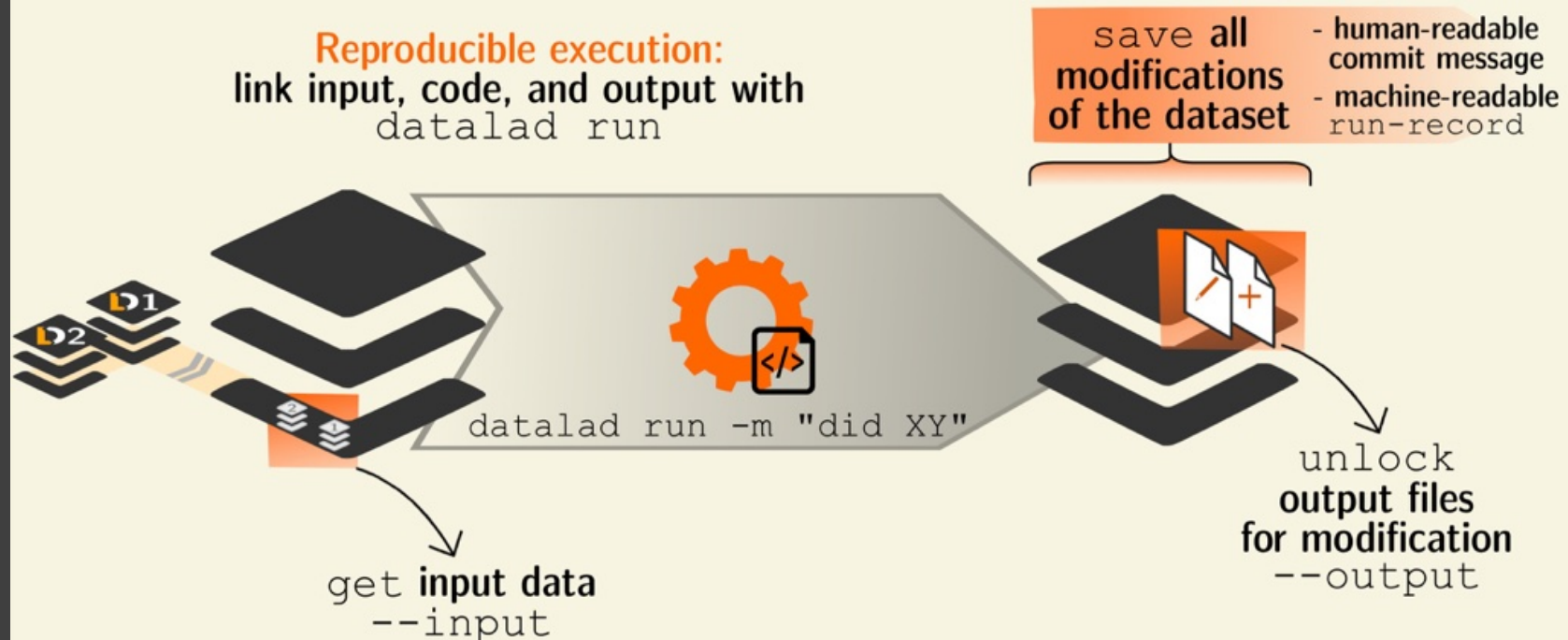
subdatasets

superdataset
subdatasets

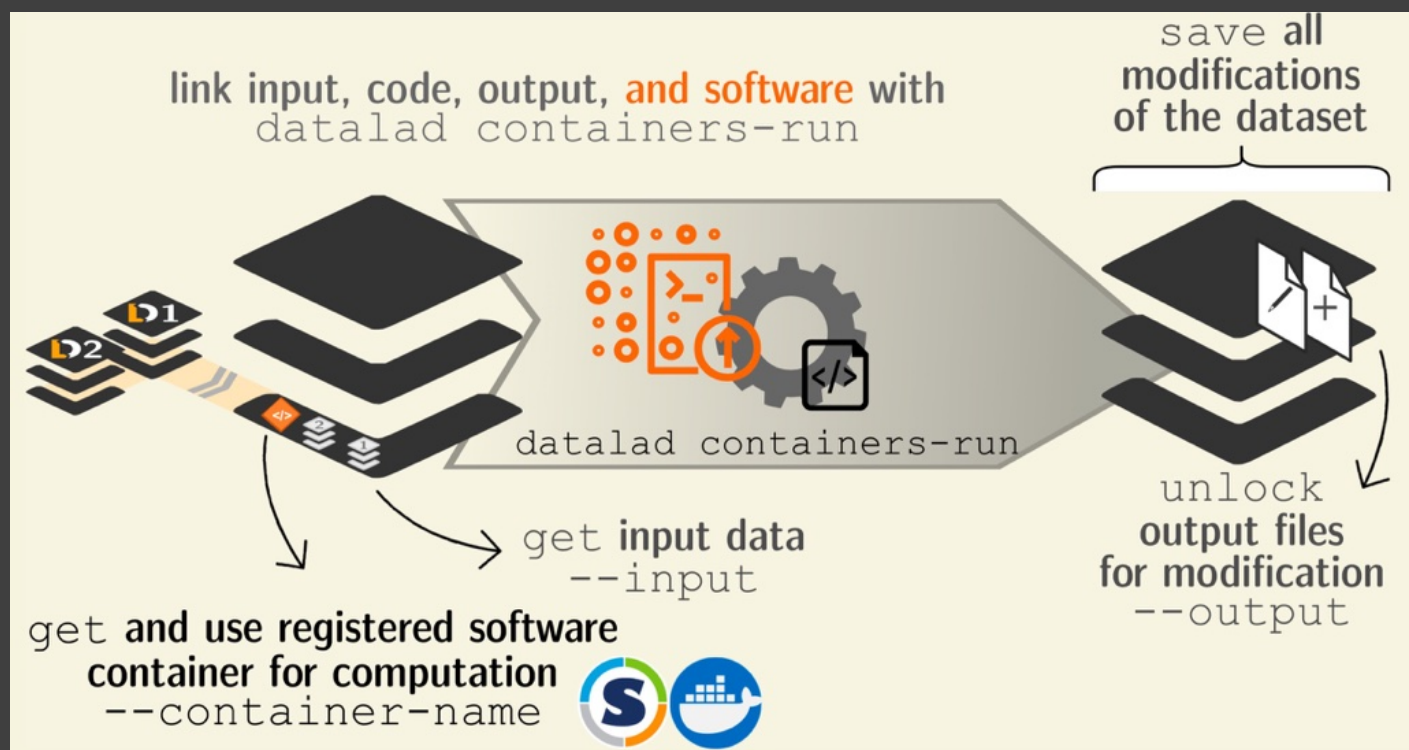


Modular structure:
an analysis is a
superdataset,
components are
subdatasets

Record where you got it from, where it is now, and what you did to it



Containerization captures software and environment as provenance



```
$ datalad run -m "analyze iris data with classification analysis" \  
  --input "input/iris.csv" \ retrieved on demand  
  --output "prediction_report.csv" \  
  --output "pairwise_relationships.png" \ } only specified output  
  "python3 code/script.py"                files get unlocked  
[INFO ] Making sure inputs are available (this may take some time)  
get(ok): input/iris.csv (file) [from web...]  
[INFO ] == Command start (output follows) =====  
[INFO ] == Command exit (modification check follows) =====  
add(ok): pairwise_relationships.png (file)  
add(ok): prediction_report.csv (file)  
save(ok): . (dataset)  
action summary:  
  add (ok: 2)  
  get (notneeded: 2, ok: 1)  
  save (notneeded: 1, ok: 1)
```

Execute the analysis script
in a computationally
reproducible manner

Date	Author	Change summary (commit message)
2020-06-05 10:58 +0200	Adina Wagner	M [master] {upstream/master} {upstream/HEAD} Merge pull request #12 from psychoinformatics-d
2020-06-05 08:24 +0200	Adina Wagner	o [finalround] {upstream/finalround} add results from computing with mean instead of median
2020-06-05 09:09 +0200	Michael Hanke	o Change wording, clarify comment
2020-06-05 07:26 +0200	Michael Hanke	M Merge remote-tracking branch 'gh-mine/finalround'
2020-05-28 16:39 +0200	Asim H Dar	o Added datalad.get() so S2SRMS() pulls data and can run standalone
2020-05-18 08:25 +0200	Adina Wagner	o {gh-asim/finalround} S2SRMS: example implementation of the S2SRMS method suggested by R2
2020-05-01 17:38 +0200	Adina Wagner	o Minor edits as suggested by reviewer 2
2020-05-29 09:04 +0200	Adina Wagner	M Merge pull request #13 from psychoinformatics-de/adswa-patch-1
2020-05-24 09:15 +0200	Adina Wagner	o {upstream/adswa-patch-1} Fix installation instructions
2020-05-24 09:53 +0200	Adina Wagner	M Merge pull request #14 from psychoinformatics-de/bf-data
2020-05-24 09:33 +0200	Adina Wagner	o [bf-data] One-time datalad import
2020-05-24 09:32 +0200	Adina Wagner	o install and get relevant subdataset data
2020-03-18 10:19 +0100	Michael Hanke	M Merge pull request #8 from psychoinformatics-de/adswa-patch-1
2019-12-19 10:22 +0100	Adina Wagner	o {gh-asim/adswa-patch-1} add sklearn to requirements
2020-03-18 10:13 +0100	Michael Hanke	o Tune new figure caption
2020-03-18 10:03 +0100	Michael Hanke	M Merge pull request #11 from ElectronicTeaCup/revision_2
2020-03-18 09:59 +0100	Adina Wagner	M [revision_2] {gh-asim/revision_2} Merge branch 'revision_2' of github.com:ElectronicTeaCup/revision_2
2020-03-18 09:58 +0100	Michael Hanke	o Last detection
2020-03-18 09:59 +0100	Adina Wagner	o name parameter in caption

- the dataset logs everything that was done, by whom & when
- reset dataset to historic state
- add, remove or revisit individual steps

```
$ git log
commit df2dae9b5af184a0c463708acf8356b877c511a8 (HEAD -> master)
Author: Adina Wagner <adina.wagner@t-online.de>
Date: Tue Dec 1 11:58:18 2020 +0100
```

```
$ datalad rerun df2dae9b5af1
datalad rerun df2dae9b5af18
[INFO ] run commit df2dae9; (analyze iris data...)
[INFO ] Making sure inputs are available (this may take some time)
unlock(ok): pairwise_relationships.png (file)
unlock(ok): prediction_report.csv (file)
[INFO ] == Command start (output follows) =====
[INFO ] == Command exit (modification check follows) =====
add(ok): pairwise_relationships.png (file)
add(ok): prediction_report.csv (file)
```

```
action summary:
```

```
add (ok: 2)
get (notneeded: 3)
save (notneeded: 2)
unlock (ok: 2)
```

- rerun recorded computations via commit hash
- helps to reproduce and verify a result (via content hash)
- granularity can be freely decided
- recompute: not everything needs to be stored or transported

Create an input dataset



If your data is not yet a DataLad dataset, you can transform it into one with the following commands.

```
# create a dataset in an existing directory  
$ datalad create -f .  
# save its contents  
$ datalad save . -m "Import all data"
```

*For large datasets (e.g. HCP, UK Biobank),
we create subdataset for each subject*

Create a new DataLad dataset

First we create a new empty dataset and then add existing containers to it.

```
$ datalad create pipeline  
$ cd pipeline
```

Add container(s)

Add a software-container to the dataset using *datalad containers-add* from the *datalad-container* extension. The *--url* parameter can be a local path to your container image, or a URL to a container hub such as Dockerhub or Singularity Hub.

```
$ datalad containers-add cat --url<path/or/url/to/image> \  
--call-fmt "singularity run -B {{pwd}} --cleanenv {img} {cmd}"
```



Re-usable
container
pipeline
dataset

Create empty dataset and then clone and add the container as subdataset

Here we create an empty dataset and then clone and register our needed container(s) as a subdataset ([documentation](#)). Arguments like `--bind` that are intended for singularity rather than the underlying command should be specified with `--call-fmt` when calling `containers-add`. It's also fine to edit the `cmdexec` value in `.datalad/config` after the fact.

```
# Create a source dataset for all analysis components
source_ds="pipeline"
datalad create $source_ds
cd $source_ds

# clone the container-dataset as a subdataset.
containername='tvbpipe-fmriprep'
containerstore="/scratch/snx3000/bp000225/tvbpipe-fmriprep"
datalad clone -d . ${containerstore} code/pipeline

# Register the container in the top-level dataset.
# If necessary, configure your own container call in the
# --call-fmt argument. If your container does not need a
# custom call format, remove the --call-fmt flag and its
# options below.
datalad containers-add \
  --call-fmt 'singularity run -B {{pwd}} --cleanenv {img} {cmd}' \
  -i /scratch/snx3000/bp000225/${source_ds}/code/pipeline/.datalad/environments/${containername}
  $containername

# amend the previous commit with a nicer commit message
git commit --amend -m 'Register pipeline dataset'
```

Analysis
superdataset to
specify all
dependencies of
an analysis

Further reading
<https://arxiv.org/abs/2102.05888>

[Submitted on 11 Feb 2021 (v1), last revised 29 Mar 2021 (this version, v2)]

Brain Modelling as a Service: The Virtual Brain on EBRAINS

Michael Schirner, Lia Domide, Dionysios Perdikis, Paul Triebkorn, Leon Stefanovski, Roopa Pai, Paula Popa, Bogdan Valean, Jessica Palmer, Chloë Langford, André Blickensdörfer, Michiel van der Vlag, Sandra Diaz-Pier, Alexander Peyser, Wouter Klijn, Dirk Pleiter, Anne Nahm, Oliver Schmid, Marmaduke Woodman, Lyuba Zehl, Jan Fousek, Spase Petkoski, Lionel Kusch, Meysam Hashemi, Daniele Marinazzo, Jean-François Mangin, Agnes Flöel, Simisola Akintoye, Bernd Carsten Stahl, Michael Cepic, Emily Johnson, Gustavo Deco, Anthony R. McIntosh, Claus C. Hilgetag, Marc Morgan, Bernd Schuller, Alex Upton, Colin McMurtrie, Timo Dickscheid, Jan G. Bjaalie, Katrin Amunts, Jochen Mersmann, Viktor Jirsa, Petra Ritter

The Virtual Brain (TVB) is now available as open-source cloud ecosystem on EBRAINS, a shared digital research platform for brain science. It offers services for constructing, simulating and analysing brain network models (BNMs) including the TVB network simulator; magnetic resonance imaging (MRI) processing pipelines to extract structural and functional connectomes; multiscale co-simulation of spiking and large-scale networks; a domain specific language for automatic high-performance code generation from user-specified models; simulation-ready BNMs of patients and healthy volunteers; Bayesian inference of epilepsy spread; data and code for mouse brain simulation; and extensive educational material. TVB cloud services facilitate reproducible online collaboration and discovery of data assets, models, and software embedded in scalable and secure workflows, a precondition for research on large cohort data sets, better generalizability and clinical translation.

Thank you for
listening.



Import custom code

Here we can copy custom code into the dataset. For example, FreeSurfer (part of fmriprep, MRtrix3_connectome) needs a license file (free but must be individually obtained), which is now copied from the home folder into the data set.

```
# committing changes after copying license file  
cp ~/license.txt code/license.txt  
datalad save -m "Add Freesurfer license file"
```

Quickly test-run the container on the login node

Before we create SLURM batch scripts we will give the container a test run to make sure we set up everything correctly so far.

```
# load Singularity module again in case there was an
# environment change
module load singularity

# the subject we test
subid="sub-CON03"

# create workdir for fmriprep inside the dataset to
# simplify singularity call PWD will be available in the
# container
mkdir -p .git/tmp/wdir

# Replace the command with an fmriprep parametrization
# that fits your analysis
datalad containers-run \
  -m "Compute ${subid}" \
  -n tvbpipe-fmriprep \
  --explicit \
  -o fmriprep/${subid} \
  -i inputs/data/${subid}/ses-postop/ \
  -i code/license.txt \
  "inputs/data/${subid} . participant --participant-label $subid \
    --anat-only -w .git/tmp/wdir --fs-no-reconall --skip-bids-validation \
    --fs-license-file code/license.txt"
```

Setting up a reproducible, version-controlled, provenance-tracked workflow

Install DataLad

With Miniconda DataLad can be installed with just three commands

```
ssh daint # or the supercomputer you are working on
module load daint-mc # supercomputer-specific module
module load cray-python/3.8.5.0 # load Python environment

# The easiest way to install DataLad with all
# dependencies on a supercomputer without root
# permissions is by using conda
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
# acknowledge license, initialize Miniconda3, close and
# re-open shell
conda install -c conda-forge datalad
```

Install DataLad extensions

Two DataLad extensions are required:

`datalad-container` and

container
support

`datalad-neuroimaging`

Metadata
extraction

Install them with pip like this:

```
pip install datalad-neuroimaging datalad-container
```

Configure DataLad

Setup your git identity. Specify your git username and email, which will be used to track changes in your DataLad data set: changes you make are associated with your name and email address.

```
cd ~  
git config --global --add user.name "MichaelSchirner"  
git config --global --add user.email m.schirner@fu-berlin.de
```

Limitations

- not every feature of DataLad may work „out of the box“
- sometimes customization is required

↗ **Find-out-more:** Fine-tuning: Enable re-running

If you want to make sure that your dataset is set up in a way that you have the ability to rerun a computation quickly, the following fMRIprep-specific consideration is important: If fMRIprep finds preexisting results, it will fail to run. Therefore, all outputs of a job need to be removed before the job is started[3]. We can simply add an attempt to do this in the script (it wouldn't do any harm if there is nothing to be removed):

```
(cd fmriprep && rm -rf logs "$subid" "$subid.html" dataset_descriptio
(cd freesurfer && rm -rf fsaverage "$subid")
```

```
# pybids (inside fmriprep) gets angry when it sees dangling symlinks
# of .json files -- wipe them out, spare only those that belong to
# the participant we want to process in this job
find sourcedata -mindepth 2 -name '*.json' -a ! -wholename "$1" '*'
```

meanwhile fixed in
PyBIDS thanks to
community
interaction
<https://github.com/bids-standard/pybids/issues/631>

