Tools for Reproducible Computational Science

Valentin Volkl, 09.09.2016
CSC student presentation

The workflow leading to a paper

-Study literature
-Think, discuss, work...
-Get new results
-Write and submit manuscript

And you're done, right?

-Share the code that you used to produce the result!

Reproducibility is a cornerstone of science.

Sharing code enables more people to reproduce your results.

Many major journals encourage you to do so.

doi:10.1038/nature10836

Excuses

source : doi:10.1038/467753a

"It is not common practice."

"People will pick holes and demand support and bug fixes."

"The code is valuable intellectual property that belongs to my institution."

"It is too much work to polish the code."

Organize your Code from the beginning

There are many tools to help you

e.g. cookiecutter-data-science

http://drivendata.github.io/cookiecutter-data-science/

Sample repository structure for a data science project

Directory structure

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
```

Ensure that your code is citable!

one option: get a **DOI**

Ensure that your code is citable!

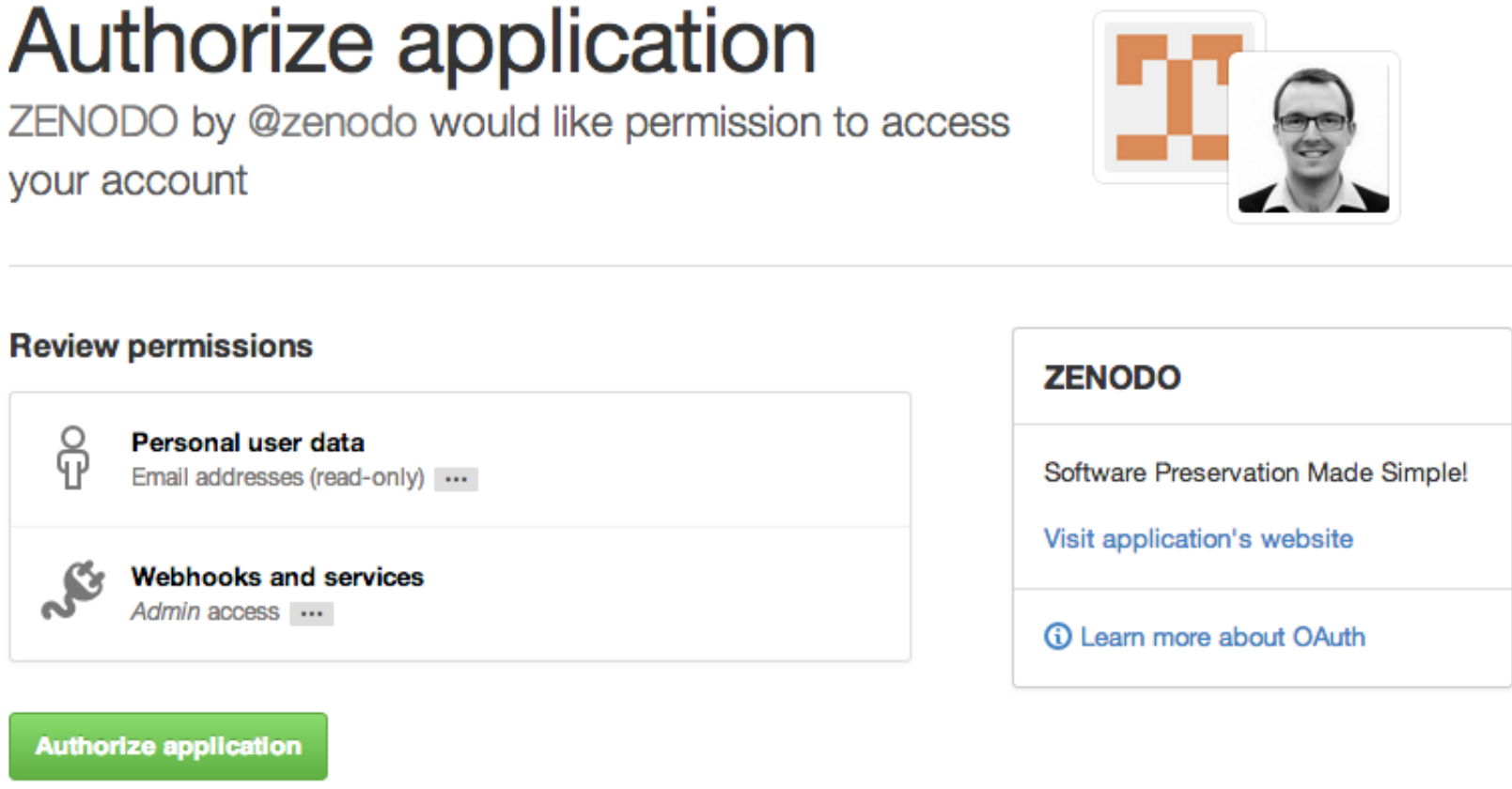
one option: get a **DOI**

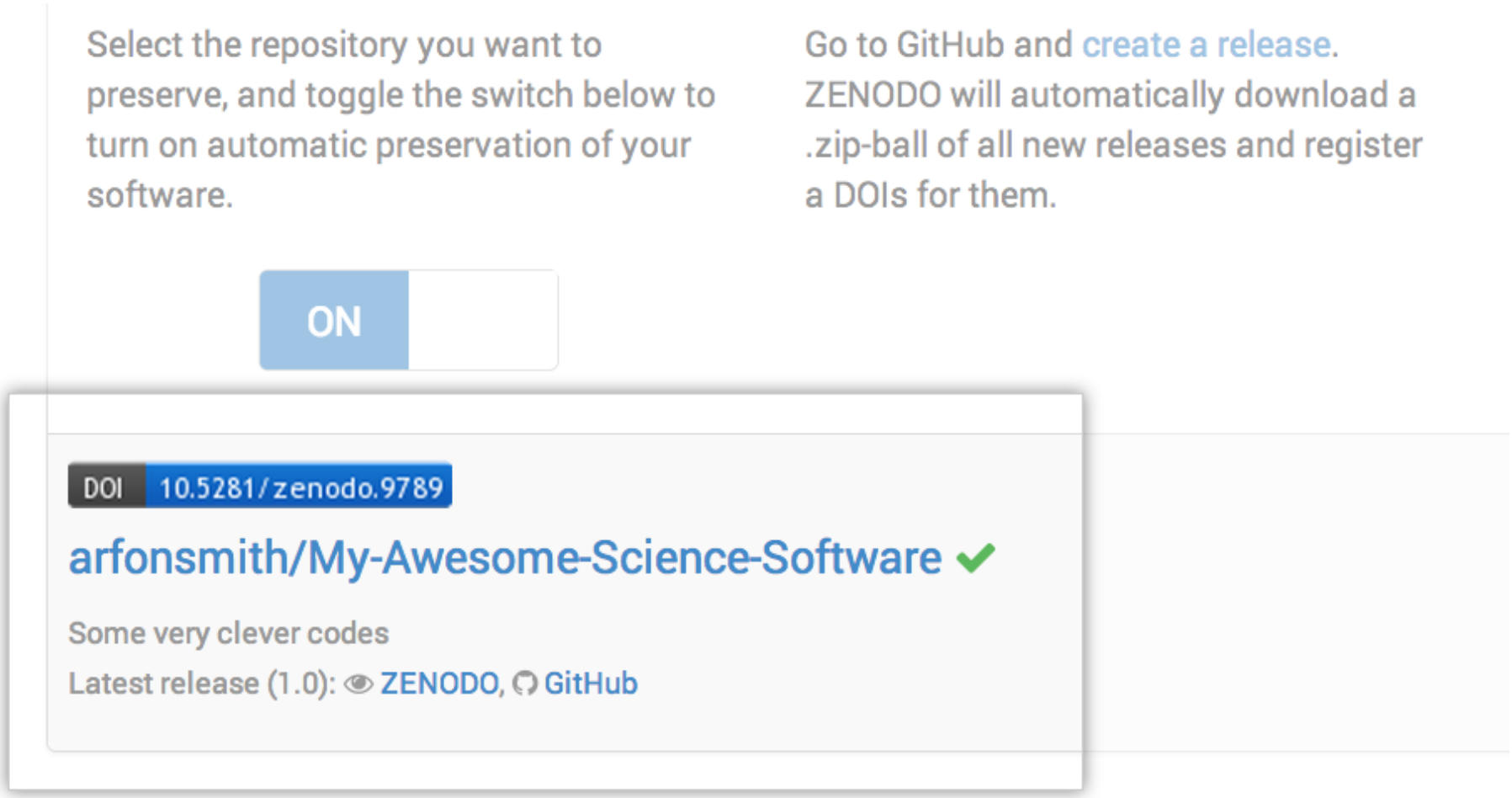
What's a doi?
"Digital Object Identifier"

- -Uniquely identifies a digital object.
- -Resolves to the object no matter where the object is actually located

How?

zenodo.org, for example, needs nothing more than a github account.




Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software.
Go to GitHub and create a release. ZENODO will automatically download a .zip-ball of all new releases and register a DOIs for them.
ON
DOI 10.5281/zenodo.9789
arfonsmith/My-Awesome-Science-Software
Some very clever codes
Latest release (1.0): ZENODO, GitHub