46770 Integrated energy grids

Marta Victoria

Lecture 13 – Part II. Open and collaborative science

Outline

1. What does “Open science” mean?

2. What are the benefits and disadvantages of making scientific results openly available for everyone?

3. What are the benefits and disadvantages of making your documents, code, results openly available for everyone? How can you do it?

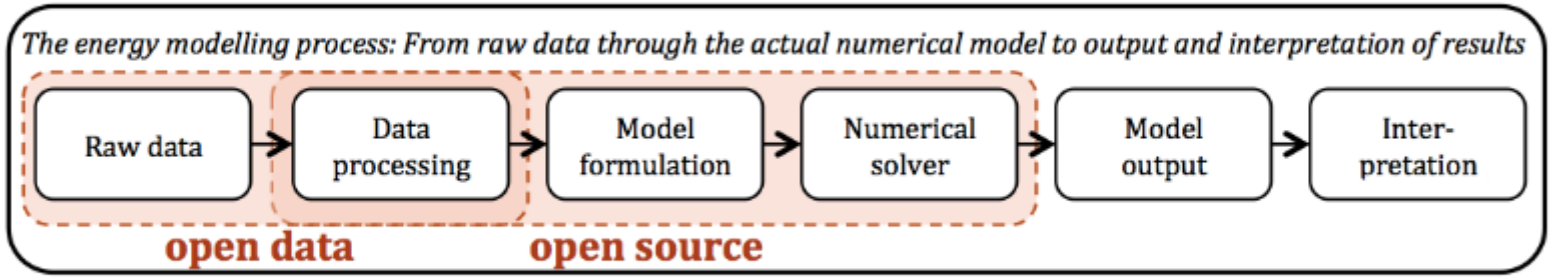
4. Collaborative scientific code writing

What does “Open Science” mean?

Open Science is the practice of science in such a way that others can collaborate and contribute, where research data, lab notes and other research processes are freely available, under terms that enable reuse, redistribution and reproduction of the research and its underlying data and methods.

What does this mean for open energy modelling?

The whole chain from raw data to modelling results should be open:



Open data + free software ⇒ Transparency + Reproducibility

Pros & Cons of “Open Science”

Pros and cons list that we wrote together.

Sharing documents

It is NOT enough to make the document available. Please also add a license!

Preferably an open license. How can you do it?

If you publish a paper, consider open journals and make the preprint available in ArXiv

https://choosealicense.com/

Choose an open source license

An open source license protects contributors and users. Businesses and savvy developers won't touch a project without this protection.

Which of the following best describes your situation?

I need to work in a community.

Use the **license preferred by the community** you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to **add a license**.

I want it simple and permissive.

The **MIT License** is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

Babel, **.NET Core**, and **Rails** use the MIT License.

I care about sharing improvements.

The **GNU GPLv3** also lets people do almost anything they want with your project, *except* distributing closed source versions.

Ansible, **Bash**, and **GIMP** use the GNU GPLv3.

An example of an open license: Creative Commons Attribution 4.0 International.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

How to add a license to your Master thesis and code

Why?

- A license clarifies the conditions under which your code, text, data, or figures can be reused.
- In the absence of a license, the author still retains proprietary copyright, and the conditions under which the materials can be used are unclear.
- You want other people to use your work, the best way is to create the conditions so that they can do it.

How? You can select a license using https://choosealicense.com/

For a master thesis, the easiest is to use Creative Commons Attribution 4.0 International. To do that you need to add the following sentence to one of the first pages of your document

Copyright © 2025 *add your name* < *add your email* >

This work is licensed under a Creative Commons Attribution 4.0 International Licence (CC-BY).

Attribute the authors when you use their images

Where to get open-license images and icons?

https://thenounproject.com/

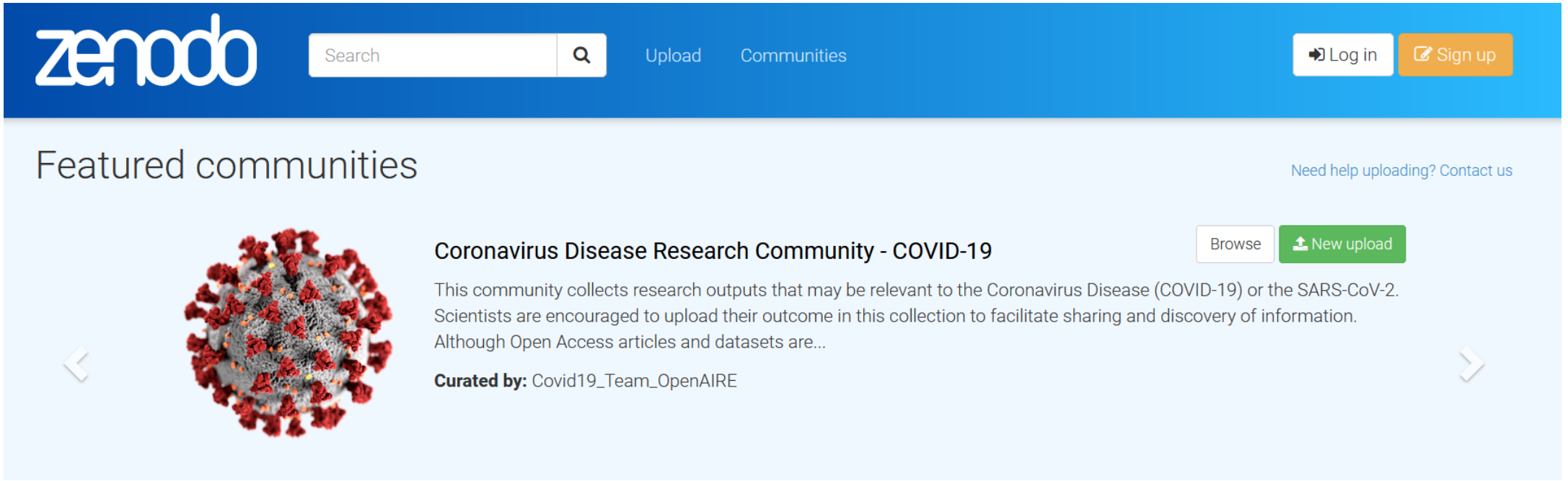
https://pixabay.com/es/

https://commons.wikimedia.org/wiki/Main_Page

Sharing data and code

It is not enough to share the data in an email. Please also upload it to a permanent repository and add a license. You can use Zenodo as a repository to share data. Zenodo adds a permanent DOI (Digital Object Identifier) to your data.

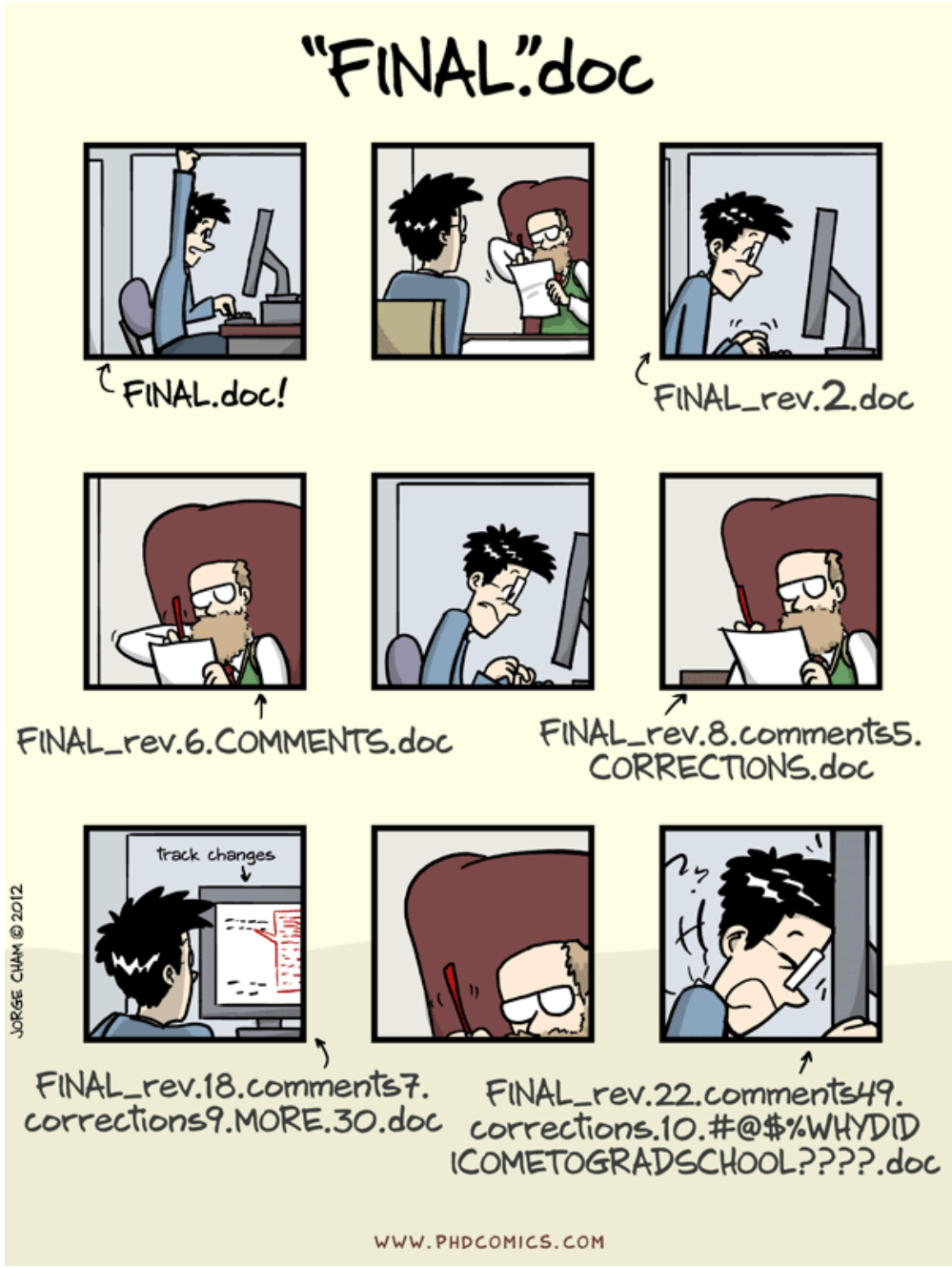
Additionally, use version control



Sharing data and code

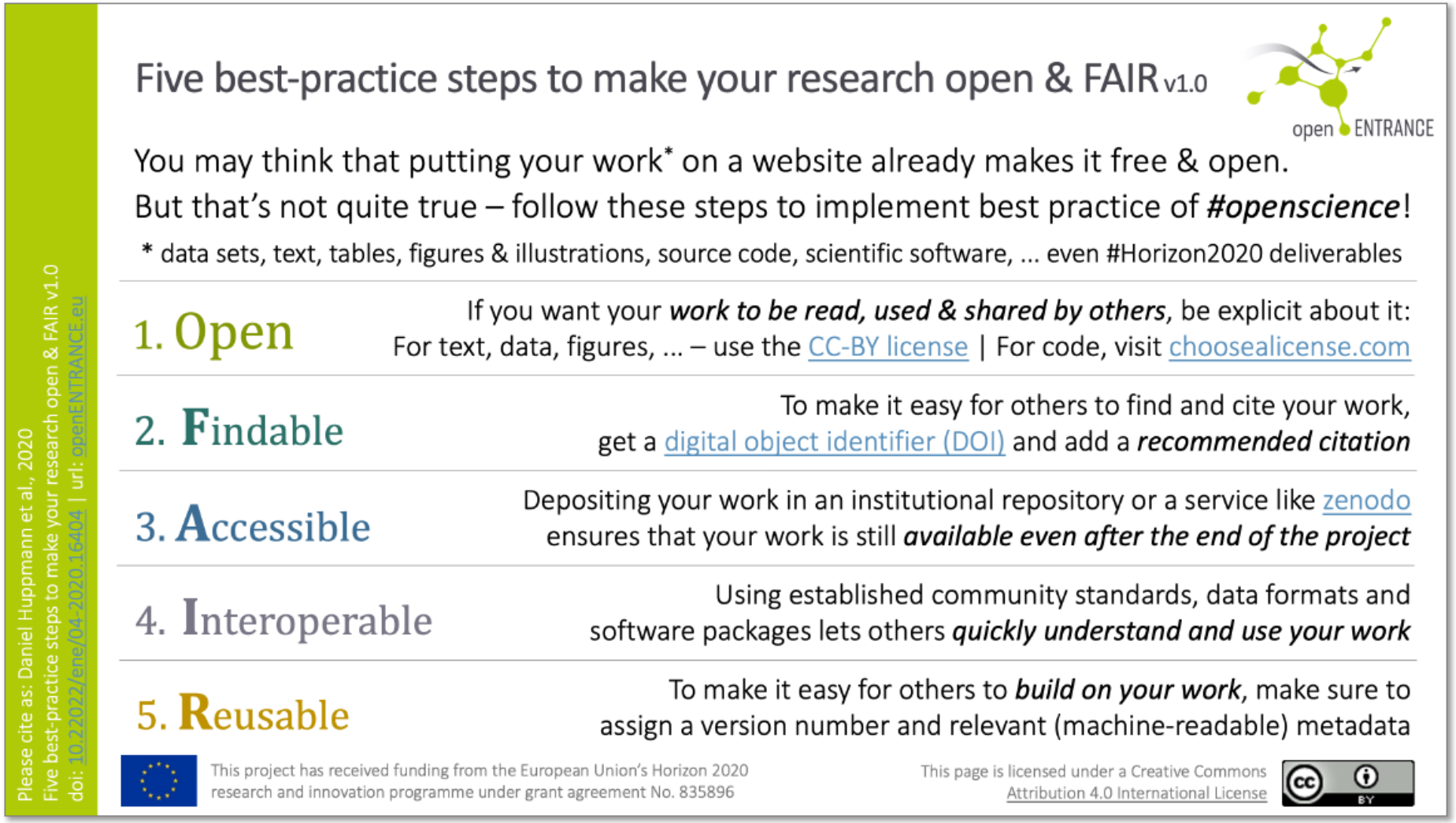
Additionally, use version control

(more on this on the next lecture)



Collaborative scientific code writting

FAIR principles



Ref: Wilkinson et al., The FAIR Guiding Principles *Scientific Data* 3:160018 (2016)

PyPSA and PyPSA-Eur are open software

- They are built on using other open codes: Python, pandas, numpy, xarray, matplotlib, cartopy …
- They use an open license

master ▾ PyPSA / LICENSE.txt Go to file ⋯

PyPSA/PyPSA is licensed under the
GNU General Public License v3.0

Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

Permissions	Limitations	Conditions
✓ Commercial use	✕ Liability	ⓘ License and copyright notice
✓ Modification	✕ Warranty	ⓘ State changes
✓ Distribution		ⓘ Disclose source
✓ Patent use		ⓘ Same license
✓ Private use		

This is not legal advice. Learn more about repository licenses.

PyPSA/PyPSA is licensed under the
MIT License

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

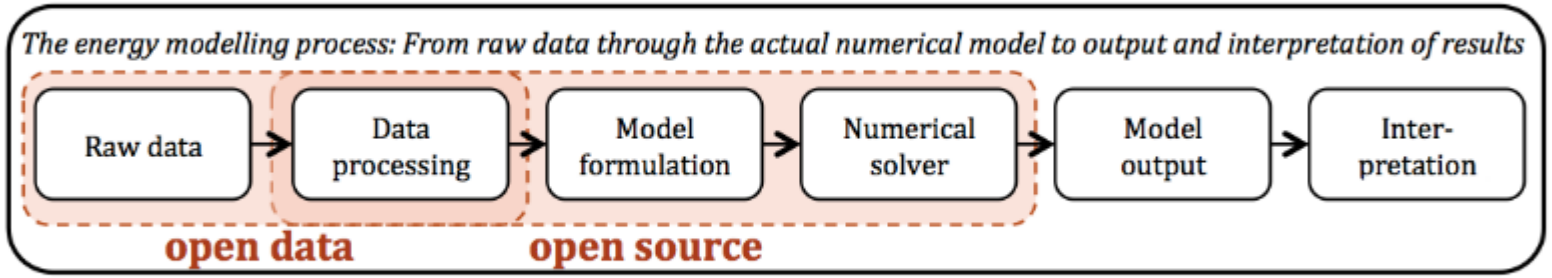
Permissions	Limitations	Conditions
✓ Commercial use	✕ Liability	ⓘ License and copyright notice
✓ Modification	✕ Warranty	
✓ Distribution		
✓ Private use		

This is not legal advice. Learn more about repository licenses.

Open Energy Modelling Platform

open energy
modelling initiative
openmod

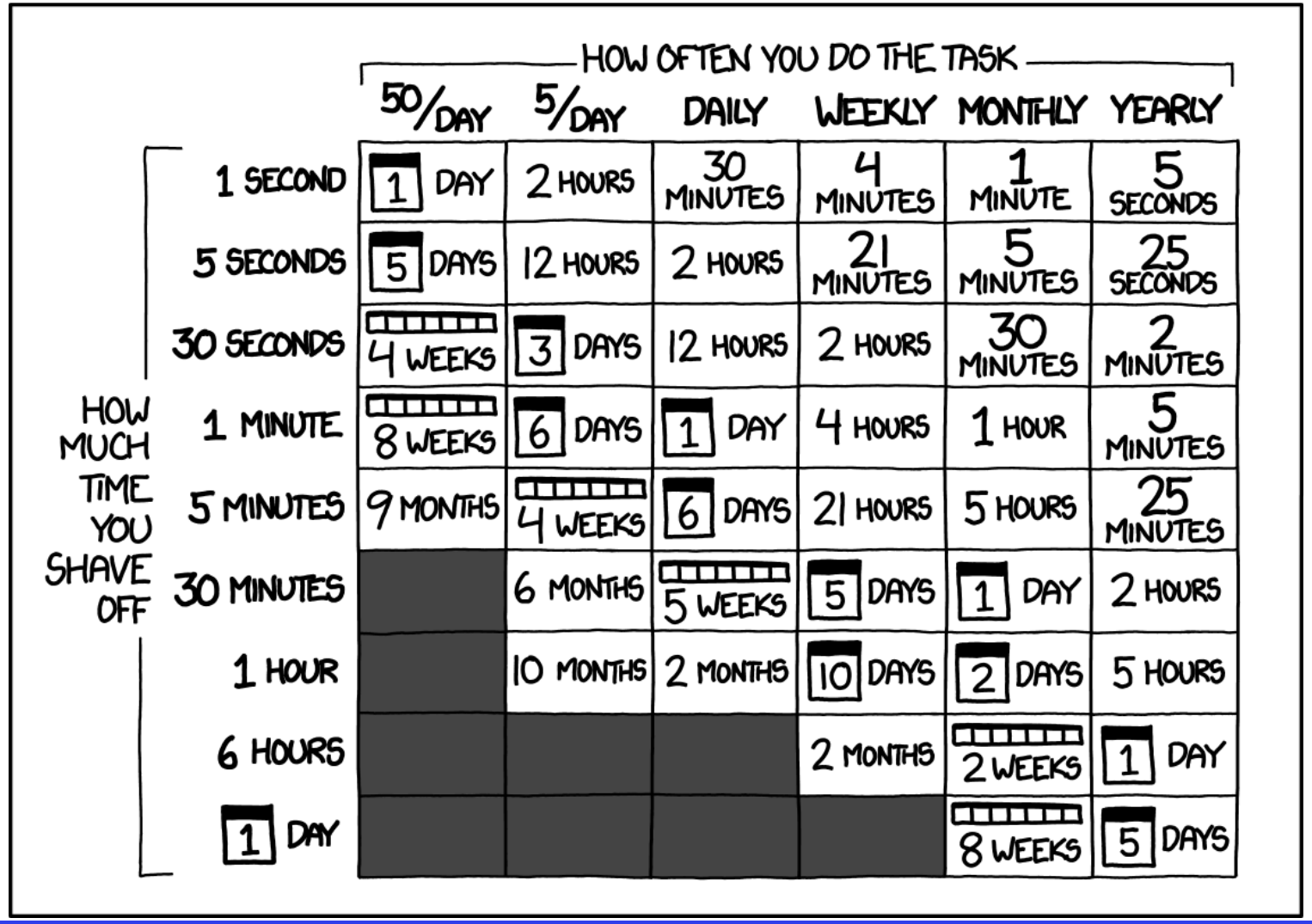
The whole chain from raw data to modelling results should be open:



Open data + free software ⇒ Transparency + Reproducibility

The Open Energy Modelling Initiative gathers Master, PhD Students and other researchers dealing with energy modelling worldwide. It is a wonderful platform to learn what others are doing and what has been done in the past. To search on a particular topic, you can: (a) register in the forum and search by topic, (b) subscribe to the distribution list where you can also search on previous topics or (c) look in the wiki.

Is it worth the time?



Ref: xkcd.com

Writing code for other to read it

Making your code open has many benefits (your credibility increases, your work is reviewed, and you may get useful feedback, you can use your Github account as a cv, it allows you to collaborate with other researchers)

Try to write clean code so that others can understand it (this will also help you understand your own code in 6 months time)

Do not worry if the code is not perfect, share it and you will improve it with time.

Recommended further reading:

PEP-8 a standard to write easy-to-read python code. https://www.python.org/dev/peps/pep-0008/

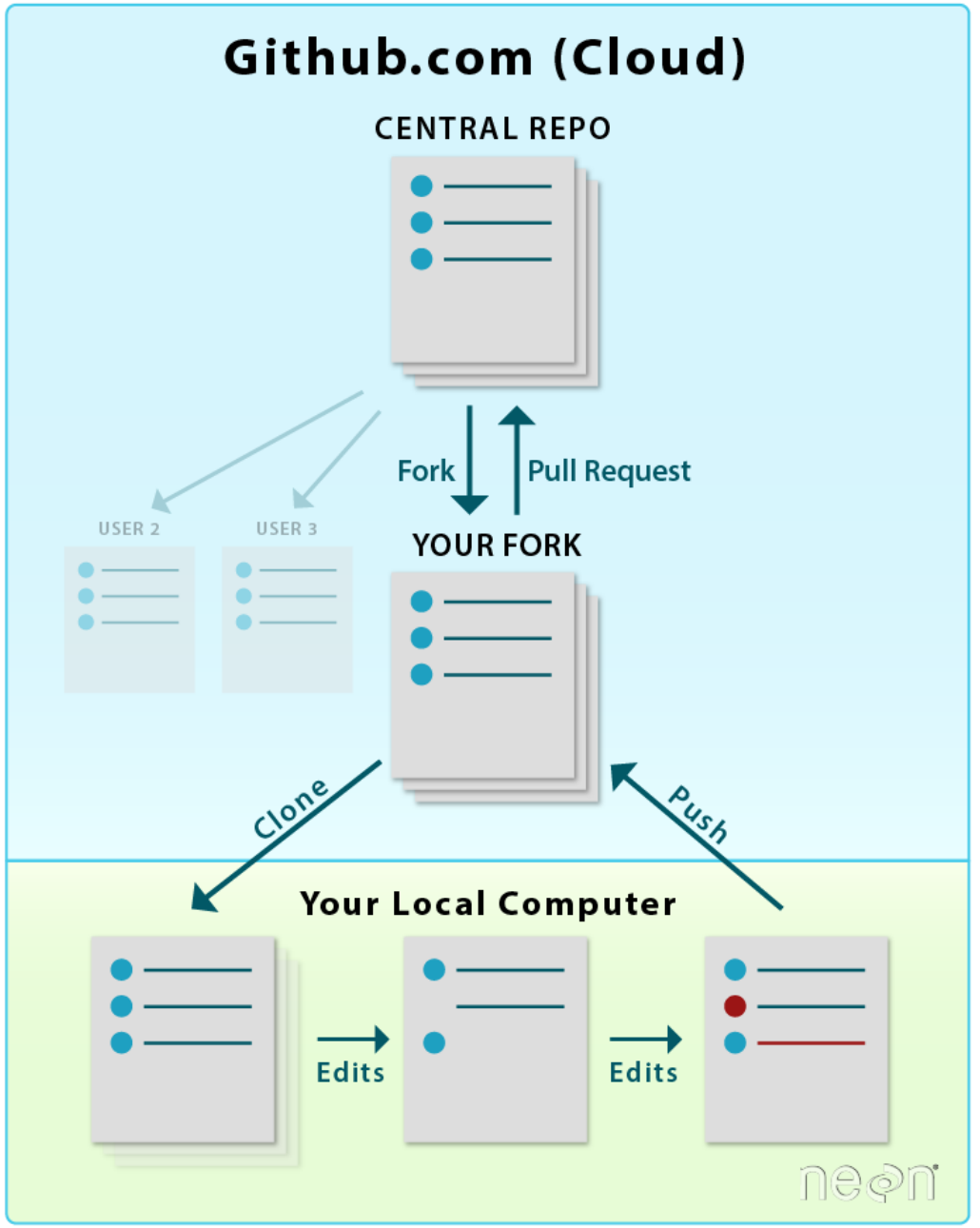
Barnes (2010). Publish your computer code: it is good enough. *Nature* 467(753):775. 10.1038/467753a

Barba (2016). The hard road to reproducibility. *Science* 354(6308):142. 10.1126/science.354.6308.142

Sharing code

Github is a version control software that facilitates collaboration among researchers

Similar software exists (e.g. Gitlab)



Steps to use GitHub

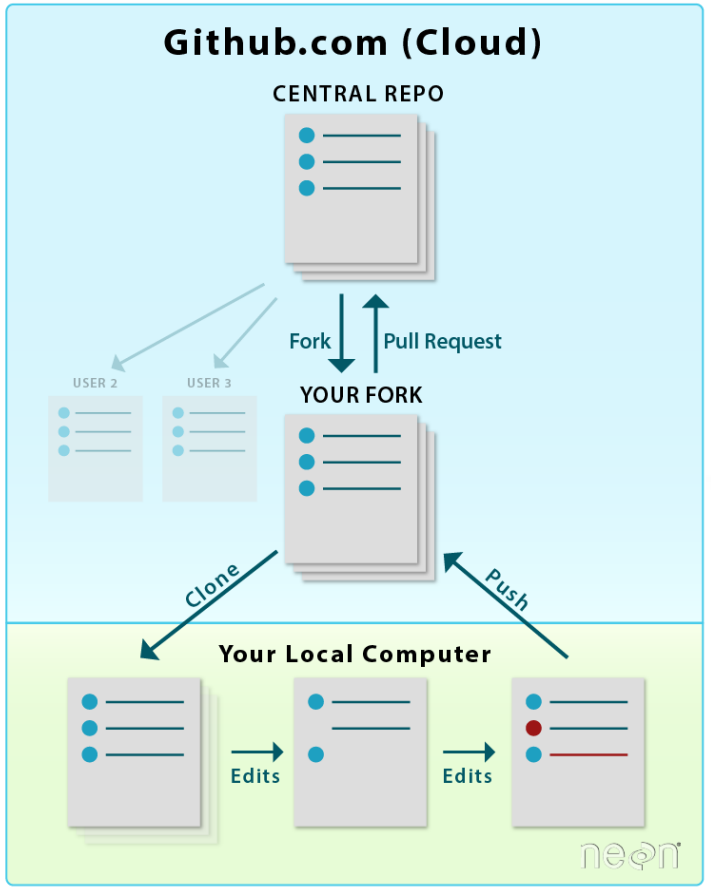
1. You make a copy (fork) of an existing repository to your Github account.

2. You clone the repository from your Github account to your local computer (you can use Github Desktop or the terminal for this).

3. You make all the changes that you want. Every time you made a substantial change, you commit adding a descriptive message (write what will happen if the commit is accepted).

4. You push all the commits to the forked repository in your Github account.

5. When you want to merge that version (branch) with the original central repository you make a pull request.



Practical exercise A: Creating a repository

1. Create your own repository
2. Add a README with a description
3. Add a license.
4. Install and open Github Desktop and clone the repository to your local computer
5. Add a file with your name on the file name, commit and push to your online repository
6. Change the README directly on the online repository and pull the changes to your local computer
7. Add a badge to the README including your license (you can use these templates)

Practical exercise B: contribute to an existing repository

1. Fork the repository IEG-tutorial-2025 to your Github account
2. Clone the repository IEG-tutorial-2025 to your local computer
3. Add a file with your name on the file name, commit and push to your forked repository
4. Create a “pull-request” (i.e., you are asking the owner of the main repository to accept your changes)
5. Update your forked repository with the latest version on the original repository and then pull the changes to your local computer.

Practical exercise B: contribute to an existing repository

Instructions if you want to use the terminal

1. Fork the repository IEG-tutorial-2025 to your Github account
2. Clone the repository IEG-tutorial-2025 to your local computer

```
git clone https://github.com/your_name/MESM-tutorial-fall2023.git
```

2. Add a file with your name on the file name, commit and push to your forked repository

```
git add .
```

```
git commit -m "add a file with my name"
```

```
git push
```

3. Create a “pull-request” (i.e., you are adding the owner of the main repository to accept your changes)

4. Update your forked repository with the latest version on the original repository and then pull the changes to your local computer.

Pull request

How does a pull request look like in reality?

https://github.com/PyPSA/pypsa-eur-sec/pull/63


DTU