LEIBNIZ INFORMATION CENTRE
FOR SCIENCE AND TECHNOLOGY
UNIVERSITY LIBRARY

TIB

Moving towards FAIRness in Research Data and Software Management

Dr. Angelina Kraft
ORCiD: https://orcid.org/0000-0002-6454-335X

Thüringer FDM-Tage 2020: *FAIR Research Software and Beyond: How to make the most of your code*
02 July 2020

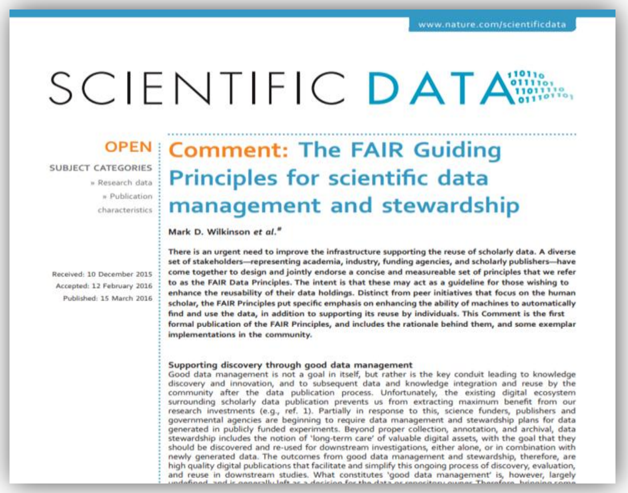
Leibniz
Association

Agenda

- FAIR Principles: Data vs. Software – general concepts
- Measures for increasing FAIRness
    - Data/Software Management Plans
    - PIDs
    - Software citation
    - Software licences
    - Version control & Project management
- Summary

FAIR Data (and Software) Principles I

In 2016:



www.nature.com/scientificdata

SCIENTIFIC DATA

OPEN

SUBJECT CATEGORIES
» Research data
» Publication characteristics

Comment: The FAIR Guiding Principles for scientific data management and stewardship

Mark D. Wilkinson *et al.*#

Received: 10 December 2015
Accepted: 12 February 2016
Published: 15 March 2016

There is an urgent need to improve the infrastructure supporting the reuse of scholarly data. A diverse set of stakeholders—representing academia, industry, funding agencies, and scholarly publishers—have come together to design and jointly endorse a concise and measureable set of principles that we refer to as the FAIR Data Principles. The intent is that these may act as a guideline for those wishing to enhance the reusability of their data holdings. Distinct from peer initiatives that focus on the human scholar, the FAIR Principles put specific emphasis on enhancing the ability of machines to automatically find and use the data, in addition to supporting its reuse by individuals. This Comment is the first formal publication of the FAIR Principles, and includes the rationale behind them, and some exemplar implementations in the community.

Supporting discovery through good data management

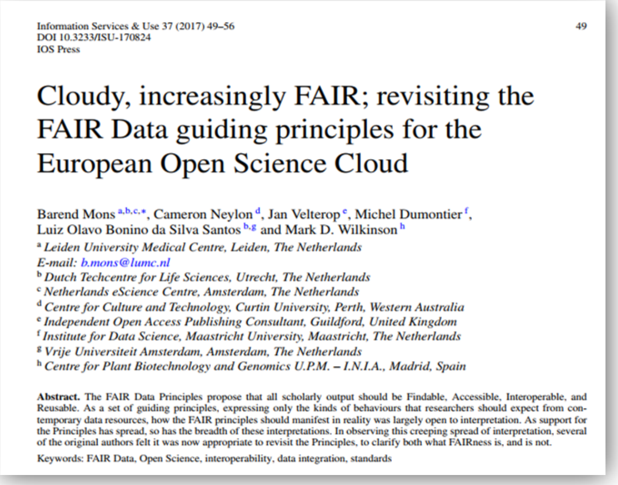
Good data management is not a goal in itself, but rather is the key conduit leading to knowledge discovery and innovation, and to subsequent data and knowledge integration and reuse by the community after the data publication process. Unfortunately, the existing digital ecosystem surrounding scholarly data publication prevents us from extracting maximum benefit from our research investments (e.g., ref. 1). Partially in response to this, science funders, publishers and governmental agencies are beginning to require data management and stewardship plans for data generated in publicly funded experiments. Beyond proper collection, annotation, and archival, data stewardship includes the notion of 'long-term care' of valuable digital assets, with the goal that they should be discovered and re-used for downstream investigations, either alone, or in combination with newly generated data. The outcomes from good data management and stewardship, therefore, are high quality digital publications that facilitate and simplify this ongoing process of discovery, evaluation, and reuse in downstream studies. What constitutes 'good data management' is, however, largely undefined, and is generally left as a decision for the data or repository owner. Therefore, bringing some

Wilkinson et al. (2016) The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data https://doi.org/10.1038/sdata.2016.18

Key point:
FAIR means FAIR
for machines (e.g. machine-readable
metadata) and only secondarily
for humans…

In 2017, 2nd paper:

i. Re-useless data
ii. Findable (PID)
iii. FAIR metadata
(PID + machine readable MD)
iv. FAIR: restricted access
v. FAIR: open access
vi. FAIR: open access, functionally linked
‚Internet of FAIR data and services‘



Information Services & Use 37 (2017) 49–56
DOI 10.3233/ISU-170824
IOS Press

49

Cloudy, increasingly FAIR; revisiting the FAIR Data guiding principles for the European Open Science Cloud

Barend Mons [a,b,c,*], Cameron Neylon [d], Jan Velterop [e], Michel Dumontier [f], Luiz Olavo Bonino da Silva Santos [b,g] and Mark D. Wilkinson [h]
[a] *Leiden University Medical Centre, Leiden, The Netherlands*
E-mail: b.mons@lumc.nl
[b] *Dutch Techcentre for Life Sciences, Utrecht, The Netherlands*
[c] *Netherlands eScience Centre, Amsterdam, The Netherlands*
[d] *Centre for Culture and Technology, Curtin University, Perth, Western Australia*
[e] *Independent Open Access Publishing Consultant, Guildford, United Kingdom*
[f] *Institute for Data Science, Maastricht University, Maastricht, The Netherlands*
[g] *Vrije Universiteit Amsterdam, Amsterdam, The Netherlands*
[h] *Centre for Plant Biotechnology and Genomics U.P.M. – I.N.I.A., Madrid, Spain*

Abstract. The FAIR Data Principles propose that all scholarly output should be Findable, Accessible, Interoperable, and Reusable. As a set of guiding principles, expressing only the kinds of behaviours that researchers should expect from contemporary data resources, how the FAIR principles should manifest in reality was largely open to interpretation. As support for the Principles has spread, so has the breadth of these interpretations. In observing this creeping spread of interpretation, several of the original authors felt it was now appropriate to revisit the Principles, to clarify both what FAIRness is, and is not.

Keywords: FAIR Data, Open Science, interoperability, data integration, standards

Mons, Barend et al. (2017) ‘Cloudy, Increasingly FAIR; Revisiting the FAIR Data Guiding Principles for the European Open Science Cloud’: 49 – 56. https://doi.org/10.3233/ISU-170824

FAIR Data (and Software) Principles II

→ FAIR: not a standard

→ Different approaches

→ About FAIRness for machines (and humans)

„*Partly FAIR may be FAIR enough*“

Mons, Barend et al. (2017) ‘Cloudy, Increasingly FAIR; Revisiting the FAIR Data Guiding Principles for the European Open Science Cloud’: 49 – 56. https://doi.org/10.3233/ISU-170824

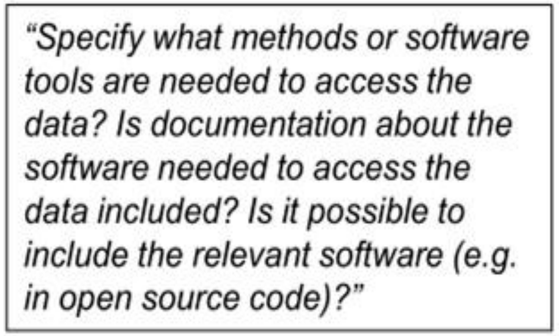
FAIR for Software?

EUROPEAN COMMISSION
Directorate-General for Research & Innovation

H2020 Programme

Guidelines on

FAIR Data Management in Horizon 2020



"Specify what methods or software tools are needed to access the data? Is documentation about the software needed to access the data included? Is it possible to include the relevant software (e.g. in open source code)?"

- Software quality guidelines existed for decades in military, industry, academia & FLOSS initative
- FLOSS = Free/Libre and Open Source Software

Examples:

- ISO 9000-3, 9126-1, 25010:2011
- GNU Quality Code
- ECSS Software Product Assurance
- CLARIAH software quality guidelines

FAIR for Software?

Example:
Jiménez RC et al. (2017) Four simple recommendations to encourage best practices in research software. F1000Research 2017, **6**:876: https://doi.org/10.12688/f1000research.11407.1

Open Source Software (OSS) Recommendations:

1. Make source code publicly accessible from day one
→ *Git, Cloud, Hub, Project Page…*

2. Make software easy to discover by providing software metadata via a popular community registry
→ *e.g. via DataCite DOI*

3. Adopt a licence and comply with the licence of third-party dependencies
→ *Apache, BSD 2&3, GNU GPL&LGPL, MIT …*

4. Define clear and transparent contribution, governance and communication processes
→ *e.g. Project website includes information*

OSS Recommendations = FAIR ?

Remember:
FAIR data principles have emphasis on enhancing machine-readability.
→ This emphasis is not present in the OSS Recommendations (expect machine readable software metadata to be available via software registries)

OSS focus:
→ Uptake of best practices
→ Measurability
→ Reuseability

Measures for increasing FAIRness

Research Data	Research Software
Data Management Plan	Software Management Plan
PIDs & Machine Readable Metadata	PIDs & Machine Readable Metadata
Machine Readable Data(sets) in Data Repositories	Machine Readable Software/Code in Software Repositories
Data Licences	Software Licences
Documentation ?	Documentation ? → Version control!

What is a Data management plan (DMP)?

A Data management plan …

- might be required by funding bodies (NSF, EU H2020)
- is a (formal) document developed at the start of a research project which outlines all aspects of data created/used
- must be updated throughout the course of research

Future:

- Post-Static/Dynamic/Machine-Actionable DMPs with PIDs (DOI, ORCiDs)

RDMO DMP ONLINE DMPTool Build your Data Management Plan

Source: pixabay.com, pixabay licence

Common checklist (all DMPs):

- Administrative information
- Data collection
- Documentation & metadata
- Ethics & legal compliance
- Storage & backup
- Selection & preservation
- Data sharing
- Resources & responsibilities

Stakeholders of a DMP:

- ➢ Researchers
- ➢ Institutions/Organizations
- ➢ Repositories/Infrastructure
- ➢ Funders
- ➢ Publishers

Software Management Plan (SMP)

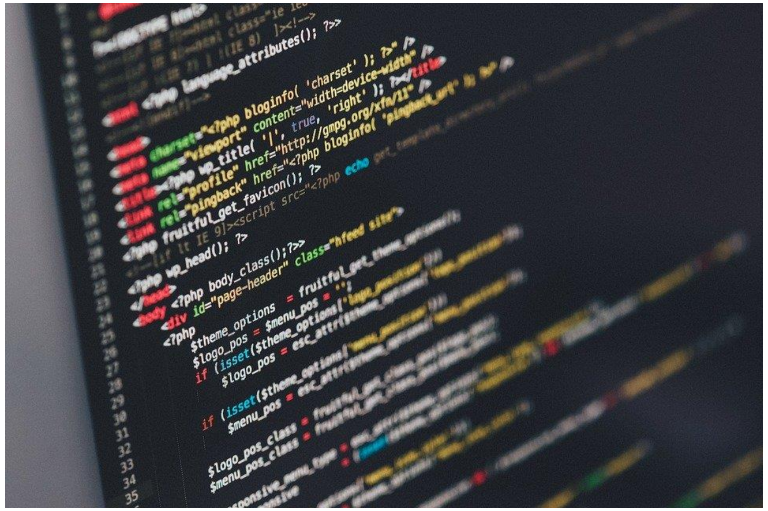
Adapted after recommendations of the Software Sustainability Institute, see:
The Software Sustainability Institute (2018) Checklist for a Software Management Plan (Version 0.2). Zenodo. http://doi.org/10.5281/zenodo.1460504

Minimum:

- Information on **outputs, documentation & related material**
- **Institution/Person responsible** for software release
- **Development/revision /version control process** used
- **PID & licence** for published version

Good practice:

- Identify software development model to be used
- Identify possible external software used & associated licences
- Method used to accept each output (e.g. review process)
- Dependencies between outputs and with external dependencies
- Major risks that might impact on the delivery of the outputs



Source: pixabay.com, pixabay licence

Stakeholders of a SMP:

➢ Developers/Researchers
➢ Institutions/Organizations
➢ Repositories/Infrastructure
➢ Funders
➢ Publishers

PIDs are everywhere:

Researcher IDs

ORCID

Scopus

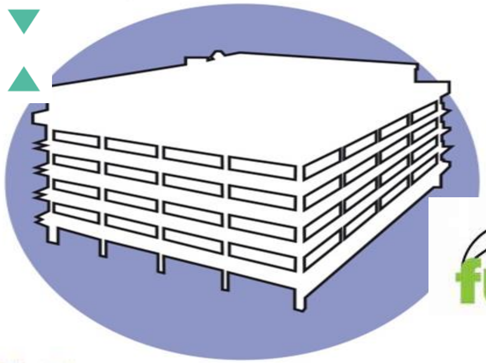
RESEARCHERID

THOMSON REUTERS

isni

Organisation IDs, Funder IDs

ROR



fundref

Ringgold

Identify

GRID

Resource IDs (articles, data, software, …)

doi®

ePIC

Persistent Identifiers for eResearch

Handle.Net®

DataCite

FIND, ACCESS, AND REUSE DATA

ARK (Archival Resource Key)

Crossref

URN-SERVICE

PICHE – Persistent Identifiers for Cultural Heritage Entities

A PID is

- Provenance
- Metadata
- Policies & Guarantees
- Machine readability
- Metrics

Source: pixabay.com, pixabay licence

Researchers & developers should know that…

Provenance means validation & credibility – a researcher/developer should comply to good scientific practices and be sure about what should get a PID (and what not).

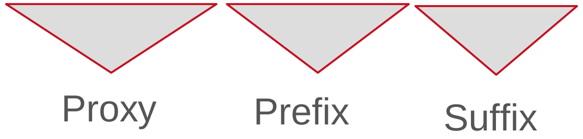
Metadata is central to visibility and citability – metadata behind a PID should be provided with consideration.

Policies behind a PID system ensure persistence in the WWW - point. At least metadata will be available for a long time.

Machine readability will be an essential part of future discoverability – resources should be checked and formats should be adjusted (as far possible).

Metrics (e.g. altmetrics) are supported by PID systems.

https://doi.org/10.15468/dl.n1glrt



Proxy Prefix Suffix

GitHub + Zenodo.org = DOI

- Official integration thanks to Codemeta project: science.Mozilla.org/projects/codemeta

- Intrinsic IDs (e.g. Git’s SHA1 hashes) vs. “minted” PIDs
  - technical vs. procedural persistence

- Zenodo: file backup & persistent landing page for each release version, powered by CERN
- Detailed guide: https://guides.github.com/activities/citable-code/ & further reading: https://genr.eu/wp/cite/

- DOI minting requires metadata information
  → Use https://search.datacite.org/works?resource-type-id=software
  → Research software with a DOI listed in results
- DOI used for persistent citation

Citing software – the background

Why citing software?

→ Ability to **replicate research that has used software**, knowing exactly the version of a research software used
→ Improve research software itself — help software developers (speed, lessons learned, …)

→ FORCE11 recommendations: Software Citation Implementation Working Group
Smith, Katz & Niemeyer 2016: Set of software citation principles across disciplines & venues
→ https://doi.org/10.7717/peerj-cs.86 contains
→ Use cases & discussion, suggestions on how to apply the principles
→ 6 Principles: Importance, Credit & Attribution, Unique Identification, Persistence, Accessibility, Specificity

Note:
Some communities already have their own conventions, e.g. R and CRAN
Examples: https://www.rdocumentation.org/packages/utils/versions/3.3/topics/citation &
https://cran.r-project.org/web/packages/knitr/citation.html

→ Software & data are similar in with regard to credit & metrics, but both have traditionally not been cited in publications
→ Citation practice needs to change

How To: Best practices for software citation

Making software citable

i. Publish it – if it's on GitHub, follow steps in https://guides.github.com/activities/citable-code/
ii. Otherwise, submit it to Software repository with appropriate metadata, & get a DOI
iii. Create a CITATION file (e.g. https://citation-file-format.github.io/), update the README
iv. Integrate software citation in researcher profile, e.g. ORCiD (https://orcid.org)
v. Optional: Writing a software paper for publication in a software journal

Citing someone else's software

Check for a CITATION file or README; if this says how to cite the software itself, if not, do your best following the principles:

- Try to include all contributors to the software (maybe by just naming the project)
- Include method for identification that is machine actionable, globally unique & interoperable → ideally via a PID(DOI), or URL to a release or product number
- If there's a landing page including metadata, point to that (not to software directly)
- Include specific version/release information
- If there's a software paper, you can cite this too, but not in place of citing the software

(Data &) Software licences I

Purpose of licences – mostly the same for research data & research software:

To share

→ Practice FAIR

To protect & restrict the use

→ Disallow commercialization or any other further use

→ Enable commercialization

To get credit & acknowledgement

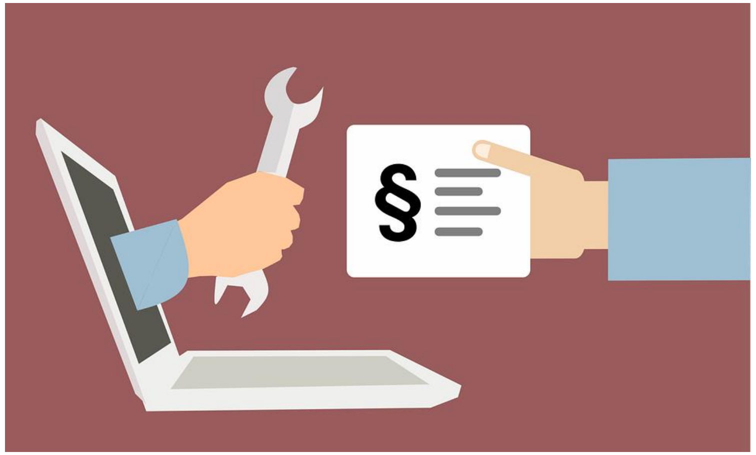
→ Register amount of use & citations

Refuse warranties

Refuse liability

Clarify which license is best for you and other stakeholders

Deliver a contract with your work



Source: pixabay.com, pixabay licence

(Data &) Software licences II

Note/Disclaimer: *Nothing in this presentation is intended as legal advice. When in doubt, ask your institution’s/employer’s/funder’s legal counsel!*

Research Data:

- “*As open as possible, as closed as necessary*” (new EU H2020 credo)
  → there is a shift from ‘open data’ towards ‘FAIR data’
- Special protection & ethical questions regarding ‘sensitive’ data & ‘mission oriented research’
- **Urheberrecht - Geistige Schöpfungshöhe might / or might not apply**
- **Other laws which might apply**: Patent law, Data privacy law, Contract law, Constitutional law, Business/trade law, *Sui generis* database right, …

→ For data accompanying scientific publications: Using Creative Commons licences are often recommended

↕

Research Software:

- **Creative work (mostly)! → Urheberrecht - Geistige Schöpfungshöhe likely to apply!**
  → Copyright protects the expression of an idea (in source code & object code)
  → A licence is a way for a copyright holder to grant rights (e.g. to copy/modify/distribute) to other people
  → End users are covered by whatever license you place on software/code you write
- Other laws which might apply: Patent law, Data privacy law, Contract law, Constitutional law, Business/trade law, *Sui generis* database right, …

Software licences III

Note/Disclaimer: *Nothing in this presentation is intended as legal advice. When in doubt, ask your institution's/employer's/funder's legal counsel!*

Some licensing issues:

- Development of complex open source solutions → adapting & integrating multiple existing components
- Resulting application/solution may look as a single program from the user point of view, but is in fact a combined work
  → Different components may be covered by different licences;
  → Question if components are compatible & legally interoperable?

- **Licences for open source software: 2 families - Copyleft licences vs. Permissive licences**
  - Copyleft: Impose the use of the same licence as soon as the distributed work is a derivative of the covered work (e.g. GNU GPLs and the EUPL)
  - Permissive: Non-copyleft open source license, compatible with most other licences, tolerating to merge, combine or improve the covered code and to re-distribute it under different licences (e.g. BSD-style, MIT/X11-style, ASLv2)

- **Get help: e.g. Open Source Initiative (OSI)**
  - Promote awareness & importance of non-proprietary software; review-process
  - OSI Approved licence trademark & program;
    >80 approved licences: https://opensource.org/licenses/alphabetical

Software licences IV

Note/Disclaimer: *Nothing in this presentation is intended as legal advice. When in doubt, ask your institution's/employer's/funder's legal counsel!*

Examples Copyleft licences:

- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- Eclipse Public License (EPL)
- Mozilla Public License 2.0 (MPL)
- Common Development and Distribution License (CDDL)
- GNU Affero General Public License (AGPL)
- European Union Public Licence (EUPL)

Examples Permissive licences:

- Apache (Software) License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- MIT license

Note:

→ Some 'data' repositories also offer 'software' licences, as they treat data as software

→ Not all licences are compatible; see licence specific compatibility (upstream/downstream) matrices & information, and constitution of an exception lists

Licence provision – Example:

```
/*
 * EasyWave - A realtime tsunami simulation program with GPU support.
 * Copyright (C) 2014  Andrey Babeyko, Johannes Spazier
 * GFZ German Research Centre for Geosciences (http://www.gfz-potsdam.de)
 *
 * Parts of this program (especially the GPU extension) were developed
 * within the context of the following publicly funded project:
 * - TRIDEC, EU 7th Framework Programme, Grant Agreement 258723
 *   (http://www.tridec-online.eu)
 *
 * Licensed under the EUPL, Version 1.1 or - as soon they will be approved by
 * the European Commission - subsequent versions of the EUPL (the "Licence"),
 * complemented with the following provision: For the scientific transparency
 * and verification of results obtained and communicated to the public after
 * using a modified version of the work, You (as the recipient of the source
 * code and author of this modified version, used to produce the published
 * results in scientific communications) commit to make this modified source
 * code available in a repository that is easily and freely accessible for a
 * duration of five years after the communication of the obtained results.
 *
 * You may not use this work except in compliance with the Licence.
 *
 * You may obtain a copy of the Licence at:
 * https://joinup.ec.europa.eu/software/page/eupl
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the Licence is distributed on an "AS IS" basis,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the Licence for the specific language governing permissions and
 * limitations under the Licence.
 */
```

Copyright note

Project description

Licence title

Additional Provision

Licence specification (rights to copy/modify/ distribute

Version control & Project management

Source Code:

- Self-documenting/-explaining a project’s evolution
- Git often used because of strongest network effects, with easy publication & collaboration opportunites
- Pull/Merge Requests enable smooth review workflow & automation potential
- Git, GitHub, Gitlab enable issue tracker, website hosting, project management, etc.
- Issue = idea, discussion, problem report, question, etc. → Labels, assignees, milestones / due dates, etc.
- (Peer-)Reviewing pull/merge request can be used for knowledge transfer within team

Beyond Code, e.g. documentation:

- Text documents: Markdown, LaTeX, GitHub/Lab Pages
- Alternative to fast-syncing tools like EtherPad, HackMD, GDocs, etc.
- Also: Overleaf.com, GitBook.com, Authorea.com, PenFlip.com, others

Summary / On the FAIR principles

- FAIR refers to ‘as open as possible, as closed as necessary’

- There are different degrees of FAIRness, as research disciplines, resource types (e.g. **data and software**) and their requirements are strongly varied - but the shared goal is good scientific practice

- FAIR (in its origins) focuses first and foremost on machine to machine interactions, only secondary on human to machine (or human to human) interactions

- DMPs/SMPs, PIDs, version control, documentation & a licence help to keep data/software FAIR

LEIBNIZ INFORMATION CENTRE
FOR SCIENCE AND TECHNOLOGY
UNIVERSITY LIBRARY

TIB

Thank you!

Contact information:
Angelina.Kraft@TIB.eu
T +49 511 762-14238

Slides derived/adapted from Leinweber, Kraft, Kuzak, Johnston, Hammitzsch & Förstner (2018). **FAIR Data and Software: A Carpentries-based workshop at TIB, Hannover**. Zenodo. http://doi.org/10.5281/zenodo.3707745

Leibniz
Leibniz Association