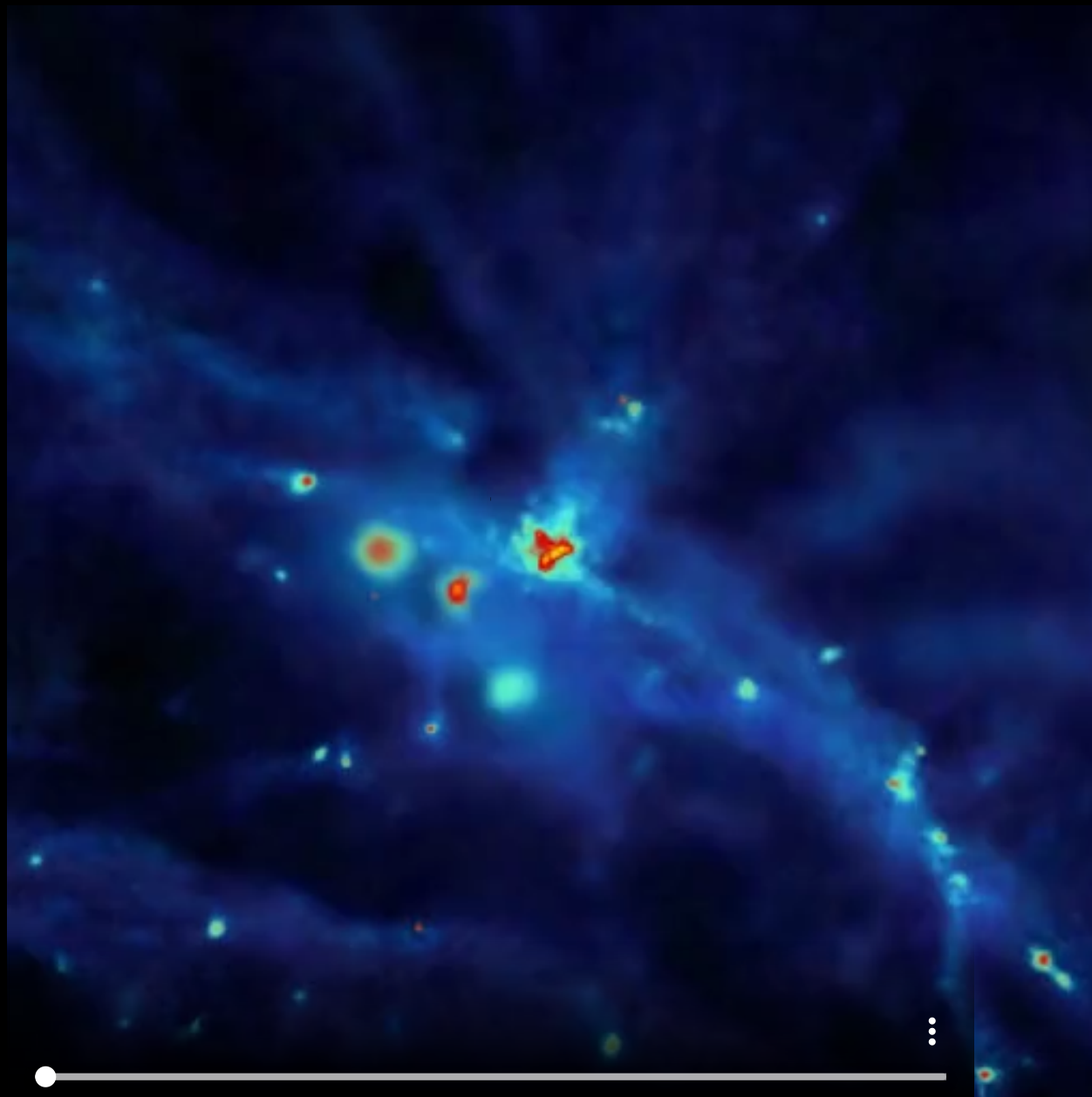# Create a data management plan
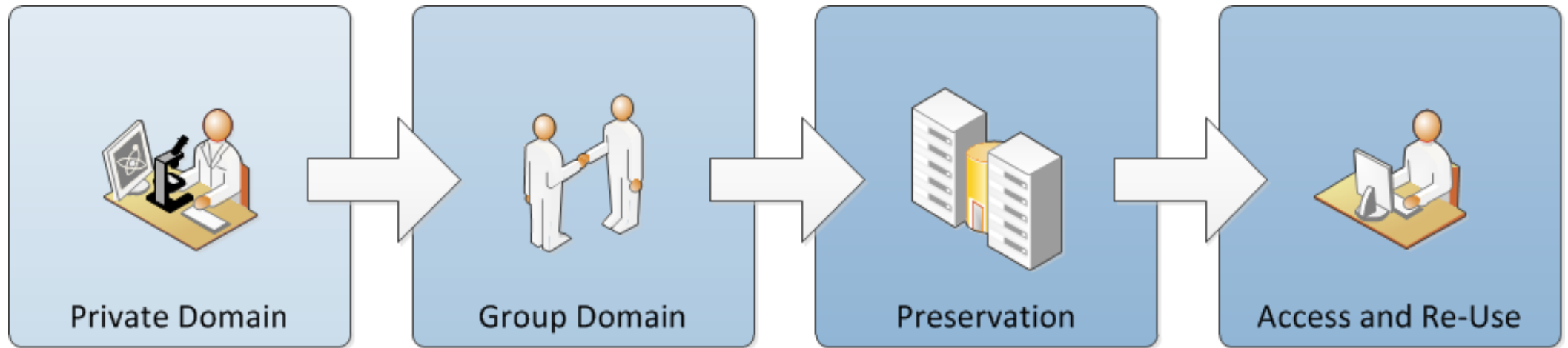
(with RDMO)

Jochen Klar

Leibniz Institute for Astrophysics Potsdam (AIP)

Astronomy!

Domain model for research data

Effort for **data management** ⟶

Complexity of **metadata** ⟶

Treloar, A., D. Groenewegen, and C. Harboe-Ree (2007), The Data Curation Continuum - Managing Data Objects in Institutional Repositories, http://dx.doi.org/10.1045/september2007-treloar

DFG-Projekt RADIESCHEN (2013): Rahmenbedingungen einer disziplinübergreifenden Forschungsdateninfrastruktur. Organisation und Struktur. http://dx.doi.org/10.2312/RADIESCHEN_005

Data management plan

A highly idealized view ...

- Optimization of data management from the earliest stages of new project, during its whole lifetime, and beyond.

- Guidelines for the different stakeholders on how to handle the data acquired/used in the project.

- Planning of data publications and long-term preservation.

- Increase of quality and efficiency of scientific work.

- Beneficial to the researcher, the institution, and the funder.

DCC. (2013). Checklist for a Data Management Plan. v.4.0. Edinburgh: Digital Curation, http://www.dcc.ac.uk/resources/data-management-plans/checklist

J. Ludwig / H. Enke (Hrsg.): Leitfaden zum Forschungsdaten-Management (2013), http://www.forschungsdaten.org/index.php/Datei:Leitfaden_Data-Management-WissGrid.pdf

Tools

DMPonline (by the Digital Curation Centre, DCC)

- https://dmponline.dcc.ac.uk
- Initially tailored to the situation in the UK, but also H2020

DMPTool (by the California Digital Library, CDL)

- https://dmptool.org
- Targeted towards the situation in the US

RDMO (by AIP, FHP, and KIT, funded by the DFG)

- https://rdmorganiser.github.io
- Organiser instead of plan, local instead of central

Data management plan

A more realistic view ...

- Data management plans are seen as
  - pure obligation in the application process
  - static document, to be filed and forgotten
- Existing tools focus on providing templates
  - producing text documents
  - no *actionable* DMP
- Existing tools are central web sites
  - limited customization (local institution, research field)
 - transfer of possible sensitive information

RDMO - organiser instead of plan

- Support of the RDM
  over the whole lifetime of the project

- Engage all stakeholders

- Collect all necessary information for
  a sustainable RDM

  - Guided questionnaire

  - Controlled vocabularies

  - Checkboxes, dropdowns, sliders

- Export in different formats using
  predefined views

- Identify follow up tasks

- Interfaces to other software, APIs

RDMO - local instead of central

- Independent operation by
  - Universities
  - Research institutions
  - Libraries
  - Collaborations
- Customizing to the local context
- Customizing to research field specific aspects
- Customizing for own *Corporate Design*
- Frictionless installation, e.g. by the IT department
- 8 instances in production, ~ 20 testing
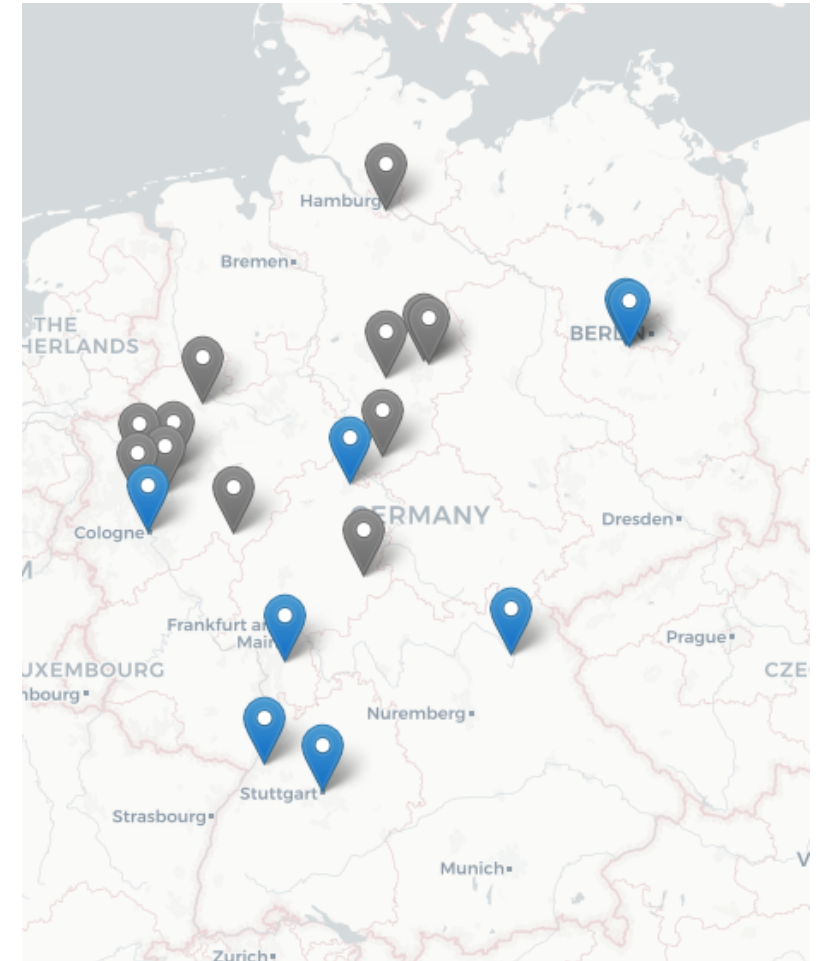
Open source

- Open source development since *day one*

- Code on GitHub at https://github.com/rdmorganiser/rdmo

- Documentation on Read the Docs at http://rdmo.readthedocs.io

- Apache 2.0 License

- Tests using Travis CI and Coveralls

- Releases with GitHub and PyPI
  ```
  pip install rdmo
  ```

- DOI with Zenodo (planned)

- optional Docker deployment
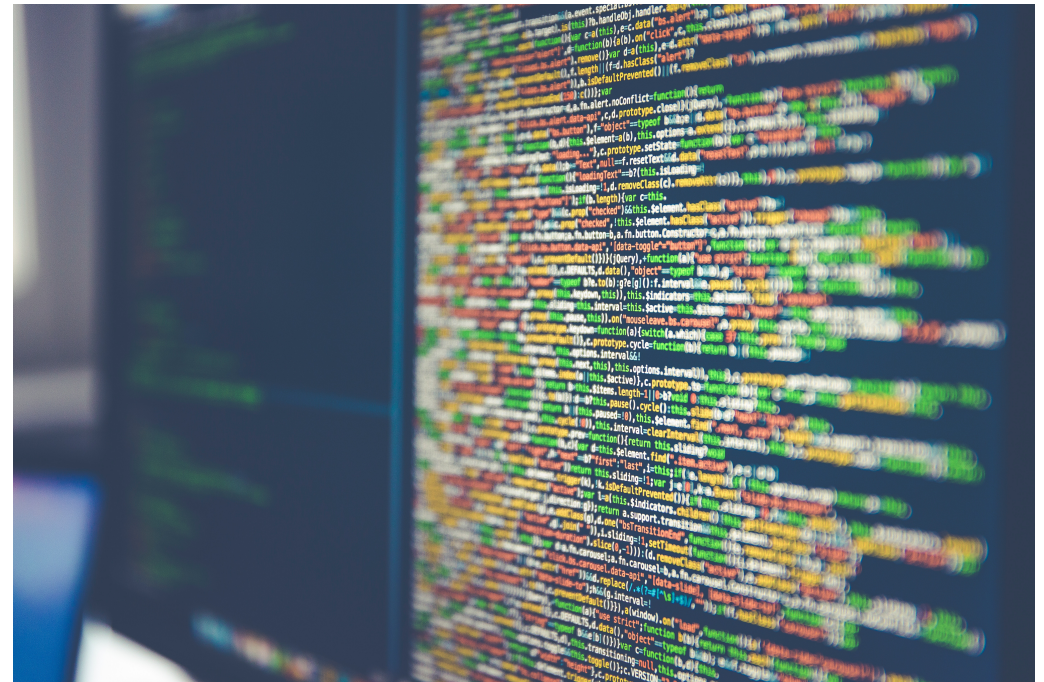
RDMO XML Export/Import format

```xml
<rdmo xmlns:dc="http://purl.org/dc/elements/1.1/">
    <catalog dc:uri="https://rdmorganiser.github.io/terms/questions/rdmo">
        <dc:comment/>
        <order>1</order>
        <title lang="en">RDMO</title>
        <title lang="de">RDMO</title>
    </catalog>
    <section dc:uri="https://rdmorganiser.github.io/terms/questions/rdmo/general">
        <dc:comment/>
        <order>0</order>
        <title lang="en">General</title>
        <title lang="de">Allgemein</title>
    </section>
    ...
</rdmo>
```

Programmable JSON API

```
curl -X GET -H 'Authorization: Token oojoh3phaighaebiNeiyeeCeiY3Peuv2eitoojoh' \
  https://rdmo.aip.de/api/v1/projects/values/?attribute__path=project/dataset/size/volume
```

```json
[
  {
    "id":10061,
    "project":"https://rdmo.aip.de/api/v1/projects/projects/69/",
    "attribute":"https://rdmo.aip.de/api/v1/domain/attributes/262/",
    "set_index":0,
    "collection_index":0,
    "text":"",
    "option":null,
    "created":"2017-05-29T14:50:20.009917Z",
    "updated":"2017-05-29T14:50:20.009924Z"
  },
  ...
]
```

RDMO Resources

Website: rdmorganiser.github.io

GitHub organisation: github.com/rdmorganiser
RDMO source code: github.com/rdmorganiser/rdmo
RDMO questionaire: github.com/rdmorganiser/rdmo-catalog

Documentation: rdmo.readthedocs.io

Demo instance: rdmo.aip.de

Mailinglist: rdmo@listserv.dfn.de
Twitter: @rdmorganiser
Slack: rdmo.slack.com
GitHub issues: github.com/rdmorganiser/rdmo/issues

Thank you for your attention

jklar@aip.de, @jochenklar

rdmorganiser.github.io

rdmo.aip.de

Appendix

Who owns the data?

- We might not have copyright to the data!
  - In most jurisdictions, data, datasets and databases are not subject to copyright as books, articles or images are. Copyright only applies on creative work which exceeds a threshold of originality, which is usually denied for measurements of observable fact in the world.
  - The investment in creating databases is protected (in the EU) under the Sui generis database right, which gives the right to object to the copying of substantial parts of a database. It is still unclear, what substantial means, especially in the context of copyright licenses.
- If we assume to have copyright: Use a well known license!
  - Make everybody's live easier and use CC0 and wave all your rights.
  - Individual licenses lead to incompatibilities to widely used licenses.

Desmet, P. 2012. Why we should publish our data under Creative Commons Zero (CC0). Canadensys.
http://community.canadensys.net/2012/why-we-should-publish-our-data-under-cc0

Which license should I use?

- **CC0** waives all rights to the work.

- **Creative Commons Attribution (CC BY)** forces people to include an attribution (when sharing), but might lead to incompatibilities and/or attribution stacking if used for derivative work (other data, software).

- **Creative Commons Attribution-NoDerivs (CC BY ND)** works like CC BY, but people must not create derivative works.

- **Creative Commons Attribution-NonCommercial (CC BY NC)** works like CC BY, but must not be used for commercial purposes (the license is considered harmful).

- **Creative Commons Attribution-ShareAlike (CC BY SA)** works like CC BY, but derivative works need to use the same license (could lead to conflicts).

- **Software licenses (BDS, MIT, Apache2, GPL, AGPL)** are considered not suitable for cultural works.

- The **Open Database License (ODbL)** is like a CC BY SA for databases.

What about DOIs?

- DOIs make data citable!

- DOIs map a chosen ID to an URL, e.g.:

  https://doi.org/10.17876/musewide/dr.1 ⟶ https://musewide.aip.de/metadata/musewide_dr1

- DataCite also collects metadata:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<resource xmlns="http://datacite.org/schema/kernel-4" xmlns:xsi="...
  <identifier identifierType="DOI">10.17876/musewide/dr.1</identifier>
  <titles>
    <title xml:lang="en-us">musewide_dr1</title>
  </titles>
  <publisher>Leibniz Institut für Astrophysik Potsdam (AIP)</publisher>
  <publicationYear>2018</publicationYear>
  <subjects>...</subjects>
  <creators>...</creators>
  ...
</resource>
```

What is version control?

- Initially from software development, but can be use on any text file!

- A version control tracks changes to a *repository*

- It provides tools to merge changes in a collaborative setup

What is version control?

- Initially from software development, but can be use on any text file!

- A version control tracks changes to a *repository*

- It provides tools to merge changes in a collaborative setup

- `git` is a distributed version control system and the de-facto standard

```
git clone <repository_url>
git init
git commit -m "Adding a lot of stuff"
git push
git pull
git status
git log
```

What is version control?

- Initially from software development, but can be use on any text file!

- A version control tracks changes to a *repository*

- It provides tools to merge changes in a collaborative setup

- `git` is a distributed version control system and the de-facto standard

```
git clone <repository_url>
git init
git commit -m "Adding a lot of stuff"
git push
git pull
git status
git log
```

- People use github.com to store, share and collaborate on code, scripts, documents, ...

Beginner tutorial at Atlassian, https://www.atlassian.com/git/tutorials/what-is-version-control