HELMHOLTZ
Open Science

Helmholtz Open Science Briefing

4th Helmholtz Open Science Forum: Research Software

May 2023

Report

Impressum

Die Onlineversion dieser Publikation finden Sie unter:
https://doi.org/10.48440/os.helmholtz.072

Verfasser:innen

Christoph Bruch, Uwe Konrad, Christian Meeßen, Lena Messerschmidt,
Johannes Reuther, Ute Schelhaas, Tobias Schlauch, Almut Scholz,

Herausgeber

Helmholtz Open Science Office

Redaktion

Roland Bertelmann, Christoph Bruch, Lena Messerschmidt, Lea Maria Ferguson,
Heinz Pampel, Antonia C. Schrader, Paul Schultze-Motel, Nina Leonie Weisweiler

Kontakt

Helmholtz Open Science Office
c/o Helmholtz-Zentrum Potsdam
Deutsches GeoForschungsZentrum GFZ
Telegrafenberg, 14473 Potsdam
E-Mail: open-science@helmholtz.de

Stand

12.12.2023. Version. 1.0

Lizenz

HELMHOLTZ

Open Science

Content

HELMHOLTZ

Open Science

Program

Table 1: Program of the 4th Helmholtz Open Science Forum: Research Software Policies

Time	Agenda	Speaker
Monday, May 22nd		
12:00 – 13:00	Lunch	
13:00 – 13:10	Welcome and organizational matters	
13:10 – 14:40	Status of each center	
14:40 – 15:00	Break	
15:00 – 15:30	Discussion / Evaluation of status round	
15:30 – 16:00	Presentation of the research software policy of the FZJ and its implementation	Ute Schelhaas FZJ
16:00 – 16:30	Process for sharing and providing software at GFZ	Almut Scholz GFZ
16:30 – 17:00	Break	
17:00 – 18:00	Helmholtz Support Services (HIFIS & HiRSE_PS)	Uwe Konrad, HZDR Johannes Reuther, HZB
18:00	Closing and optional dinner	
Tuesday, May 23rd		
09:00 – 09:15	Presentation of the Helmholtz Research Software Directory	Christian Meeßen GFZ
09:15 – 09:30	Presentation of the Helmholtz Incubator Software Award	Uwe Konrad HZDR
09:30 – 10:30	Software Licensing	Tobias Schlauch DLR
10:30 – 11:00	Break	
11:00 – 12:30	Discussion of specific issues, e.g. copyright, attribution of authorship	Uwe Konrad HZDR
12:30	Closing	

Executive Summary

The workshop was organized against the background that the Helmholtz Centers are aiming to adopt a research software policy by 2025. It is to provide an update on the status at the individual centers and provide an opportunity to learn from their experiences and to network to support each other in reaching the goal of deciding and implementing a research software policy. In this sense Reports form three centers were supplemented by the introduction of the Helmholtz Incubator Software Award and of services offered by Helmholtz Projects HIFIS and HiRSE-PS.

All but one Center sent at least two representatives to participate in the workshop. The one center not represented could not send representatives because of a parallel meeting. In the invitation for the workshops all centers were asked to send participants who could represent different communities of the center. This request reflected the desirability to design the process of formulating the research software policy as inclusive as possible.

Appendices — Presentation Slides

Appendix 1
Research Software Policy and Implementation at FZJ
(Ute Schelhaas, FZJ)

Appendix 2
Process on Dissemination of Research Software at GFZ
(Almut Scholz, GFZ)

Appendix 3
HIFIS Software Services
(Uwe Konrad, HZDR)

Appendix 4
Helmholtz Platform for Research Software Engineering - Preparatory Study (HiRSE_PS)
(Johannes Reuther, HZB)

Appendix 5
Increasing the visibility of Research Software: The Helmholtz Research Software Directory
(Christian Meeßen, GFZ)

Appendix 6
Software Licensing
(Tobias Schlauch, DLR)

HELMHOLTZ

Open Science

Appendix 1 — Research Software Policy and Implementation at FZJ

(Ute Schelhaas, FZJ)

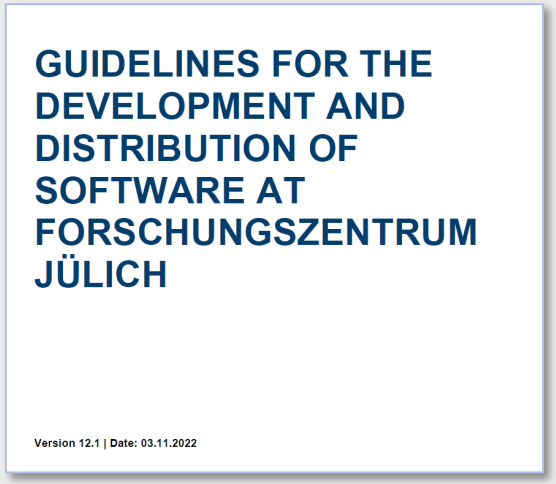
RESEARCH SOFTWARE POLICY AND IMPLEMENTATION AT FZJ

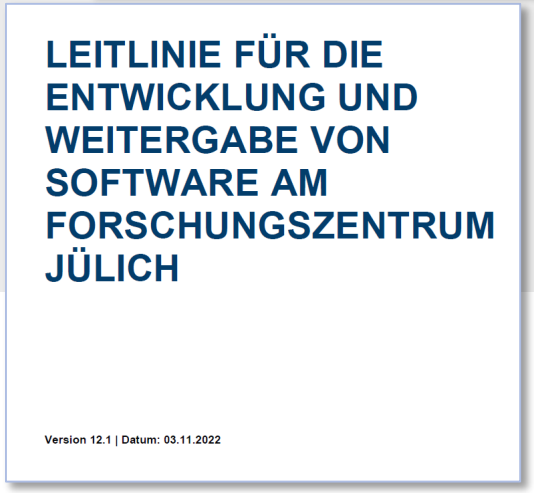
HELMHOLTZ OPEN SCIENCE FORUM "RESEARCH SOFTWARE POLICIES" 22-23 MAY 2023

22 MAY 2023 |

UTE SCHELHAAS
CORPORATE DEVELOPMENT - INNOVATION AND STRATEGY (UE-I)
FORSCHUNGSZENTRUM JUELICH

WHERE WE ARE

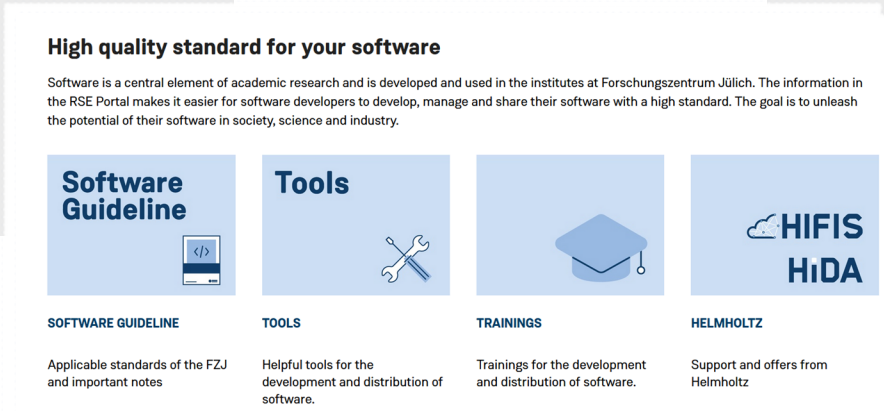




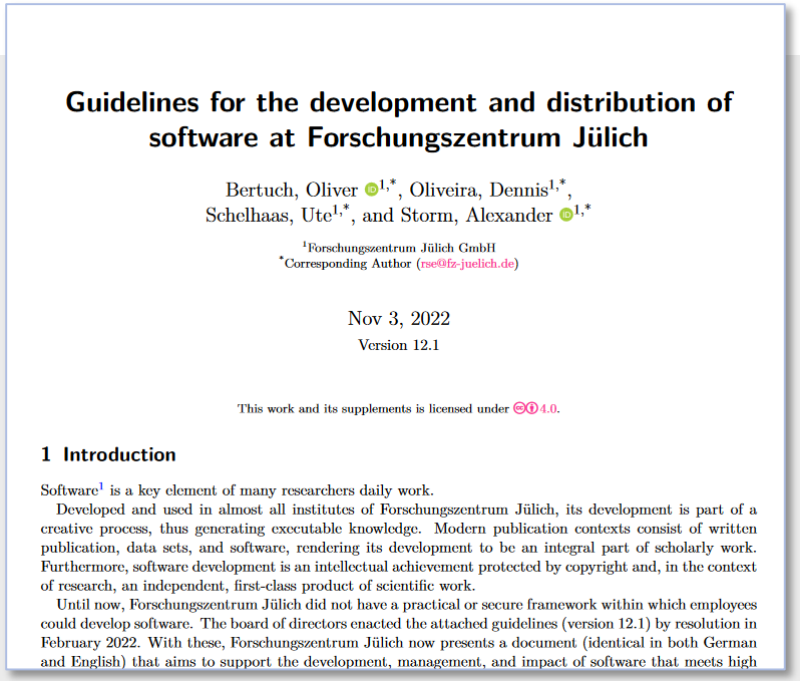
RSE-PORTAL

ENG

Information, support and contacts on the topic of Research Software Engineering (RSE).



https://intranet.fz-juelich.de/de/themen/rse



https://juser.fz-juelich.de/record/916100

...still a lot to do…

WHERE WE STARTED (OCTOBER 2020)

- FZJ is a major interdisciplinary research institutions with more than 7.000 employees
- Software is developed and used in almost all institutes
- Most scientists who write code are not trained software developers
- No standardized recommendations for action on the topic of development and transfer of software at FZJ
- Constantly increasing need for individual advice on various software projects by UE-I, legal dep. + central library
- Licensing:
  - Open source: handled individually by institutes
  - Proprietary licensing: preparation + negotiation of contracts by UE-I in cooperation with institutes
- Conditions and consequences of transfer, copyright and rights of third parties are most frequent issues
- Increasing number of software-based start-up projects
- Topic had no natural “Heimat” in the organization
- and no extra resources

“Write down, what applies in Jülich (… but not what is generally valid)”

Wanted: "Certainty and safety of action, empowerment and consistent standards on the subject of software development“

TEAM

UE received a mandate from the Board of Directors

Working Group Coordination + Implementation

+

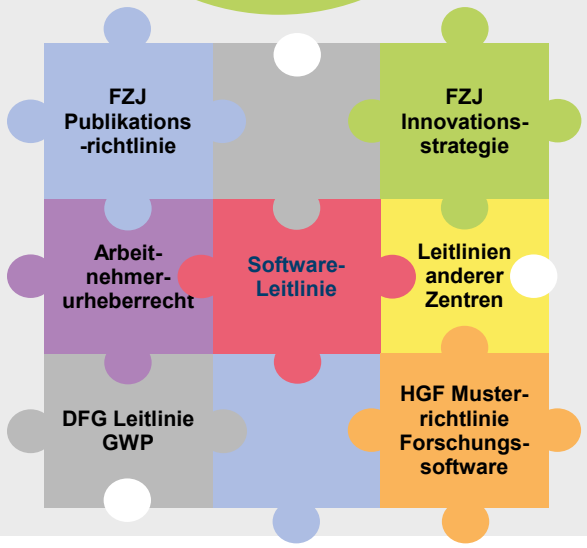
UE	Dennis Oliveira, Ute Schelhaas
R	Alexander Storm
ZB	Oliver Bertuch
WTR	Sascha Pust
FDM	Sven Rank

Expert Group Software

+

Literature, good practice, colleagues …

15 software developer with distinct backgrounds, at different levels of hirarchy and with various R&D topics



SOFTWARE-GUIDELINE AT FZJ

“.. involve all internal committees in the right order of priority”

Time Structure

Okt. 2020

Vorbereitung
- Vorankündigung des Themas durch **Prof. Marquardt** in der GD-Runde
- Einladung der Teilnehmer für AG Koordination und Umsetzung und Expertenkreis Software

Nov.-Dez. 2020

Auftakttreffen der AG Koordination und Umsetzung und des Expertenkreises Software
- Skizzierung des Prozesses
- Abstimmung des Vorgehens
- Rollendefinierung

Jan. 2021

1. Workshop Expertenkreis Software
- Erarbeitung von Inhalten, Passfähigkeit und Bedürfnisse des FZJ
- Absteckung der Ziele

Jan.-Apr. 2021

Erarbeitung des ersten Leitlinien-Entwurfs AG Koordination und Umsetzung
- Verfassen der Leitlinie unter Zusammenarbeit der Vertreter der AG-Koordination und Umsetzung

Mai 2021

Review-Prozess durch Expertenkreis Software
- Begutachtung, Korrektur und Kommentierung des Leitlinienentwurfs

Jun.-Okt. 2021

Überarbeitung Leitlinien-Entwurf AG Koordination und Umsetzung
- Anpassen des Entwurf unter Berücksichtigung der erhaltenen Inhalte

Wiederholung der Iterationsschleifen
- 2. Workshop mit dem Expertenkreis Software
- Einzelgespräche mit ausgewählten Software-Experten zwecks Feedback

Okt. 2021-Jan. 2022

Finalisierung
- Finalisierung des Entwurfs durch AG Koordination und Umsetzung
- Vorstellung bei **VS-Sitzung GzS**
- Vorstellung GD-Runde

SOFTWARE-GUIDELINE AT FZJ

Table Of Contents

Application Classes and Quality Measures

- Classification according to application class
  - Dependent mainly on use/distribution
  - » Switching is possible
- » The application classes (AC) help to define appropriate software quality measures
  - » Standards + measures are defined
  - » Real-life examples and decision-making tools
  - » Brief Overview:

Application class (AC)	0	1	2	3
Development	Internally FZJ	Internally FZJ	Internally FZJ or with third contributors	Internally FZJ or with third contributors
Use	Personally and internally in the project team	In the institute	Software from third-party projects in which maintenance and long-term usage matter beyond the project	Product characteristics
Quality measures	Compliance with legal aspects	Version control system is necessary. Further development is possible for uninvolved parties.	Maintainability and usability are given. The rights of use and exploitation are held by FZJ.	Test automation, release + maintenance management are in place
Distribution	Not intended for use outside of the development team	Only in a strictly limited context, e.g. in the institute or to the institution of the supervisors. In case of required transfer to persons outside the FZJ, if possible with license	Yes, with license (OSS or proprietary)	Yes, with license (OSS or proprietary)
Example	Code on the smallest scale, individual functions, simple scripts	Software from dissertations with demonstration character	Software publications. ¶ Software developed and used in cooperation with partners.	Software that is intended to be commercially exploited, e.g. in a spin-off ¶ Software that is developed in large open source projects

Distribution, Release and Citation

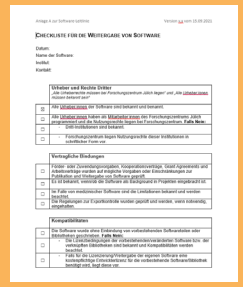
- » Procedures and main aspects
- » What needs to be considered before sharing
- » Contractual situation, legal aspects, FAIR, etc.
- » Licensing:
  - » Open Source License vs. Proprietary License
  - » *Excursus*: Dual Licensing (Expl. Android)
  - » Approval + Release Processes:
    - » Open Source Licensing: Institute (Analogous to text publications)
    - » Proprietary Licensing: UE-I

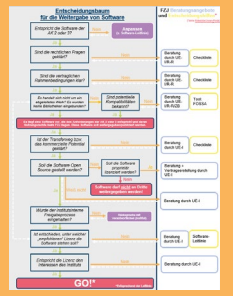
designed by Freepik from Flaticon

- » Citation:
  - » Citation and referencing are core elements for reproducibility of results
  - » *Author, name of software, PID, release date, [release number if available].*

Support and Services at FZJ

- » Consultancy services offered by UE, R, ZB
- » Decision-making tools:
  - » Check lists
  - » Decision tree for distribution of Software





- » RSE-Portal
  - » Compilation of documents
  - » Internal/external workshops & tranings
  - » FAQs and Tools
  - » Main contact for help, general E-Mail

WHERE WE ARE

Achievements

- √ Guideline approved and implemented
- √ RSE- Portal up and running
- √ Postings in the internal news
- √ Positive effects are seen in the counselling
- √ Topic Research-Software-Policy has digital home
- √ Integrated approach helped to build network
- √ 1-2 FTE at Supercomputing Center for RSE-support (under negotiation)
- √ Obligation to register software output - Jülich Data → Indikatorik

Challenges

- ○ resources (time and budget)
- ○ keeping the experts network alive
- ○ increase the awareness for RSE
- ○ „voluntary“ guideline vs. „strict“ ruling
- ○ organisational home still unclear
- ○ set up own training courses
- ○ establish RSE support services

REASEARCH SOFTWARE POLICY AND IMPLEMENTATION AT FZJ

Next Steps

- Joint actions planned for the Open Science Week in October
- Focus on Engineering support in JSC‘s RSE group
- Extent the portal‘s content and activities to community topics and guidelines for administrative departments

THANK YOU FOR YOUR ATTENTION !

u.schelhaas@fz-juelich.de
d.oliveira@fz-juelich.de
a.storm@fz-juelich.de
o.bertuch@fz-juelich.de

SOFTWARE-GUIDELINE AT FZJ

Workshop results for the implementation of the guidelines and the portal

Expertenkreis Software

Inhalt

Portal in englischer Sprache

Dokumente, die am FZJ gelten

Einstieg niederschwellig, für alle Level und den gesamten Software-Lebenszyklus

Ansprechpartner, Erste Hilfe und Zuständigkeiten

Links zu allen allgemeingültigen Informationen, Tools, Papern etc.

Kalender mit internen und externen Schulungen

FAQ

Links zu Angeboten und Gruppen der HGF

Jülicher Infrastruktur, z.B. JuGit

Professionalisierung

Community-Themen

www.flaticon.com

SOFTWARE-GUIDELINE AT FZJ

Workshop results for the implementation of the guidelines and the portal

Expertenkreis Software

Inhalt

Professionalisierung

- Programmieren ist mehr als Softskill
- Sichtbarkeit bestehender Angebote erhöhen
- Synergien zwischen Instituten schaffen
- Best practice Beispiele präsentieren
- Augenmerk auf neue MA ebenso wie auf Sicherung von Knowhow trotz Fluktuation
- zentrale Schulungs-Angebote für Basistechniken am FZJ notwendig

Community-Themen

www.flaticon.com

SOFTWARE-GUIDELINE AT FZJ

Workshop results for the implementation of the guidelines and the portal

Expertenkreis Software

Inhalt

Professionalisierung

Community-Themen

Netzwerk ist wichtiger als Tools

Chat- und Blog- Funktion für „User-Content“

Software Katalog

www.flaticon.com

Appendix 2 — Process on Dissemination of Research Software at GFZ

(Almut Scholz, GFZ)

Helmholtz Open Science Forum
Forschungssoftware-Policies
Berlin, 22./23.05.2023

Process on Dissemination of Research Software at GFZ

Almut Scholz, Legal Department
Martin Hammitzsch, eScience
Dr. Christian Meeßen, eScience/HIFIS
Helmholtz-Centre Potsdam – GFZ German Research Centre for Geosciences

Building blocks of our SW policy framework

Policy on Use and Licensing of Research Software
(since April 2020; update in April 2023)

Guidelines on Research Software Development, use and dissemination
(since January 2019; update in April 2023)

Establishment of services for counselling and support
(since January 2019)

Software distribution process and Materials for licensing and exploitation of research software
(since January 2019; since 2021 online-form for dissemination requests)

Policy as overview with strategic goals (“Richtlinie”)

1. Strategic measures for the valorisation of software
2. Principles for development, use, exploitation and licensing of research software
3. Support and further information (as Meta-level-Document)

GFZ
Helmholtz-Zentrum Potsdam
Helmholtz-Zentrum Potsdam
Deutsches GeoForschungsZentrum

Policy on Use and Licensing of Research Software

Version 1.1, 12. April 2023;

Version 1.0 from 10. March 2020, valid from 1. April 2020

Imprint

The online version of this publication can be found at: http://doi.org/10.5880/GFZ.12.4.2023.001

Contact and editorial office

Martin Hammitzsch, Werner Köckeritz, Jörn Krupa, Almut Scholz

Contact

Helmholtz Centre Potsdam, GFZ German Research Centre for Geosciences
Telegrafenberg, 14473 Potsdam
E-mail: software-legal@gfz-potsdam.de

License

All texts in this publication, excluding quotations, are licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license agreement.
see: https://creativecommons.org/licenses/by/4.0

www.gfz-potsdam.de

HELMHOLTZ

Strategic goals for valorisation

a) systematic recording with link to internal reporting (i.e. SW-index)

b) software repository, access to internal and external users via software portal

c) resources for the upkeep, maintenance and user-support are to be taken into account in the section-budgets

d) monetary incentives should be introduced, e.g. for commercially usable software

e) non-monetary incentives shall be strengthened at all levels

f) DOIs for certain releases, e.g. for better evaluation of performance support

Principles for dealing with software at GFZ

- *when developing*: ensuring comprehensive grant of rights to GFZ; when integrating FOSS – permissive licenses are recommended to keep all exploitation options
- *when using:* clarifying all necessary rights for use at GFZ
- *before publishing:* examining the potential of commercial exploitation by developers and Transfer&Innovation. Process of *Guidelines on Research SW* to be followed. Protection through appropriate licenses requested.
- *when licensing to third parties:* license recommendations for classic license agreement or FOSS license (i.e. scientific community), observing license dependencies

License recommendations

- If possible and adequate: proprietary license.

- If open-source:
  - European Union Public License (EUPL)
  - Strong Copyleft Licenses (AGPL, GPL)
  - Licenses with restricted copyleft (LGPL, Eclipse Public License EPL)
  - Licenses without copyleft effect / permissive (Apache)

- For joint developments within Helmholtz - Research Field *Earth and Environment:*

  Helmholtz Earth and Environment Software Infrastructure License (HEESIL)

EUPL advantages

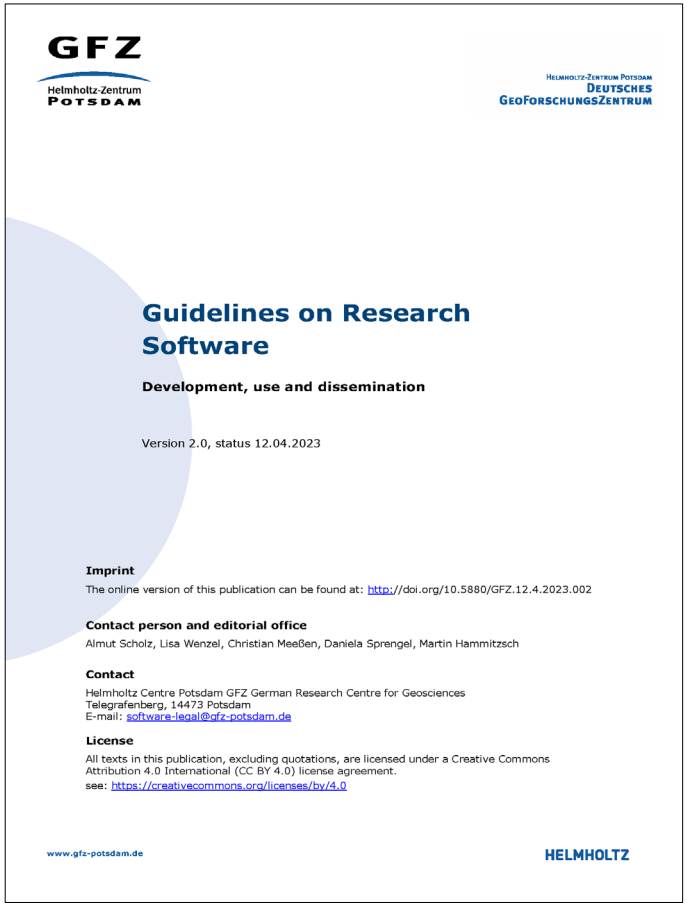
- first open source license to be released by an international governing body
- available and valid in 23 languages
- valid in all EU Member States, with precisely formulated limitations of liability or warranty, and conform to EU law requirements
- downstream compatible with many other copyleft licenses, including business-friendly OSS licenses and especially GPL-v.2/GPL-v.3
- covers also use case of software as a service/cloud services (without distribution of software)

Helmholtz Earth and Environment Software Infrastructure License - HEESIL

- based on EUPL.
- enables the sharing and joint development of software within the Research Field Earth and Environment.
- maintains the software protection.
- allows a later joint clarification on the application of a suitable license.
- A use by other Helmholtz centers outside the Research Field Earth and Environment as well as a transfer to other third parties is still possible after agreement of the centers involved in the development.

Guidelines (“Leitfaden”) for

- developing SW
- use of SW at the GFZ
- dissemination of SW outside the GFZ
  - approval and consultation process via online form
- obtaining rights of use at the GFZ



GFZ
Helmholtz-Zentrum
POTSDAM

HELMHOLTZ-ZENTRUM POTSDAM
DEUTSCHES
GEOFORSCHUNGSZENTRUM

Guidelines on Research Software

Development, use and dissemination

Version 2.0, status 12.04.2023

Imprint

The online version of this publication can be found at: http://doi.org/10.5880/GFZ.12.4.2023.002

Contact person and editorial office

Almut Scholz, Lisa Wenzel, Christian Meeßen, Daniela Sprengel, Martin Hammitzsch

Contact

Helmholtz Centre Potsdam GFZ German Research Centre for Geosciences
Telegrafenberg, 14473 Potsdam
E-mail: software-legal@gfz-potsdam.de

License

www.gfz-potsdam.de

HELMHOLTZ

Developing Software – recommendations for institution-wide standards

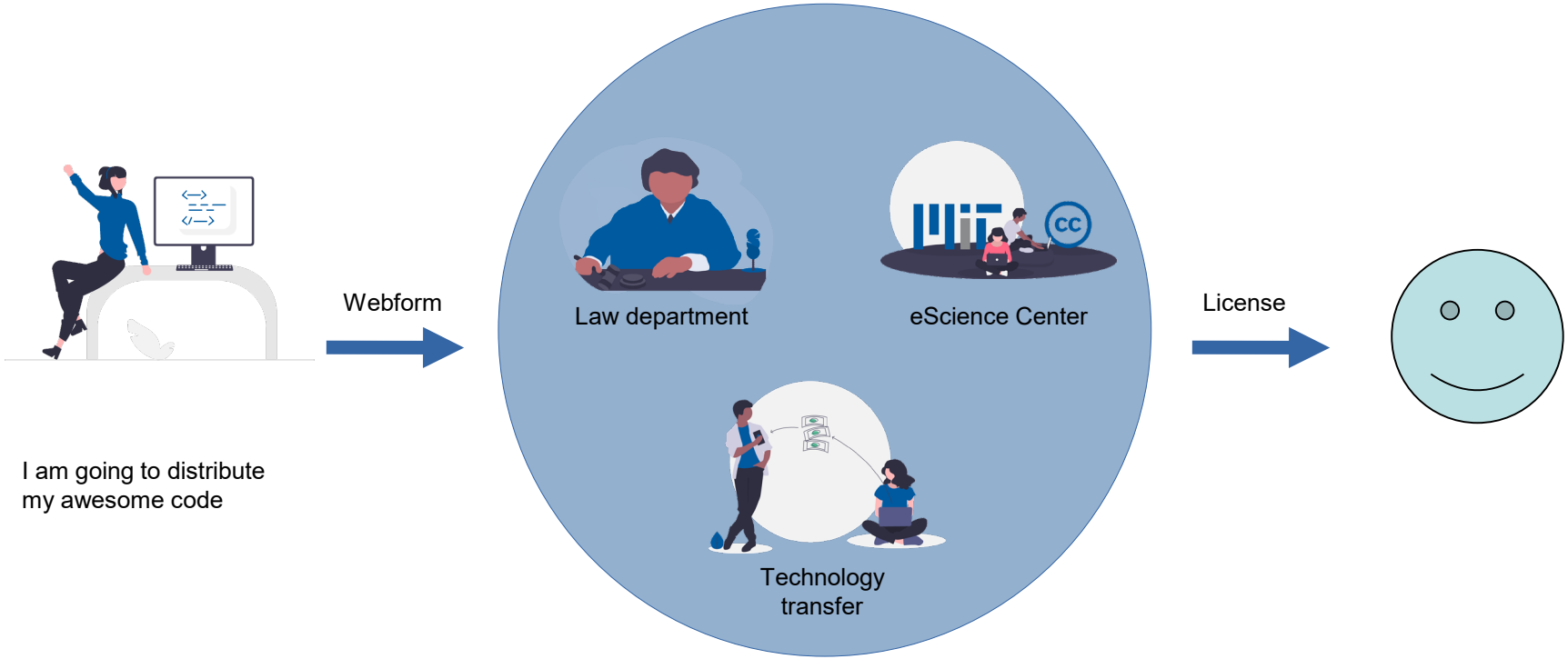
- Recommendation of best practices (further details/material in Intranet)
- Use of central infrastructure for version control, backup etc.
- Encourage application of community standard code styles, testing and continuous integration
- Documentation of SW
- REUSE-specification
- Recording of all developers involved, obtaining rights of use for further editing
- Off-boarding process

Update Dissemination Process

- Online-form for a review and approval of the planned distribution by all relevant stakeholders at GFZ, inter alia:
  - Check of license dependencies (eScience, Legal)
  - Consultation on possible exploitation (Transfer&Innovation TI)
  - Consultation on license recommendation (eScience, Legal, TI)

→ Providing orientation & workflow

The process in a nutshell



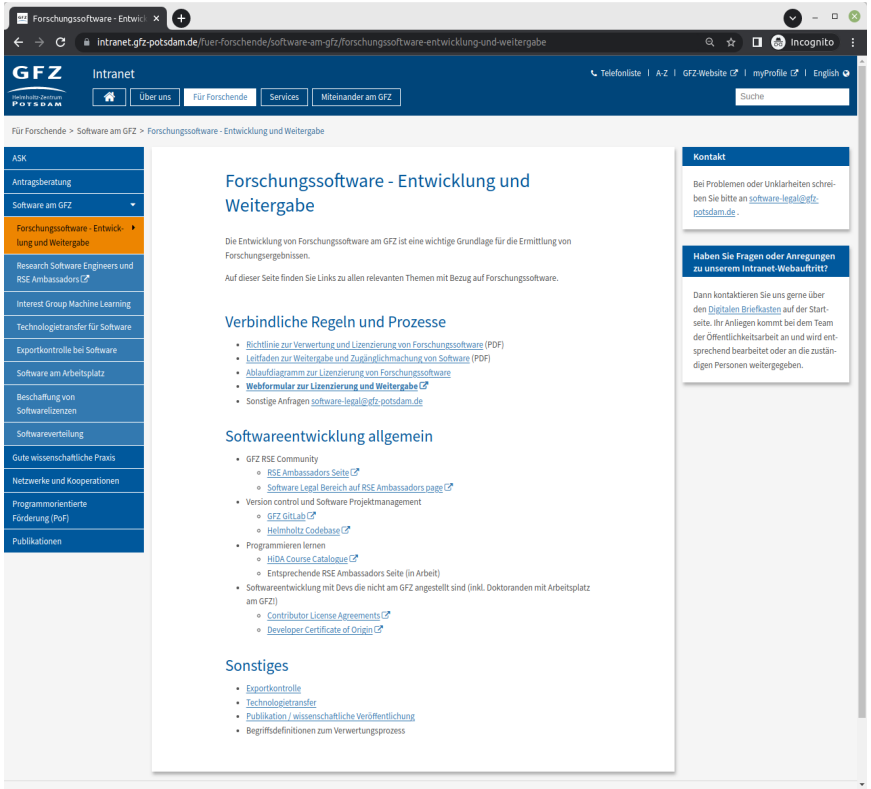
software-legal@gfz-potsdam.de

Intranet “Research Software at GFZ”

Explanations

Regulations

Webform



Templates

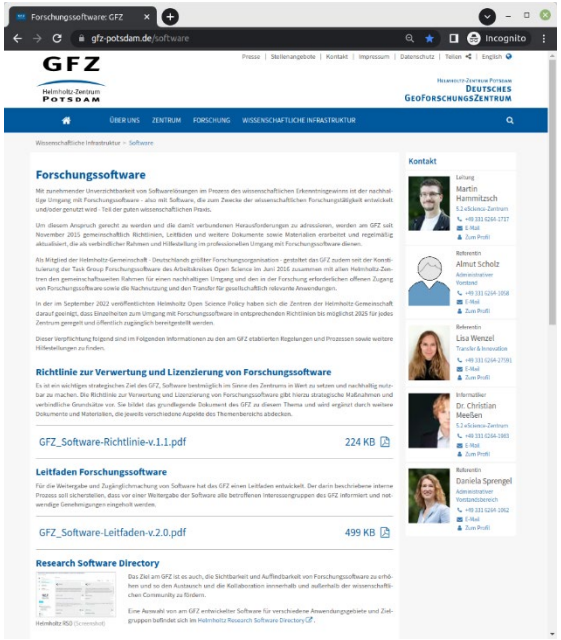
Recommended practices

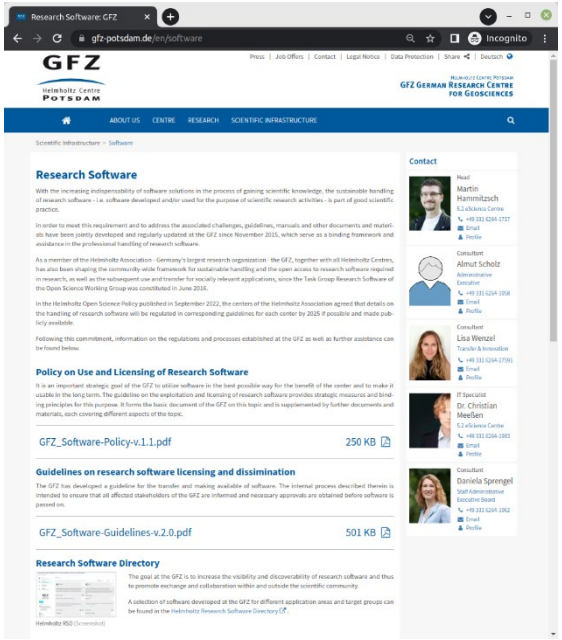
Contact

Challenges

- So far focus on process to enable at least a discussion about the possible exploitation of SW;
- Community-driven SW-developments follow rules of specific communities, are often based on trust and short-term ideas
- How to get employees to comply with the process?
- In 2022: 26 SW-license-requests via online-form; high dark figure
- Very time-intensive consultation process on a case-by-case-basis; clustering of SW-issues is challenging.

https://www.gfz-potsdam.de/en/software





How to contact us?

software-legal@gfz-potsdam.de

- Martin Hammitzsch, eScience Centre
- Dr. Christian Meeßen, eScience Centre/HIFIS
- Almut Scholz, Legal Department
- Daniela Sprengel, Legal Department
- Lisa Wenzel, Transfer&Innovation

Appendix 3 — HIFIS Software Services

(Uwe Konrad, HZDR)

HELMHOLTZ

RESEARCH FOR
GRAND CHALLENGES

HIFIS | HELMHOLTZ FEDERATED IT SERVICES

Software Services

Status, April 2023
Team Software Services

Software Services

Facts & Figures

~100
Training Workshops

~2.000
Attendees
17 Centers

HIFIS Team supports Research Software Engineering with

- **Education & Training**,
- **Community**,
- **Consulting** and
- **Technology** Services

>100
Software Spotlights

~70
Consultings Finished

~800
Peak number of monthly active GitLab projects

~4.000
Peak number of monthly active Helmholtz Codebase users
>10.000 registered users

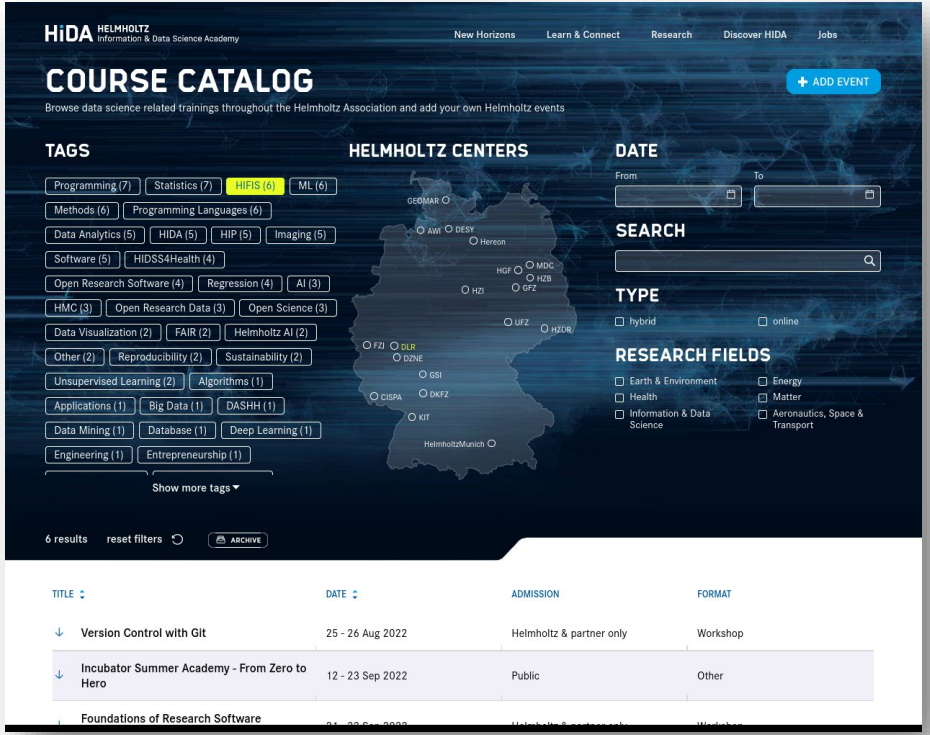
~250.000
GitLab CI Jobs Processed in 2022

Software Services

Education & Training

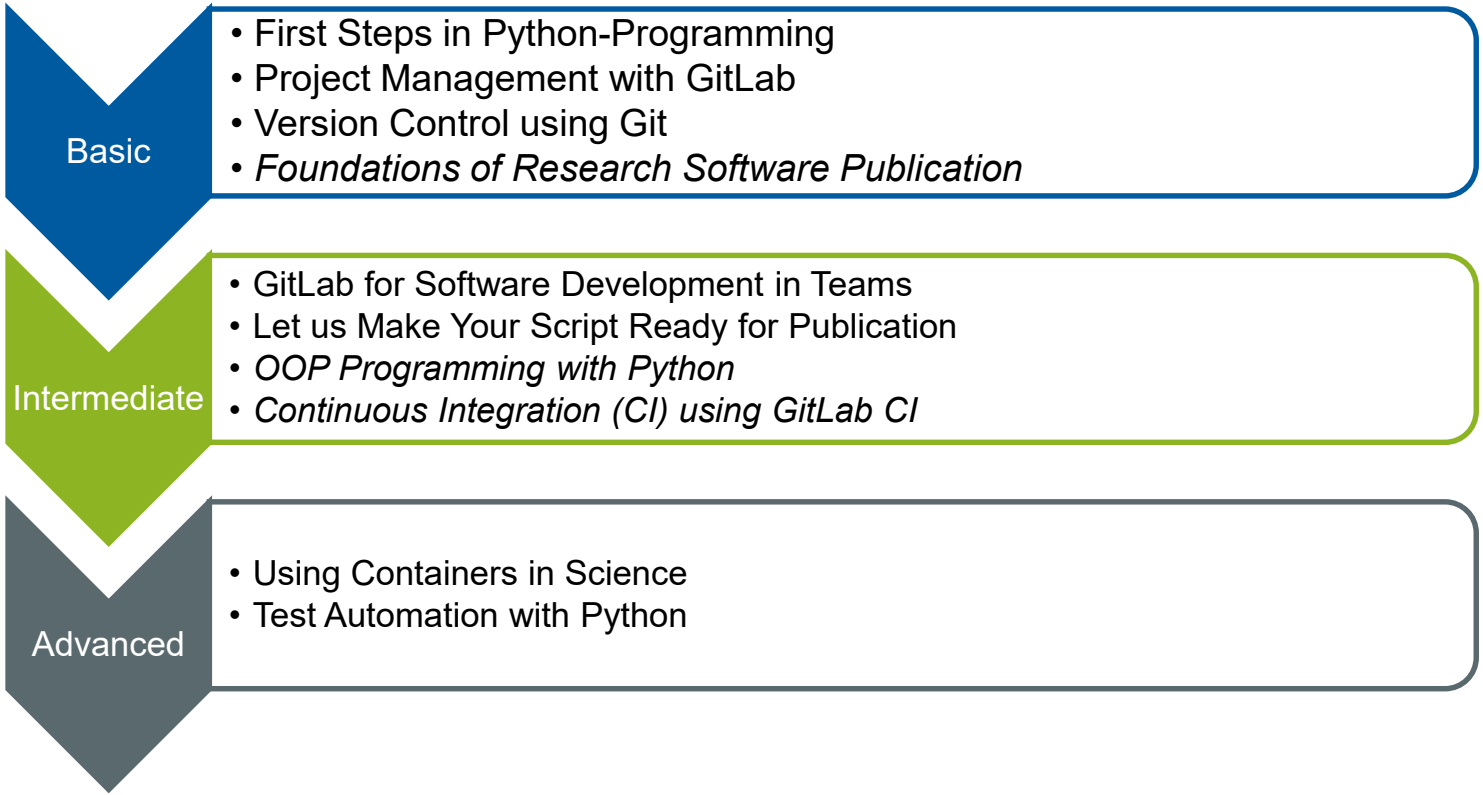
In collaboration with HIDA, HMC, Helmholtz Imaging and Helmholtz AI

- ***HIDA Course Catalogue*** - Central place to announce courses in Helmholtz
- *TEACH* **Conferences** – Talk about Education Across Communities in Helmholtz together
- Support **Hackathons** and other platform events



Software Services

Education & Training

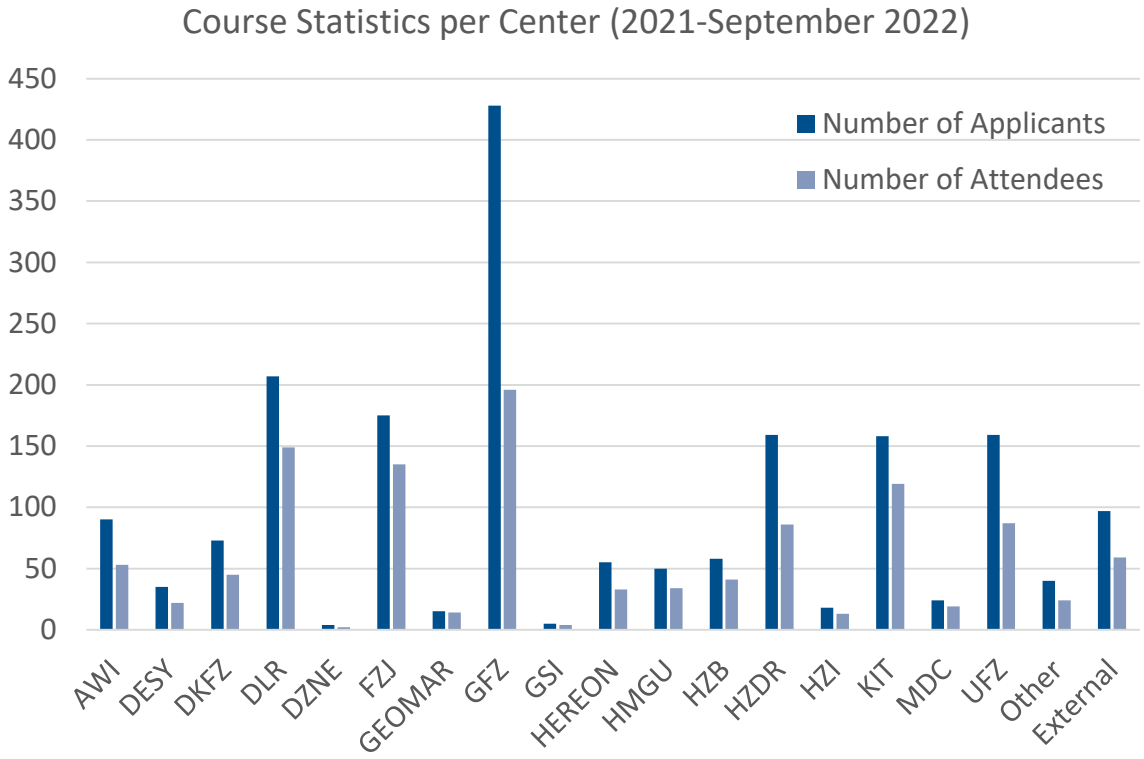


https://hifis.net/services/overall/learning-materials

Software Services

Education & Training

- **84** online workshops from 2020 until September 2022
- **1426** attendees, **867 hours** of instruction duration
- Learning material available via hifis.net*

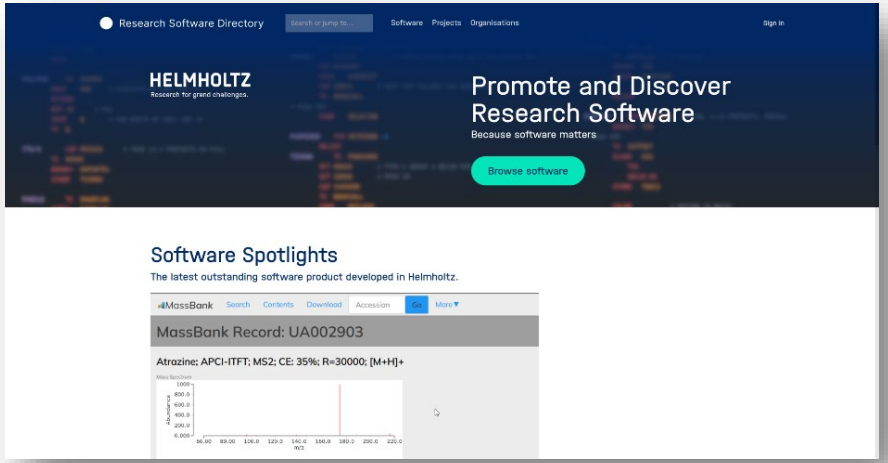


*https://hifis.net/services/overall/learning-materials

Software Services

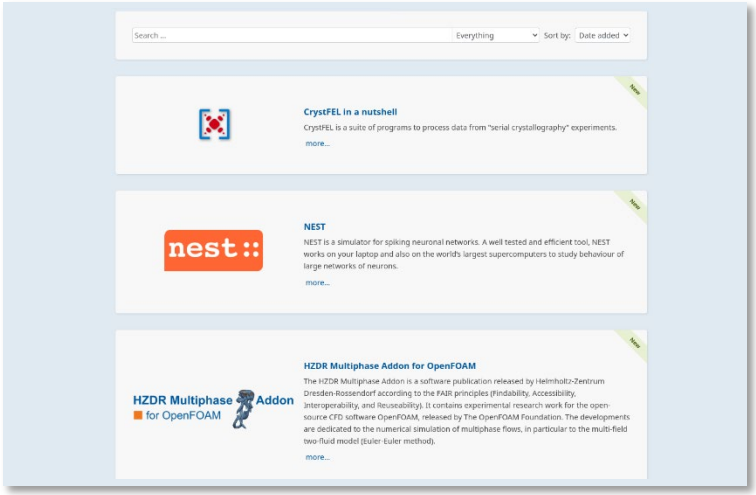
Community

Research Software Directory



- Research Software Directory for Helmholtz available https://helmholtz.software
- Built on top of and in cooperation with the Netherlands eScience center solution

Software Spotlights



- Present & Promote top success stories of Research Software Engineering in Helmholtz: https://hifis.net/spotlights
- approx. **100 Proposals**, thereof 31 being presented

Software Services

Community

- **Build recognition** and **Cross-linking** of communities
  - **Helmholtz Software Forum**
    - Joint exchange format driven by HIFIS and the Helmholtz Open Science Office
    - ~ 70-100 participants each time organized
    - Helmholtz SW award planned in 2023!
  - Contribute to **community activities** like deRSE
  - **Cross-linking** with similar activities of MPG, FhG, NFDI, EOSC ...

HELMHOLTZ Open Science

de RSE

nfdi
Nationale Forschungsdaten Infrastruktur

EUROPEAN OPEN SCIENCE CLOUD

Software Services

Consulting

- Free-of-charge **software consulting** for research groups within Helmholtz
- Possible topics include, but are not limited to **licensing and Open Source, setting up new projects**, code migrations etc.
- **Material** collection:
  - Consulting Handbook: https://hifis.net/consulting-handbook/
  - Awesome List RSE: https://github.com/hifis-net/awesome-rse

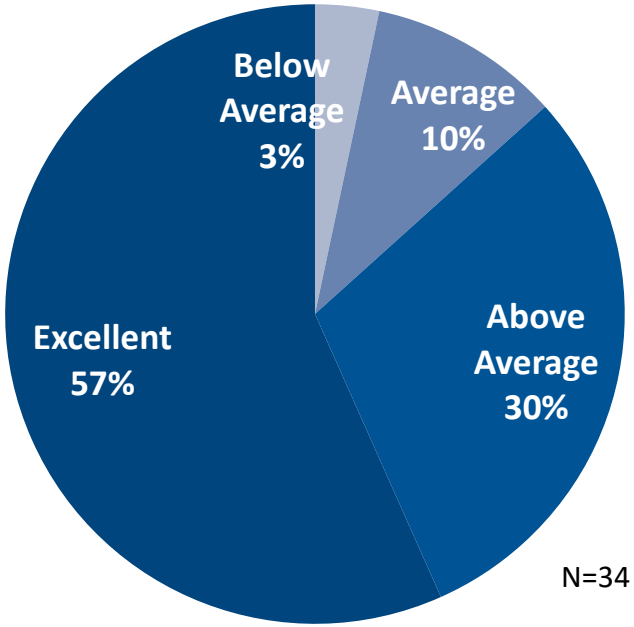
Software Services

Consulting

Great idea and a great support especially since there is no other person programming in my research group. Very happy that you came up with this!

For us, it would be perfect to have such a consulting service over a longer period of time, e.g. for 6-12 months with regular meetings.

Impact of the consultation on your project or work



N=34

Software Services

Technology

Supporting the whole software development lifecycle

Dependabot

Security: Automate dependency updates

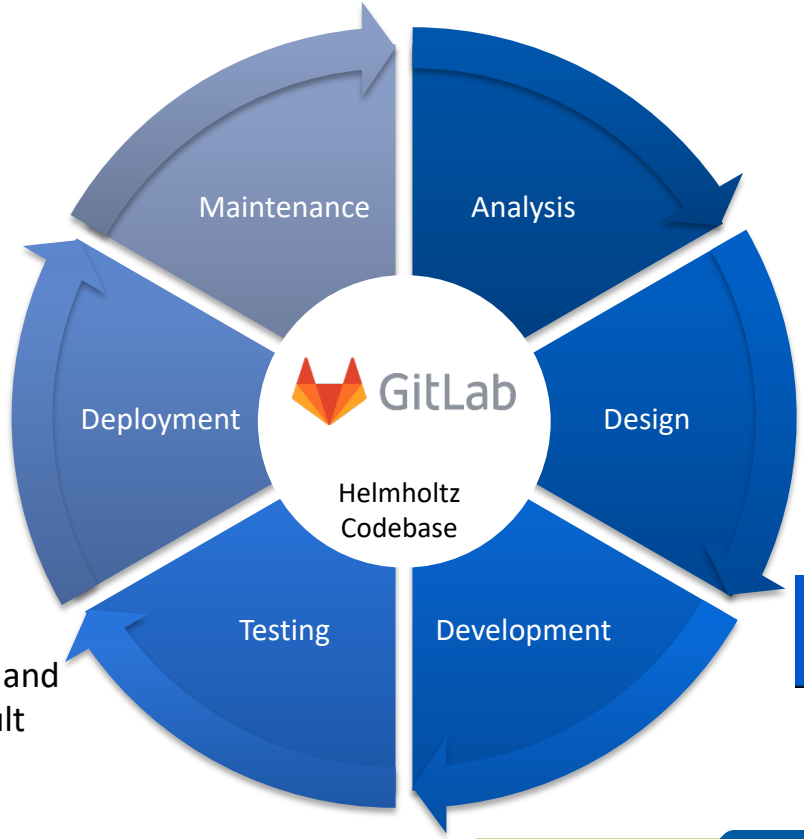
Deployed and made available with

ANSIBLE

https://github.com/hifis-net

GitLab CI

Continuous Integration and Deployment by default



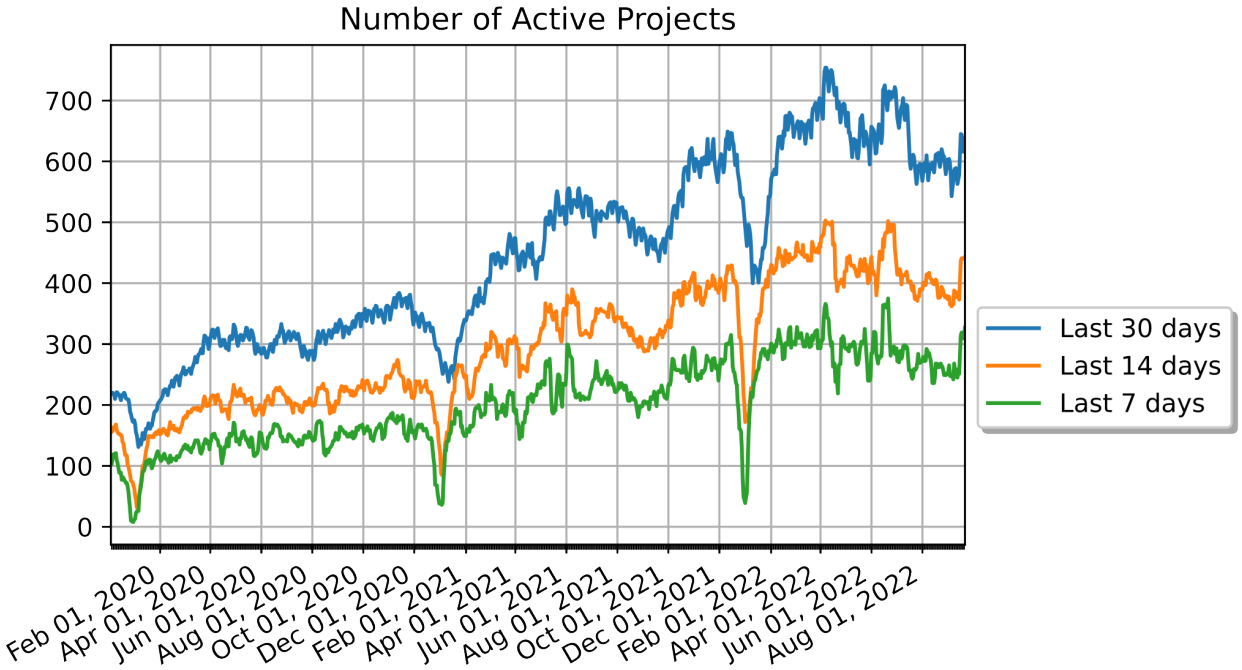
Kroki

Diagram creation

Mattermost

Team communication

Technology

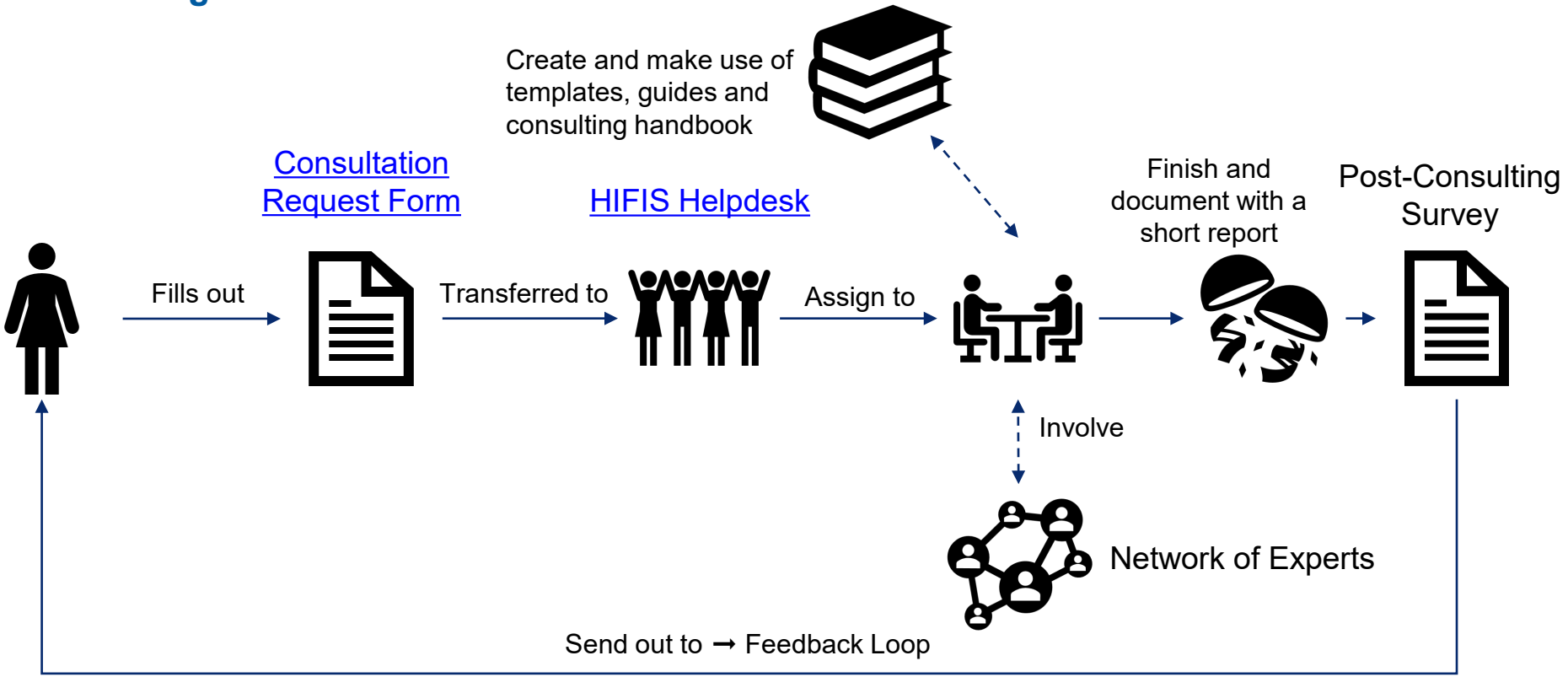


Helmholtz Codebase - Usage Statistics

GitLab

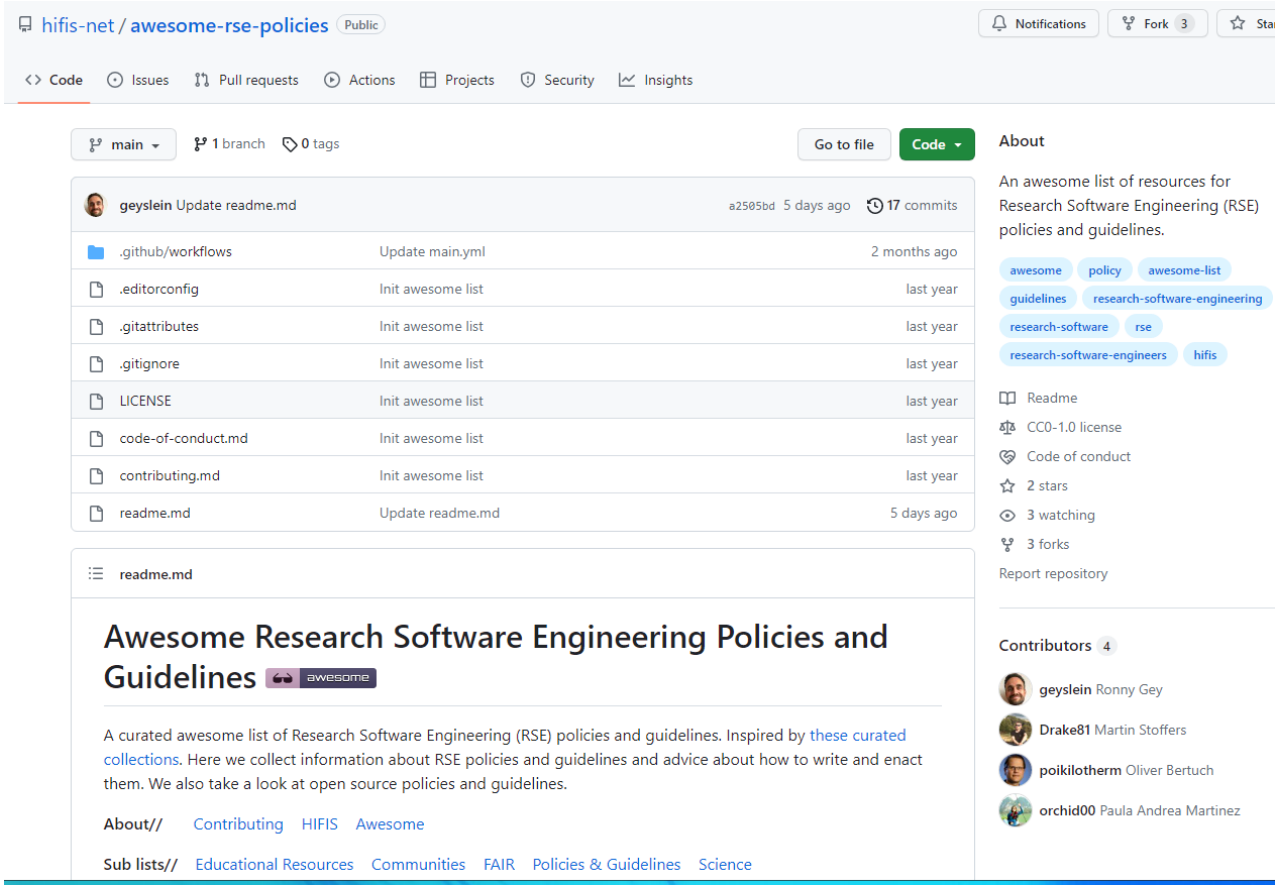
Software Services

Consulting



Software Services

Consulting



HIFIS Cloud Services

www.hifis.net— Fact Check

~4.000
users from non-Helmholtz

>17.000
individual users in Helmholtz Cloud

All 18 Centres
One Login!

31
Helmholtz Cloud Services and Research Pipelines

HIFIS | HELMHOLTZ FEDERATED IT SERVICES

>940
processed support tickets in 2022
(+190 in 2023 so far)

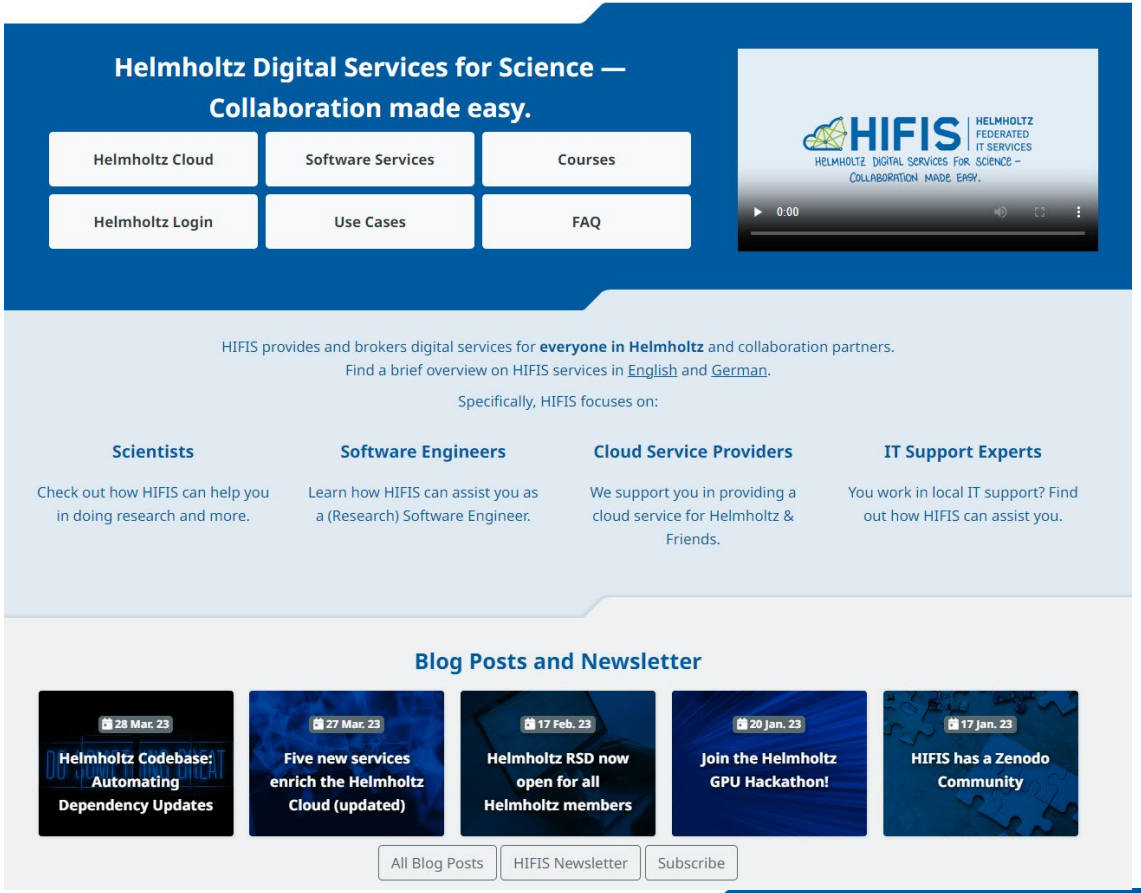
>150%
average increased usage of Helmholtz Cloud Services since onboarding

>100
collaborating groups (VO)

>70
workshops

Thanks.

visit us at
https://hifis.net



HELMHOLTZ

Open Science

Appendix 4 — Helmholtz Platform for Research Software Engineering - Preparatory Study (HiRSE_PS)

(Johannes Reuther, HZB)

HELMHOLTZ

SPITZENFORSCHUNG FÜR
GROSSE HERAUSFORDERUNGEN

Helmholtz Platform for Research Software Engineering – Preparatory Study (HiRSE_PS)

Johannes Reuther

Helmholtz-Zentrum Berlin

HIRSE_PS

Together with Markus Diesmann, Stefan Blügel, Robert Speck (Jülich), Achim Streit, René Caspart, Markus Götz (KIT), Christian Cyron, Regine Willumeit-Römer & Daniel Höche (Hereon)

JÜLICH
Forschungszentrum

KIT
Karlsruher Institut für Technologie

HZB Helmholtz Zentrum Berlin

Helmholtz-Zentrum
hereon

Research Software Engineering (RSE) – Why?

HIRSE_PS

(Open Source) Software =

- Key component of scientific work
- Software ≈ data ≈ devices
- Software = research infrastructure
- Valuable assets

in all research fields of Helmholtz

Sources: https://www.software.ac.uk/about, https://www.helmholtz.de/en/research/

Innovation pool project HiRSE_PS

https://www.helmholtz-hirse.de

HIRSE_PS

- **HiRSE_PS** can only provide a first impetus of a RF-wide, much larger activity
  - Testbed for structural RSE support within a research field (toward HiRSE)
  - Focus on software as an infrastructure (open, reliable, sustainable, reproduceable)
  - Human-centric view: enable RSEs to work best and together on their codes/project
- **3 pillars** in 2 work packages
  - Specific **Community Software Infrastructure (CSI) groups** for already existing/established Community Codes Teams
    - NEST and FLEUR (FZJ), HeAT (FZJ/KIT/DLR/Intel), PFFRG (HZB), 4C (Hereon)
  - Central **support and consulting unit** for RSE (FZJ, KIT)
  - Open **HiRSE Seminar** for a regular exchange of information

WP 1: CSI groups (PI: Markus Diesmann)

Community Software Infrastructure Groups

HIRSE_PS

- **Goals of the WP**
  - Establish five CSI groups (topical width across all 3 programs in the RF-Information and diversity of already established and young codes)
  - Fostering sustainability and long-term stability of specific codes
  - Supporting the community
  - Generates insights about the requirements for WP2
- **Structure and location of CSI-groups**
  - Expert knowledge from a scientific domain
  - High community trust through in-person-responsibilities in the domain-institute
- **Tasks of the CSI-groups**
  - Coordination of the development, e.g. doing code reviews, generating releases, monitoring of Cx-technologies
  - Taking over hard and longer-lasting development tasks (e.g. refactoring of existing codes)
  - Organization of trainings and Hackathons

WP 2: Consulting & Networking (PI: Achim Streit)

Goals and structure

HIRSE_PS

- **Goals:**
  - Establishing the technological basis for RSE
  - Supporting established CSI groups and codes in Cx environment usage and software engineering
  - Taking young codes by the hand introducing modern RSE practices
- **3 sub-WPs**
  1. Framework for CI/CT/CD
  2. Support & consulting
  3. Community building & networking

- **Usage of modern Supercomputing infrastructures**
  - E.g., JUWELS (FZJ-JSC) or HoreKa (KIT-SCC)
  - Future Technologies Partition at KIT-SCC
    for CI on different HPC resources and architectures
  - Cloud resources via OpenStack in the Helmholtz Data Federation

openstack.

First activities and results

HIRSE_PS

- National **HiRSE Seminar Series** established (currently 17 talks, ~20-70 attendees per talk)
- Outreach and networking: regular meetings with HIFIS, joint development of the **Helmholtz Incubator Software Award,** co-chairing deRSE23 and un-deRSE23
- **HiRSE_PS Hackathon:** Sept. 2022 on Continuous Integration, Sept./Oct. 2023 on Continuous Benchmarking

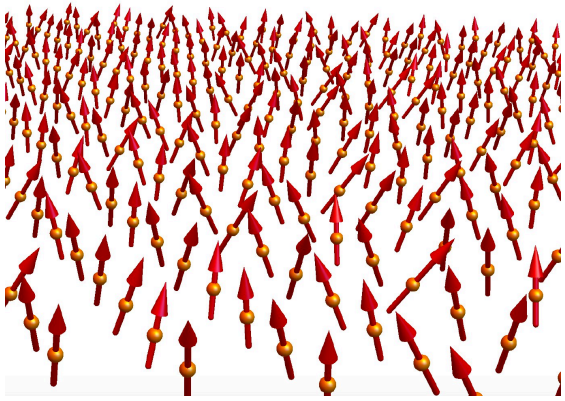
de RSE CONFERENCE 2023

- **Cross-fertilization** of CSI groups, between young and established codes
- **Software sustainability**: enabling/extending usage of Cx, code refactoring, docs-as-code
- Energy consumption as **software quality metric**, ongoing discussions on KPIs
- Recruited **RSE Community Manager** (Claire Wyatt) to support local, RF, and HGF RSE communities

Background of PFFRG

HIRSE_PS

- **PFFRG** is a software for simulating quantum magnetic materials **including the accurate description of quantum fluctuations**.
- First code developed approx. 15 years ago.
- **Publications in prestigious journals** (Nature Physics, Nat. Comm., PRL, PRX,…)
- Increasing software problems:
  - Bad documentation
  - Not ready for HPC computing (bad parallelization)
  - No unit tests
  - Only private code (despite interested community)
- Strong embedding in science community but **isolation in terms of RSE topics!**
- **New code in Julia!**



julia

PFFRG at HiRSE_PS

HIRSE_PS

- **Participation in joint meetings/workshops**
- **Being actively approached by software experts within HiRSE_PS** (WP 2: Consulting & Networking)
- **Additional funding!**
- **Integration into an RSE community, getting out of RSE isolation, community building important for HiRSE_PS!**
- **Large improvement of code within relatively short time** (Code in public repository, more contributors, improved documentation, new unit tests, new scientific features,…)

Nils Niggemann

My case is not uncommon!

From ``Excellent scientist doing good science with bad codes'' to``Excellent scientists doing excellent science with excellent codes''!

Summary and Outlook

HIRSE_PS

- **Extension of HiRSE Seminar Series**
  - special online events, e.g. HiRSE Summer of Testing
  - HiRSE Tutorials: short (2-3h) tutorials on a specific topic
- **More collaboration and outreach**
  - GI RSE interest group
  - close-by universities and local partners
- **Integration of more groups**
  - Simulation and Data Labs (SDLs) within the RF Information
  - Support for individual RSEs within the centers
- **More visibility of RSE and RSEs within RF and HGF**

JÜLICH Forschungszentrum

KIT Karlsruher Institut für Technologie

HZB Helmholtz Zentrum Berlin

Helmholtz-Zentrum hereon

HELMHOLTZ

Open Science

Appendix 5 — Increasing the visibility of Research Software: The Helmholtz Research Software Directory

(Christian Meeßen, GFZ)

HIFIS HELMHOLTZ FEDERATED IT SERVICES

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

Increasing the visibility of Research Software: The Helmholtz Research Software Directory

Christian Meeßen[1]

Helmholtz Open Science Forum Research Software Policies

23. April 2023

Contributions
Felix Mühlbauer, Tobias Huste, Norman Ziegner, Martin Hammitzsch, Uwe Konrad

1) Helmholtz Zentrum Potsdam, Deutsches GeoForschungsZentrum Potsdam GFZ

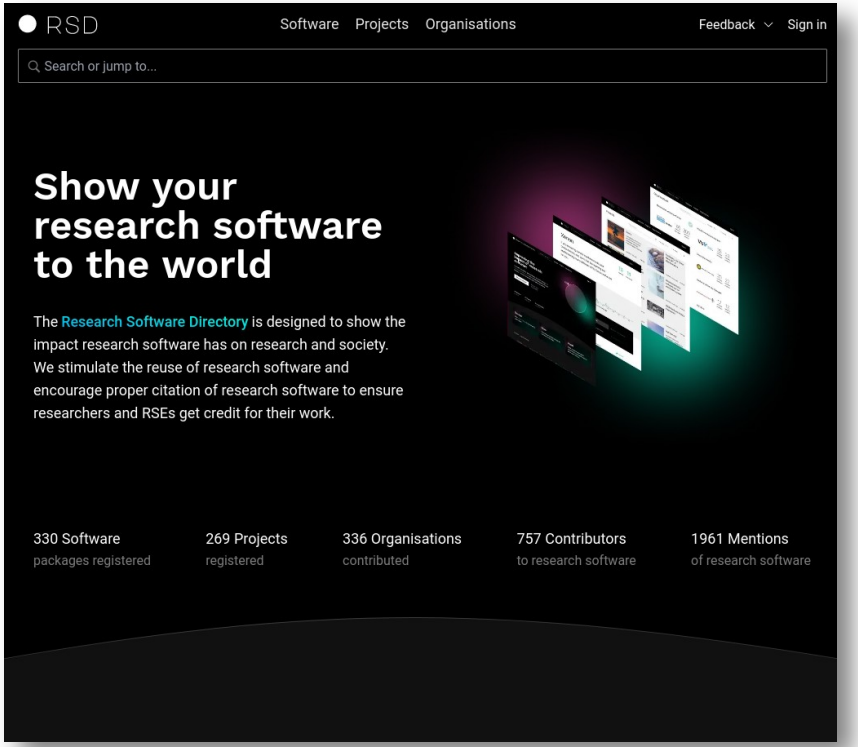
Towards more visibility

First step: The Helmholtz Software Spotlights

- Aim: increase awareness about importance and visibility of Research Software
- Centres suggested 100 candidates
- Helmholtz Software Spotlights on https://hifis.net/spotlights
- 30 online

Increasing the visibility of Research Software

Next step: The Research Software Directory

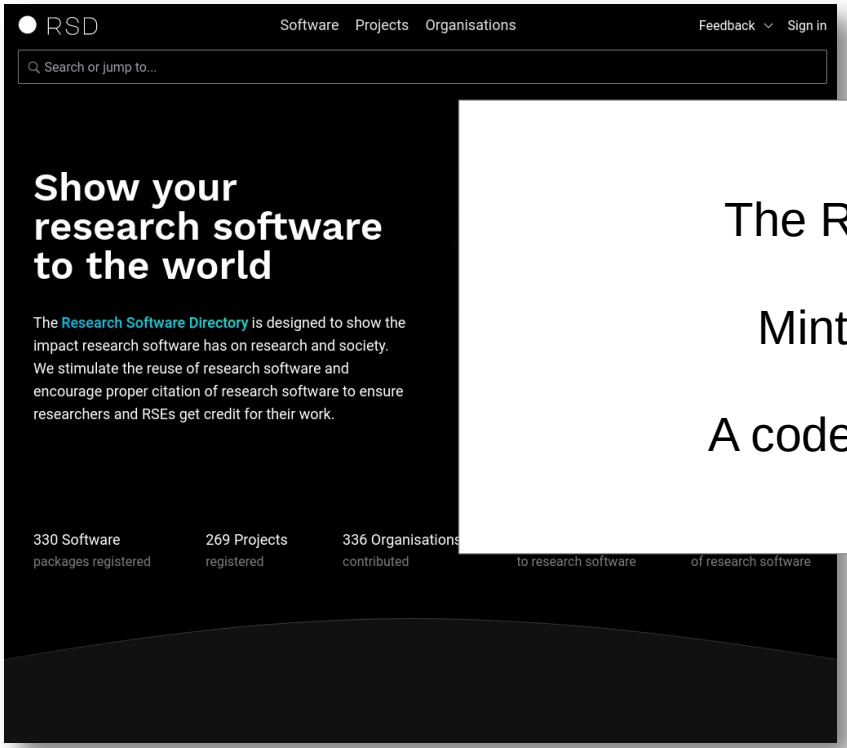


https://research-software-directory.org

- Software Catalog for RSEs and Engineers
- Online service designed to collect and present software metadata
- Enables **RSEs** to show the impact their software has
- Helps **researchers** tofind softwarey they need
- Encourages **citation** of research software in other research outputs
- Allows **organisations** to collect information about their software output
- FOSS and licensed Apache-2.0
- Supported by HIFIS Software Community
- Completely new development since 2022

Increasing the visibility of Research Software

Next step: The Research Software Directory



https://research-software-directory.org

The RSD is not:

Minting PIDs

A code repository

- Software Catalog for RSEs and Engineers
- Online service designed to collect and present software

ow the impact their software has

ofind softwarey they need

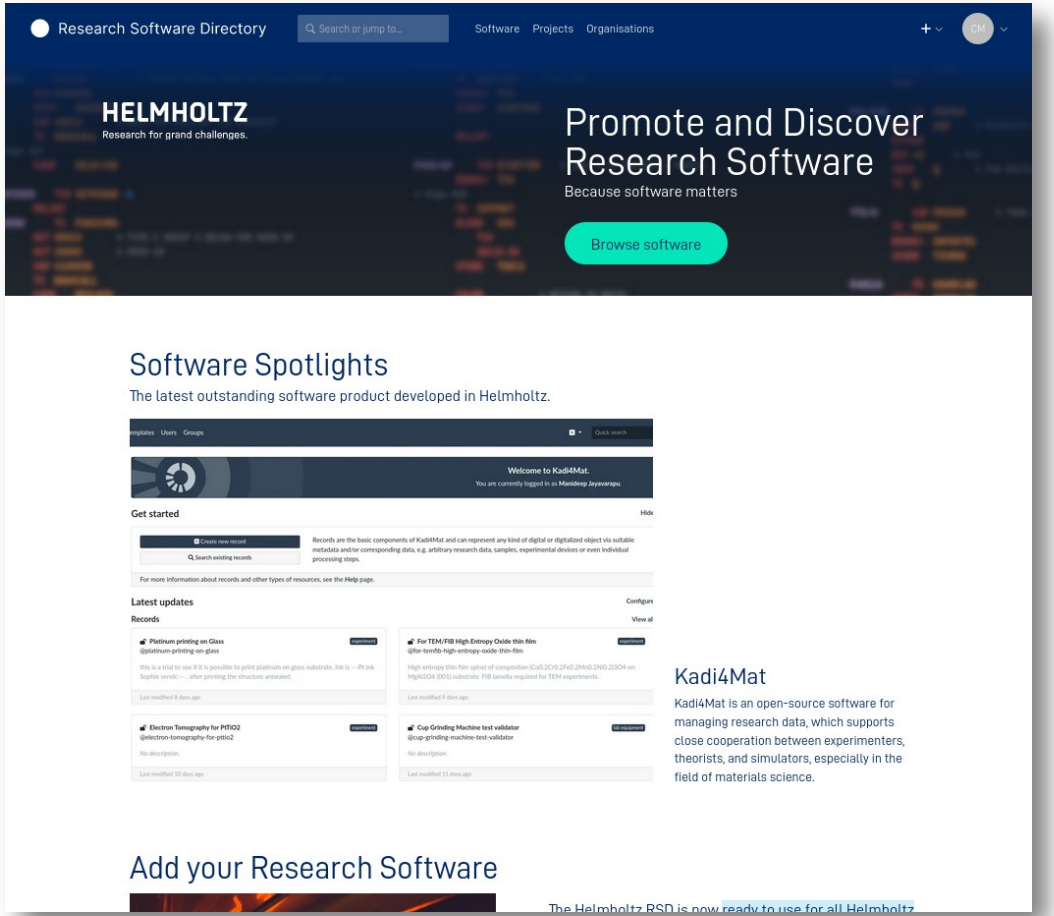
of research software in other

s to collect information about their

Apache-2.0

- Supported by HIFIS Software Community
- Completely new development since 2022

The Helmholtz Research Software Directory



- Pilot: July 2022
- Official launch: February 2023
- Forked
- FOSS and EUPL-1.2
- Helmholtz-specific
  - Software Spotlights (WIP)
  - Research Fields (WIP)
  - Connected to Helmholtz AAI
  - Hosting Helmholtz Software Award
- Helmholtz Cloud Service
- https://helmholtz.software

Data sources

Authentication

HELMHOLTZ AAI

Software metadata

GitLab

GitHub

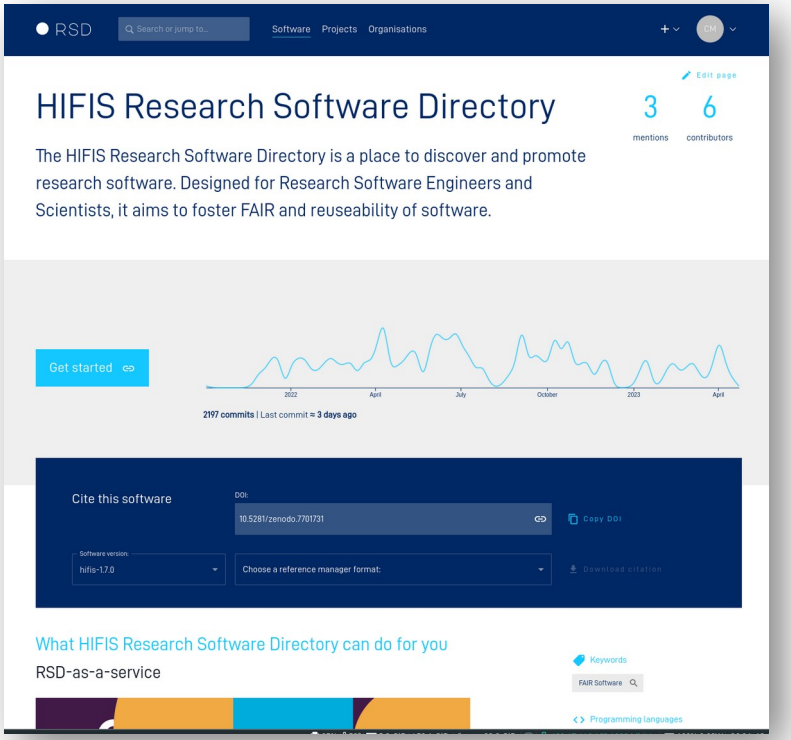
Maven

python Package Index

ANACONDA

docker hub

npm



Authentication

ROR

Contributors

ORCID

References

Crossref

DataCite

FIND, ACCESS, AND REUSE DATA

The Helmholtz RSD in numbers

13
Helmholtz centres with software contributions

35
Partner organisations

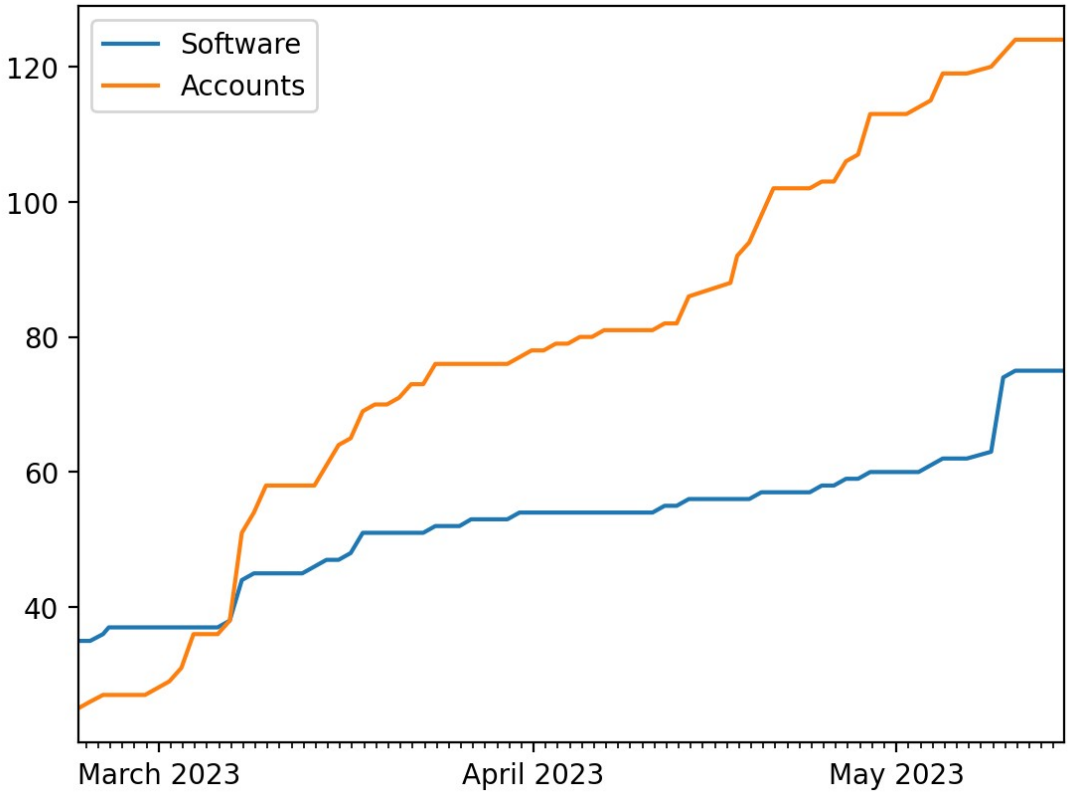
216
Software contributors

75
Software entries

404
Software mentions

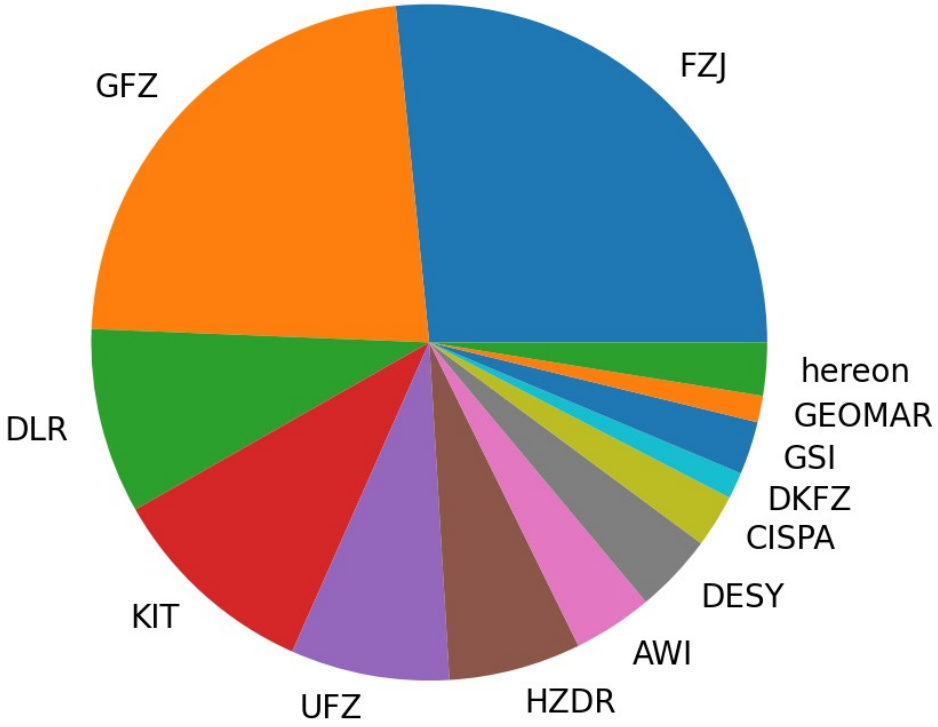
Data as of 15. May 2023

Evolution of software entries and accounts since launch



Data as of 15. May 2023

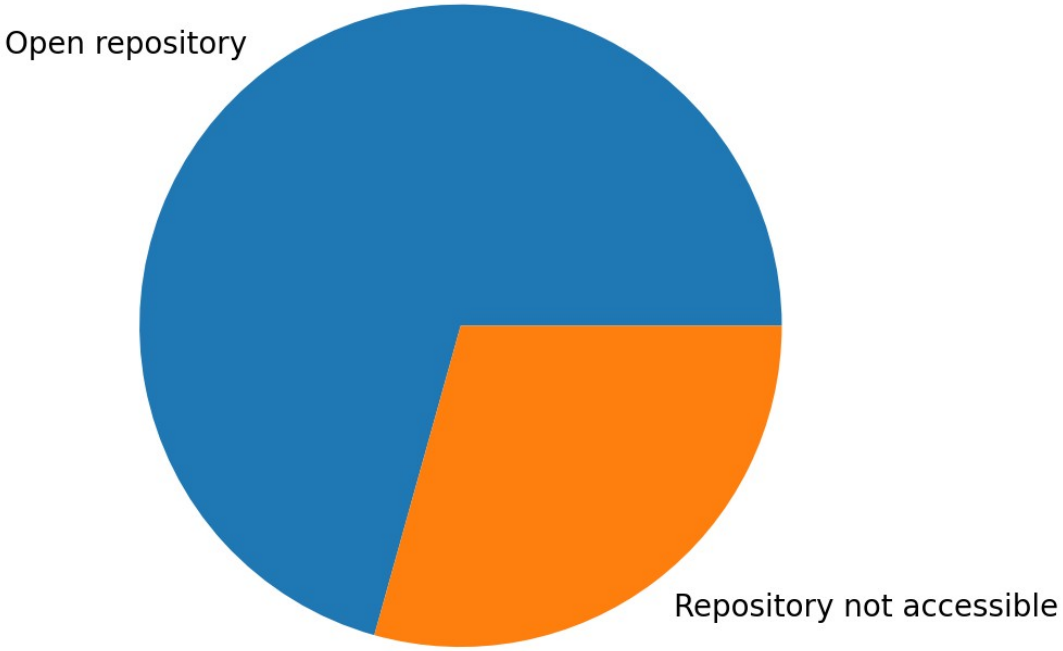
Software count by research centre



Total: 75

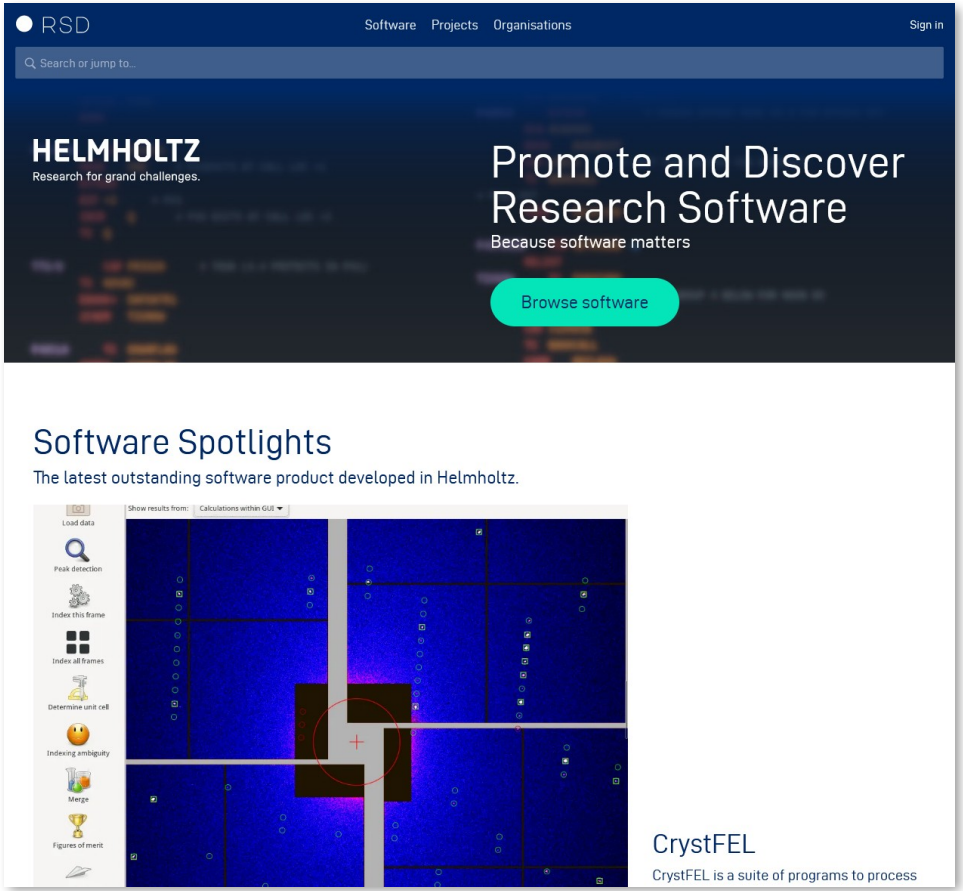
Data as of 15. May 2023

Repository accessibility of available software entries

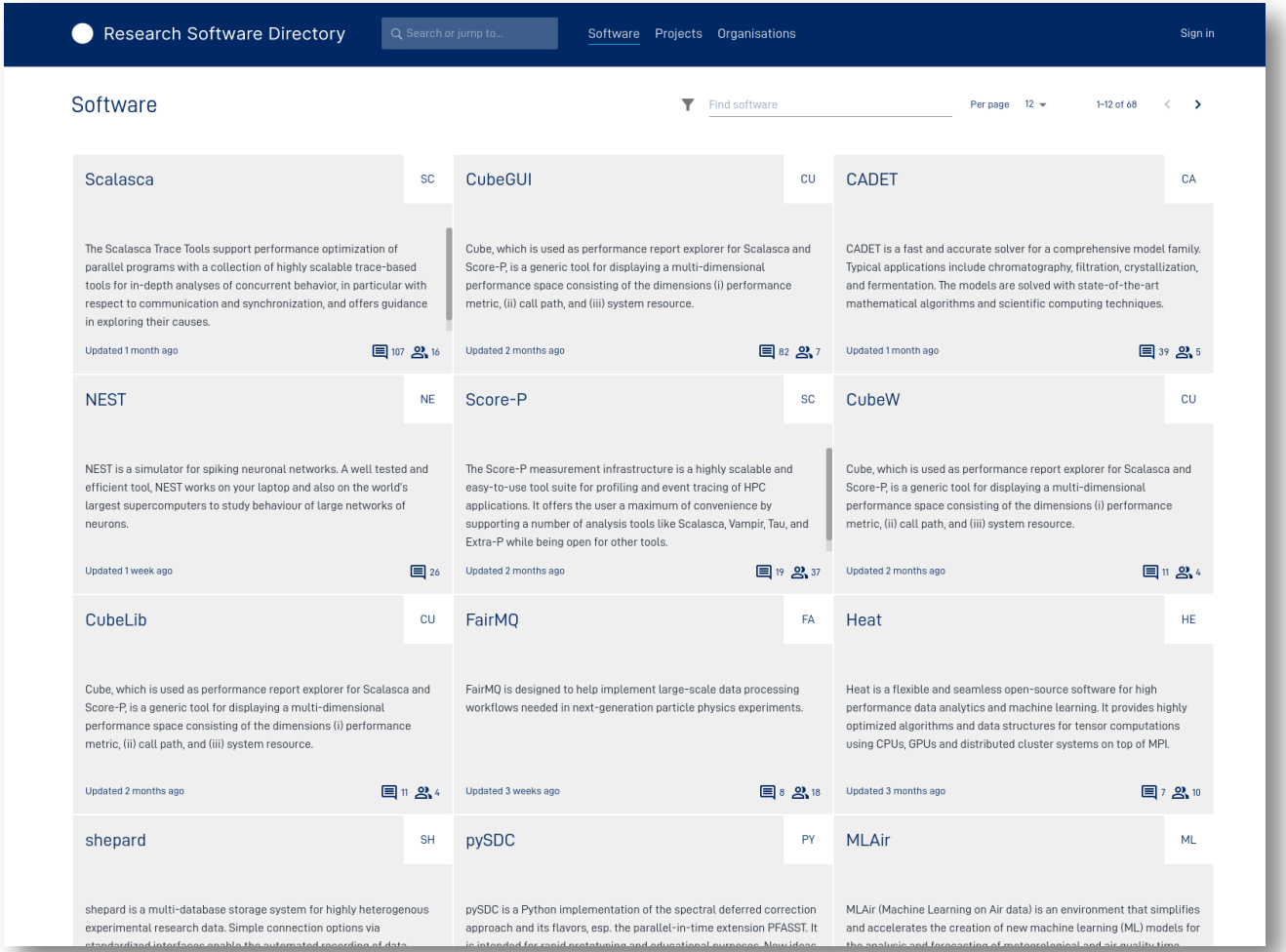


Data as of 15. May 2023

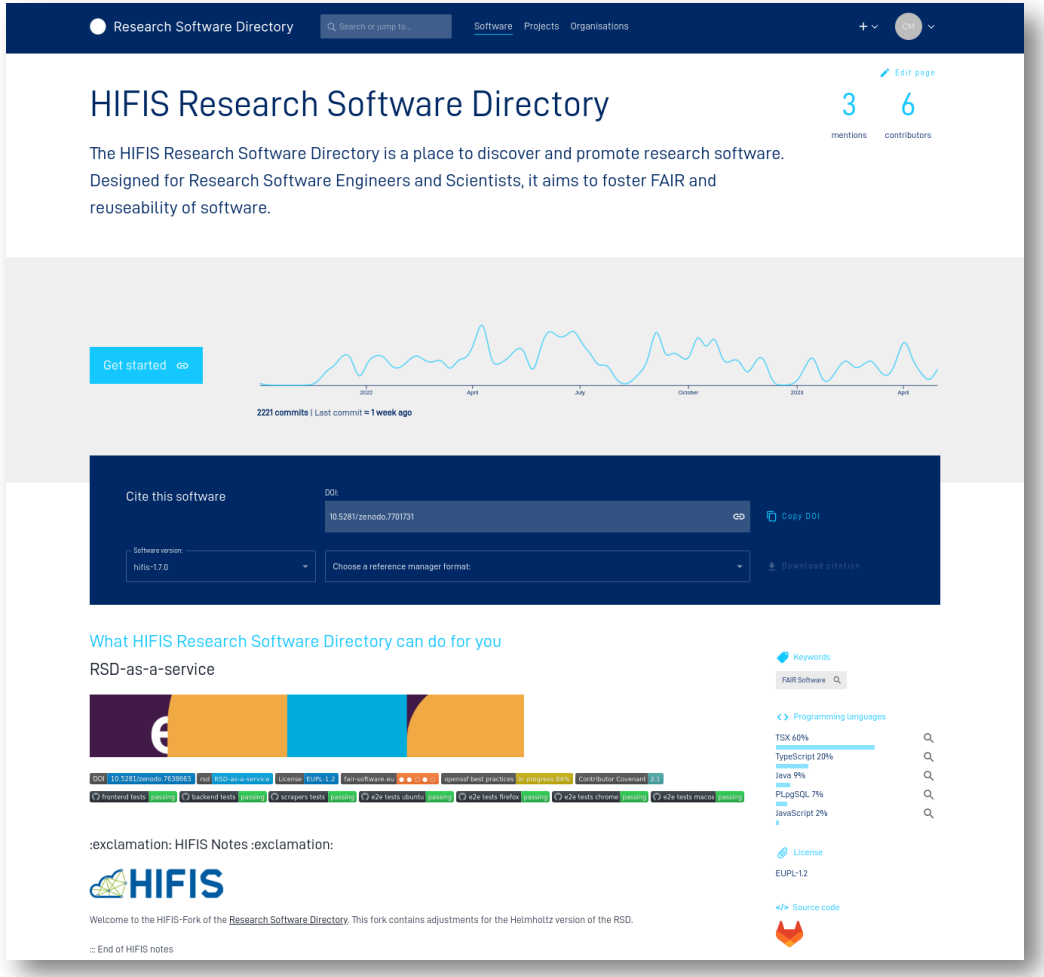
Live Demo


RSD
Software
Projects
Organisations
Sign in
Search or jump to...
HELMHOLTZ
Research for grand challenges.
Promote and Discover Research Software
Because software matters
Browse software
Software Spotlights
The latest outstanding software product developed in Helmholtz.
Load data
Peak detection
Index this frame
Index all frames
Determine unit cell
Indexing ambiguity
Merge
Figures of merit
CrystFEL
CrystFEL is a suite of programs to process

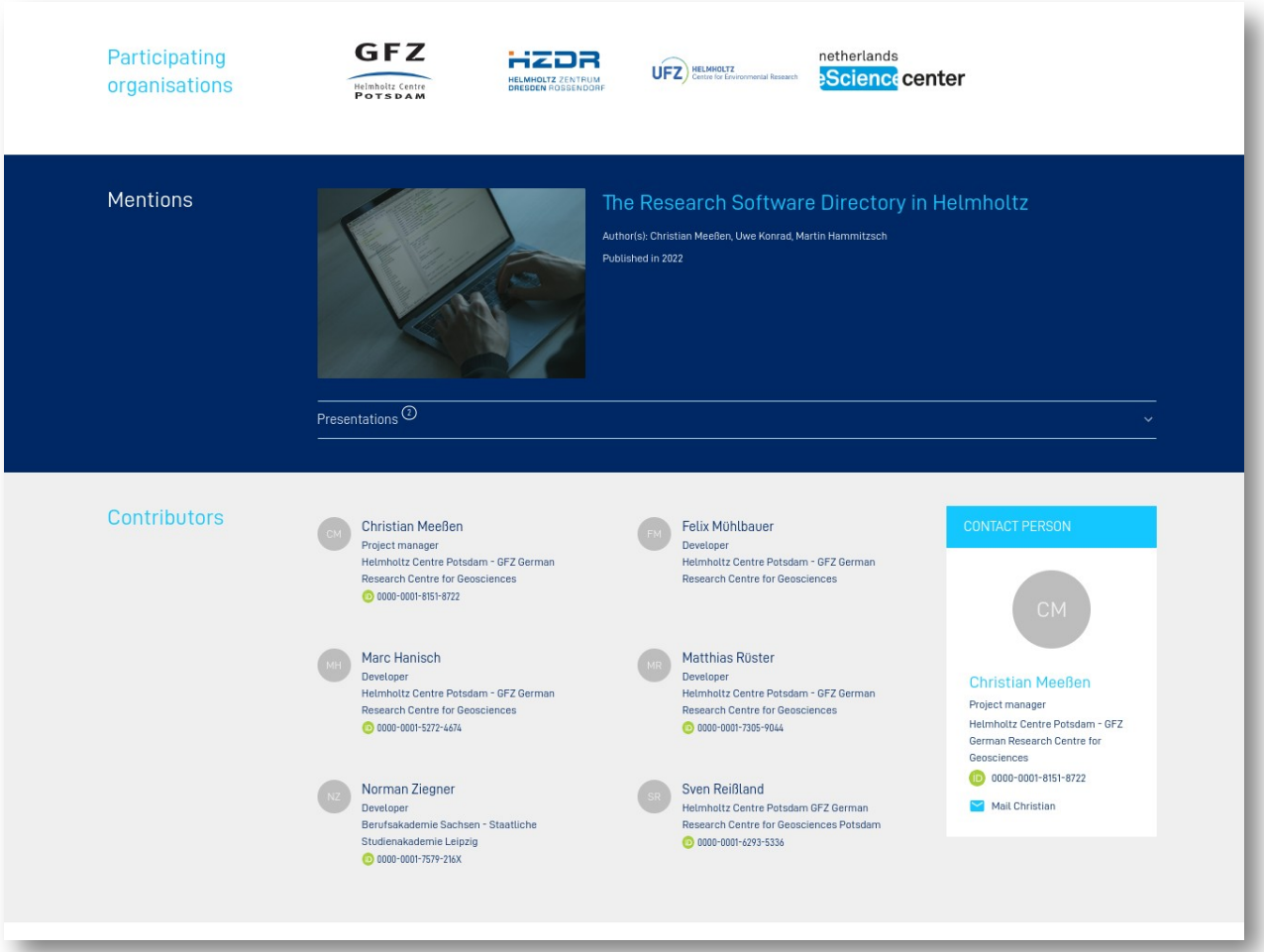
Software overview



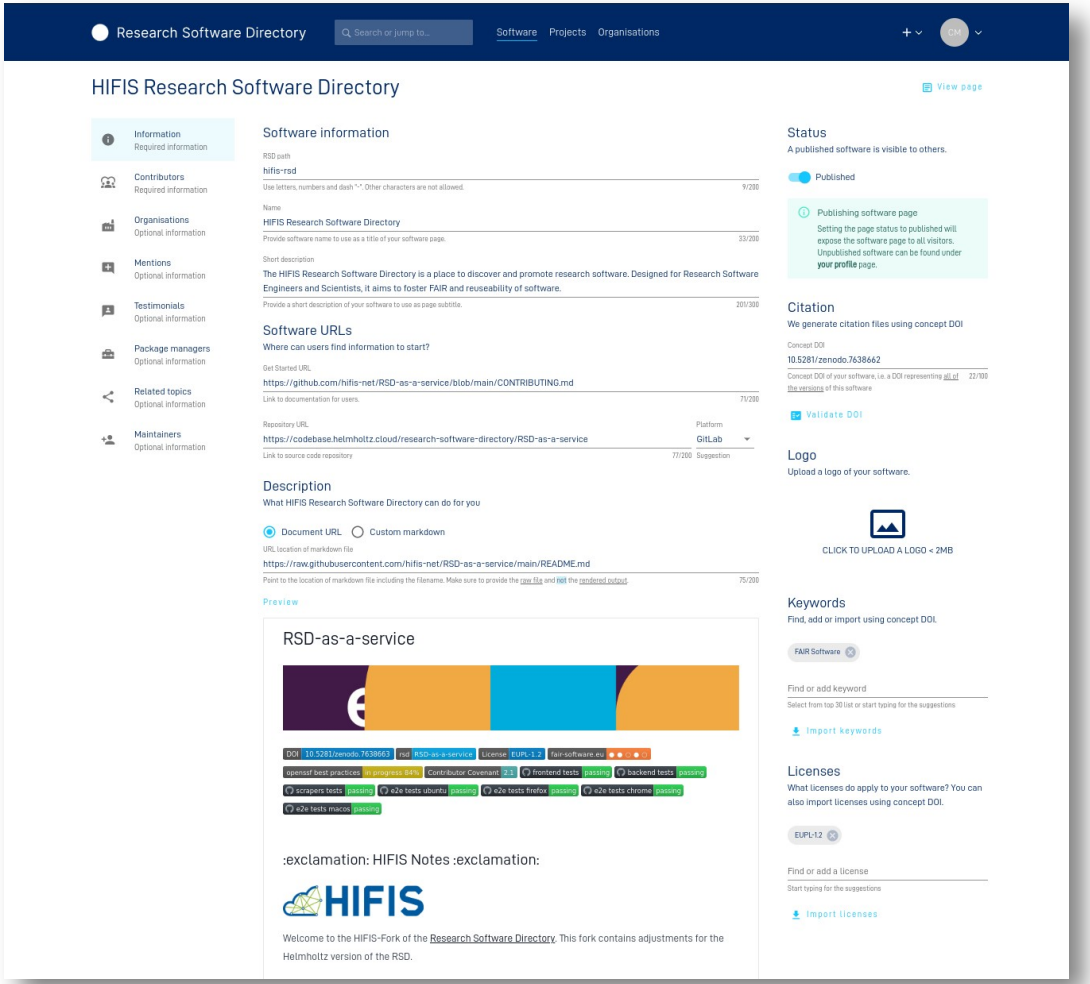
Software entry



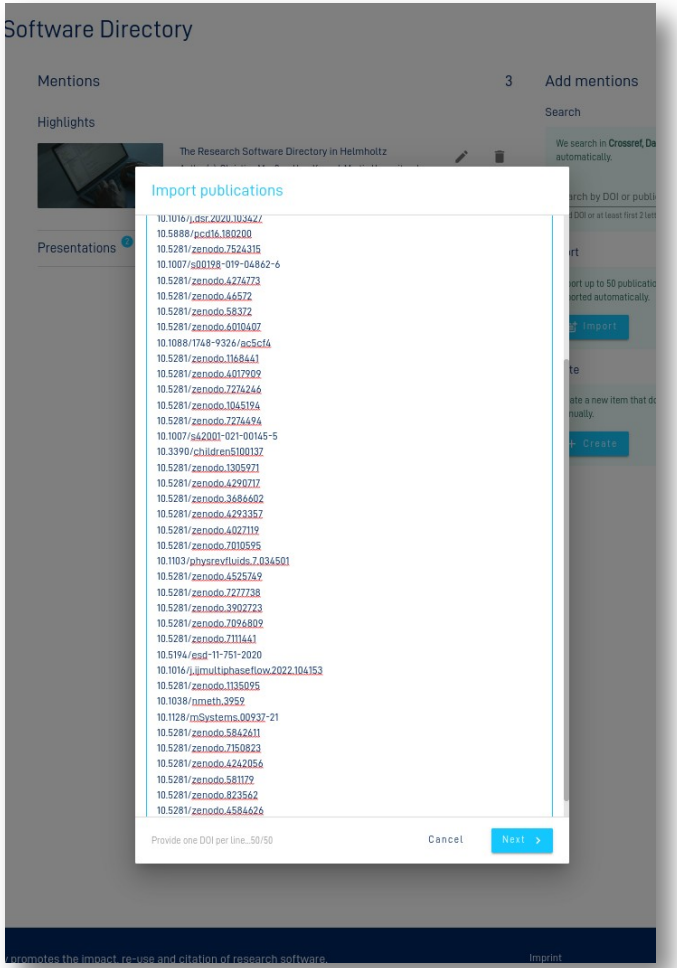
Software entry

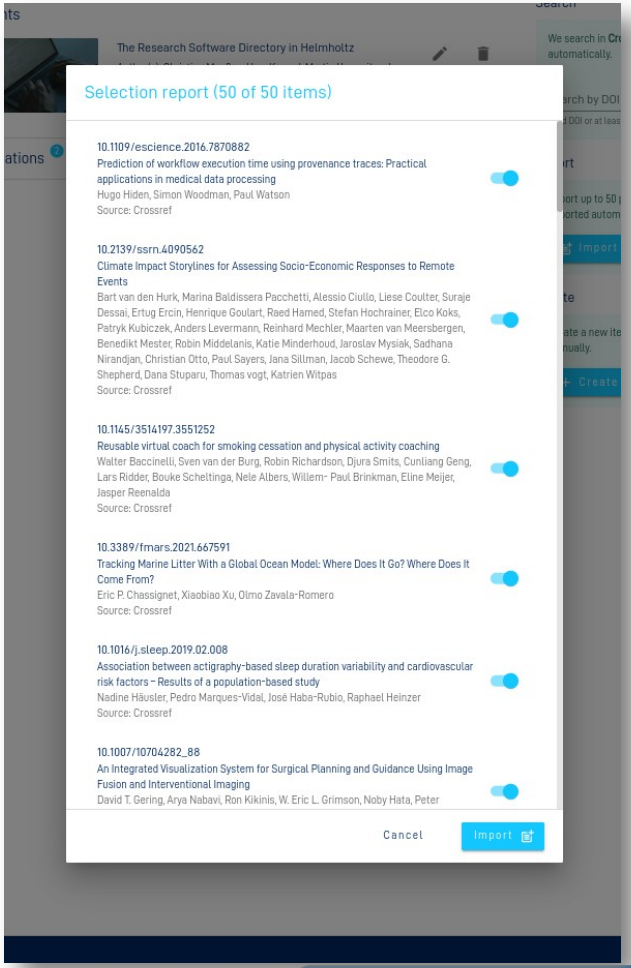


Software edit page

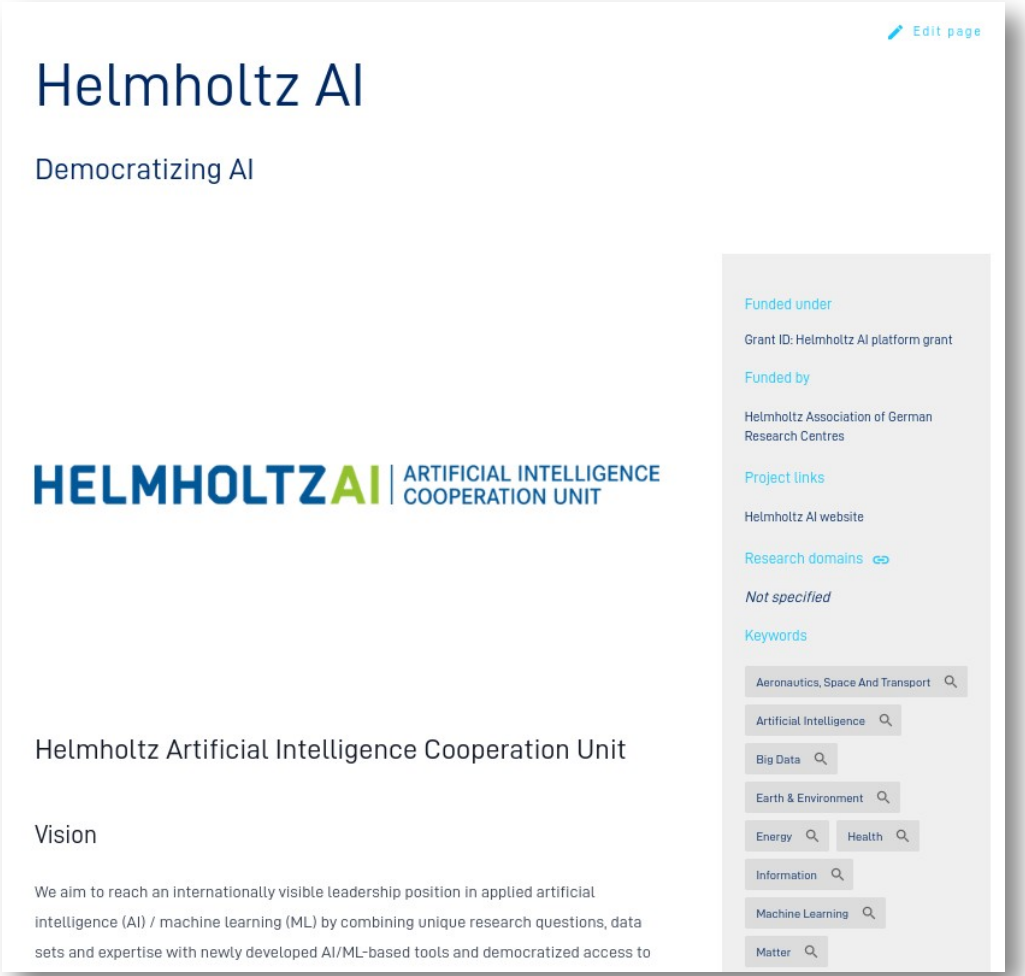


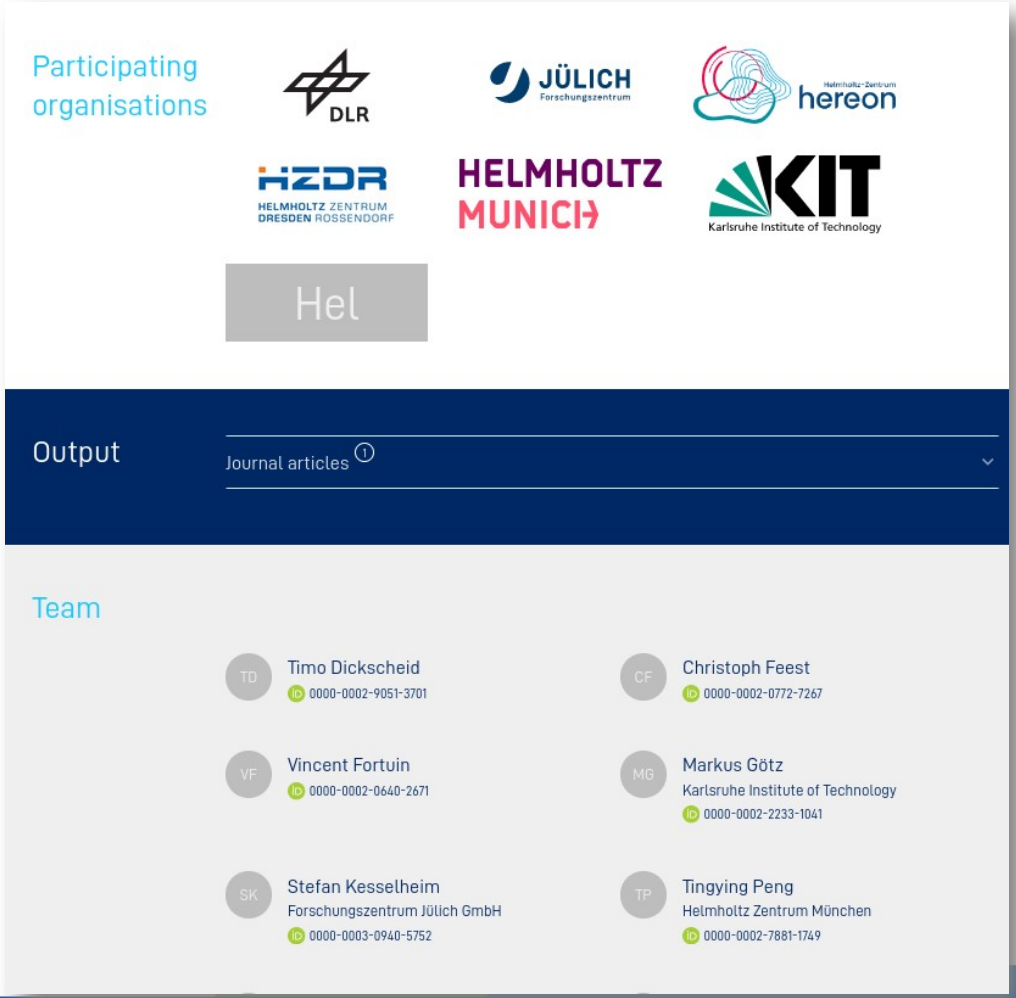
Bulk importing mentions



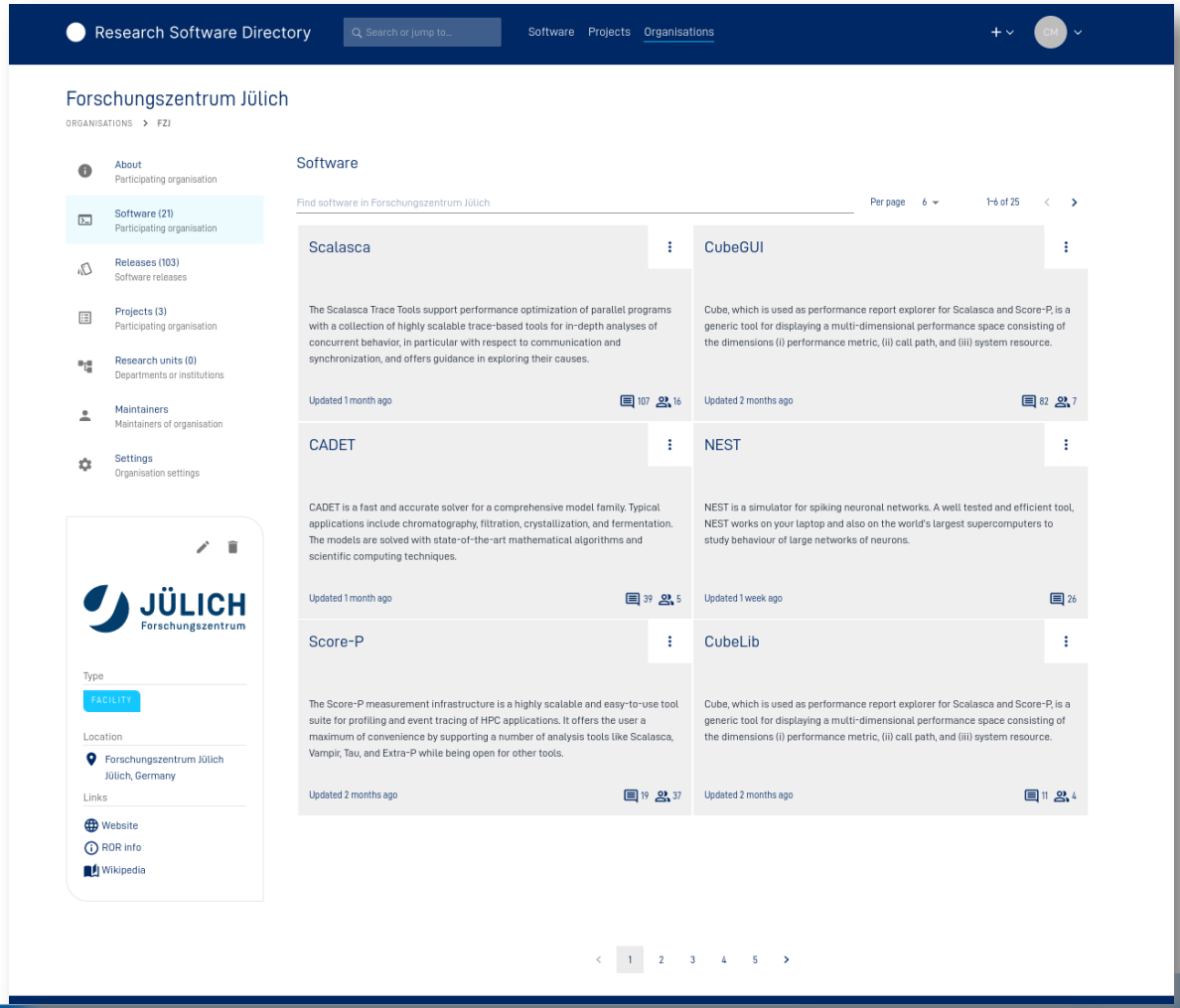


Project pages

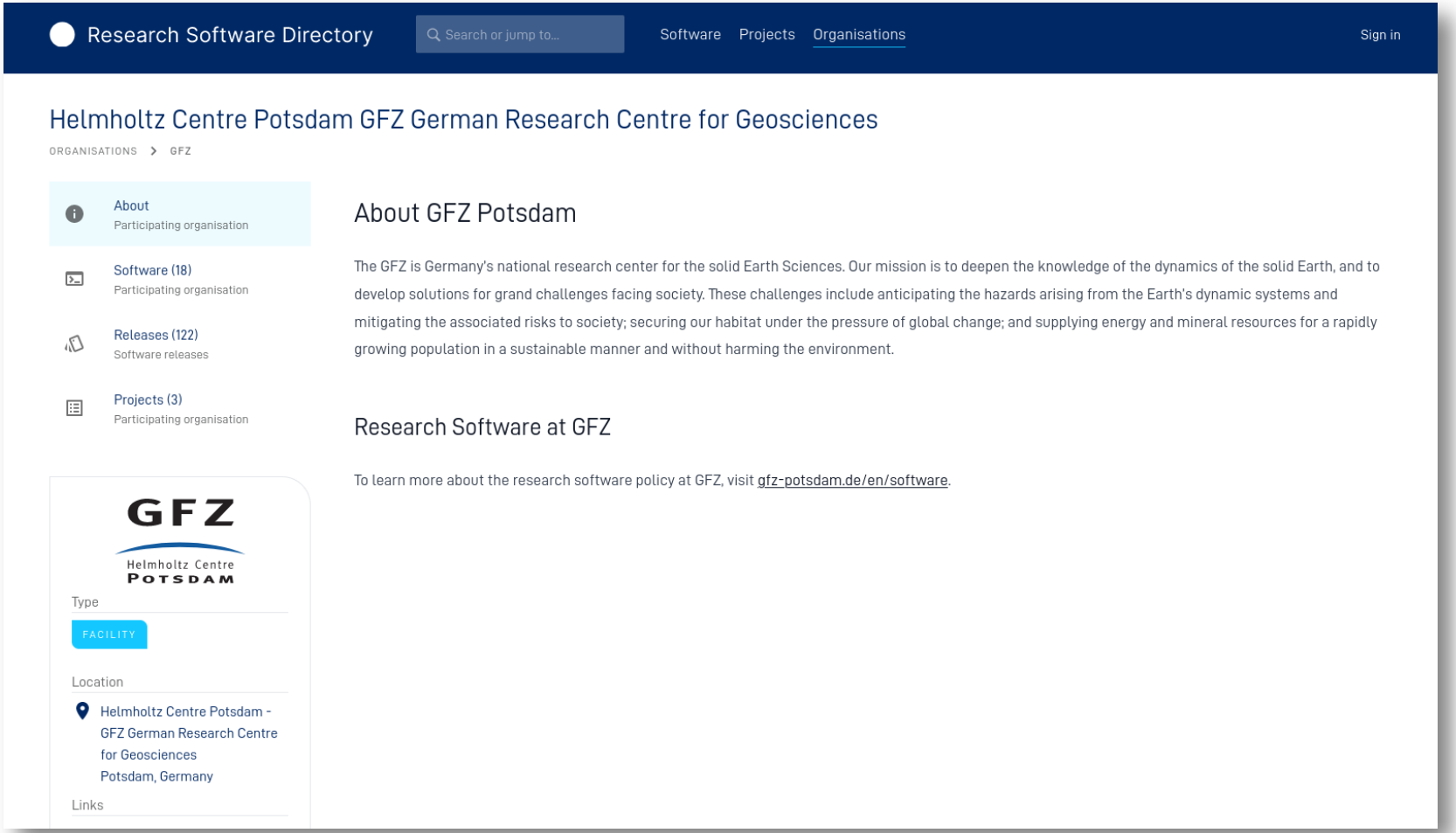




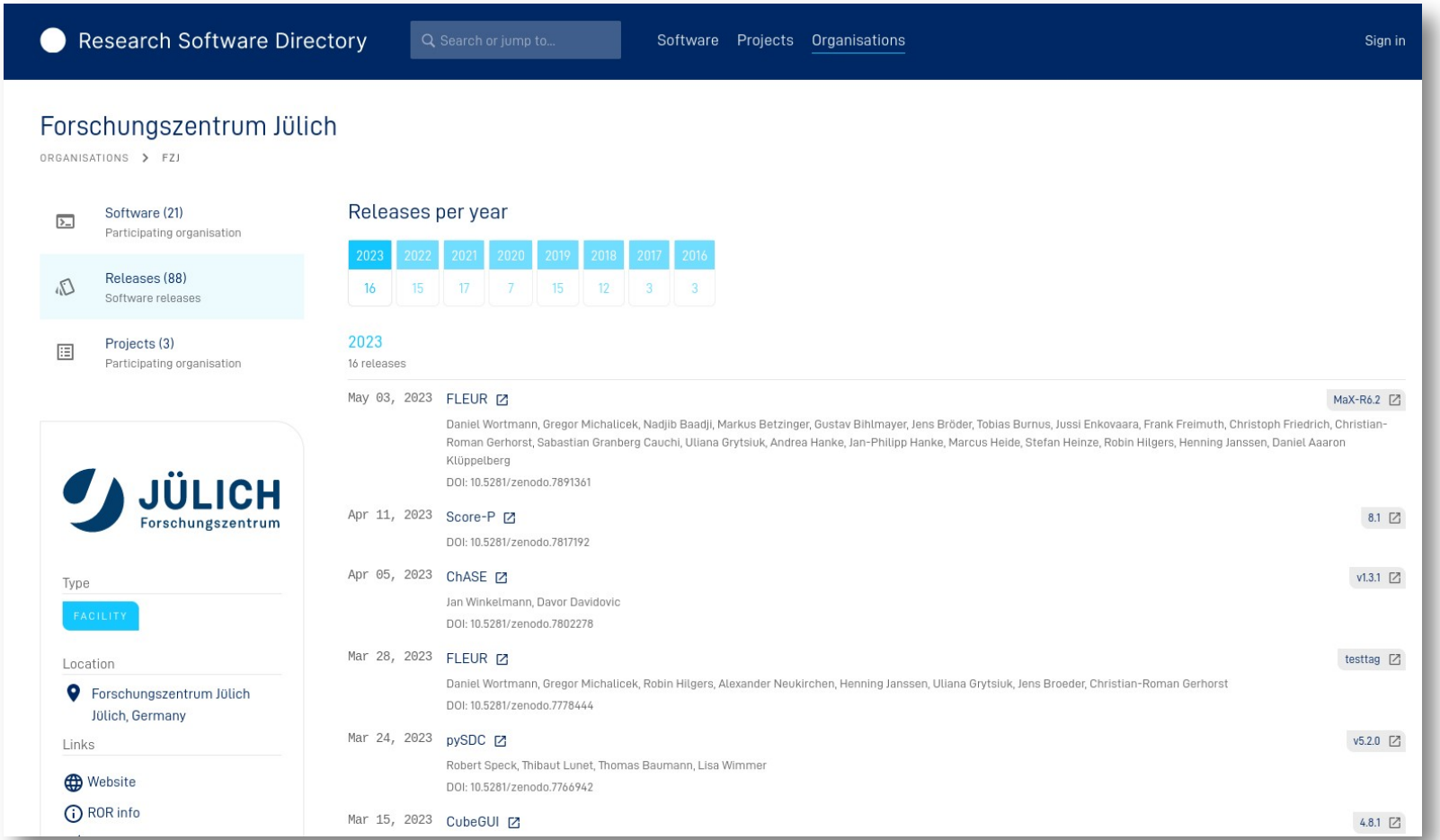
Organisation page – Software Overview



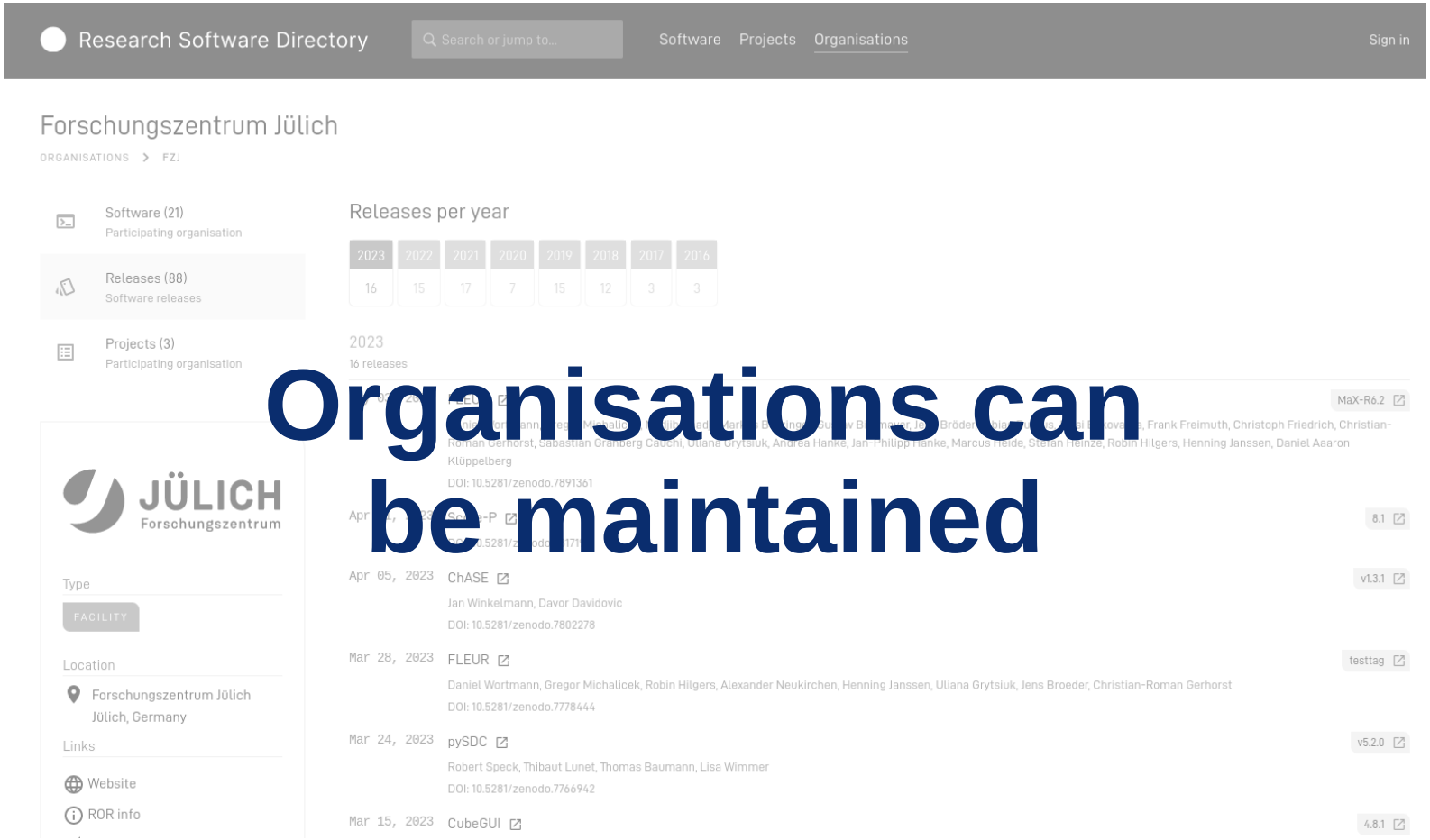
Organisation page – About section



Organisation page – Release metrics



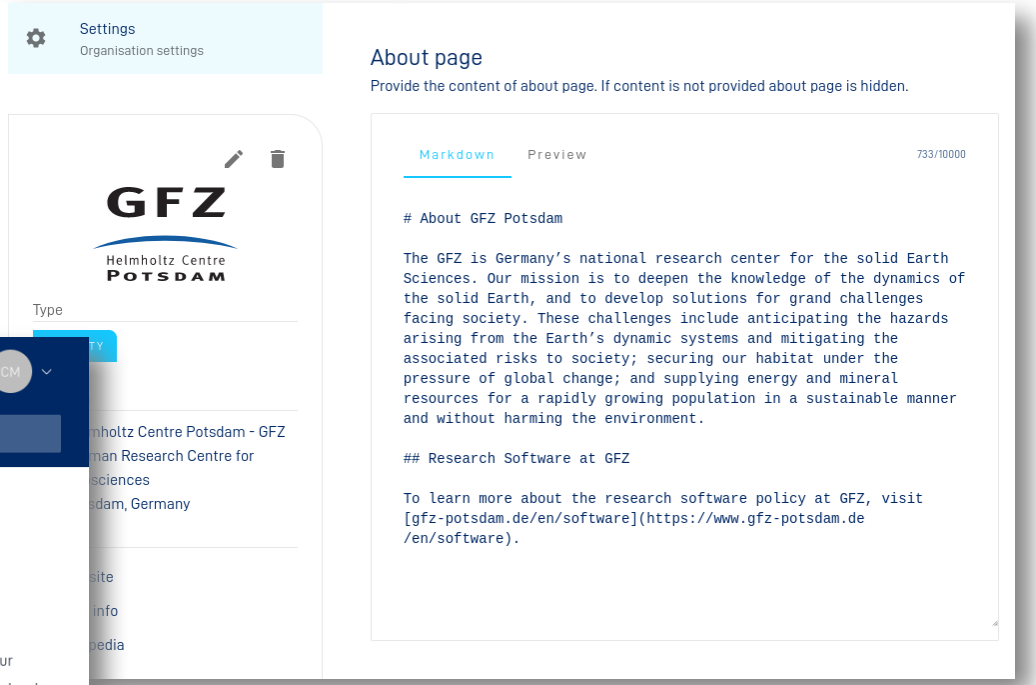
Organisation page – Release metrics



Organisations can be maintained

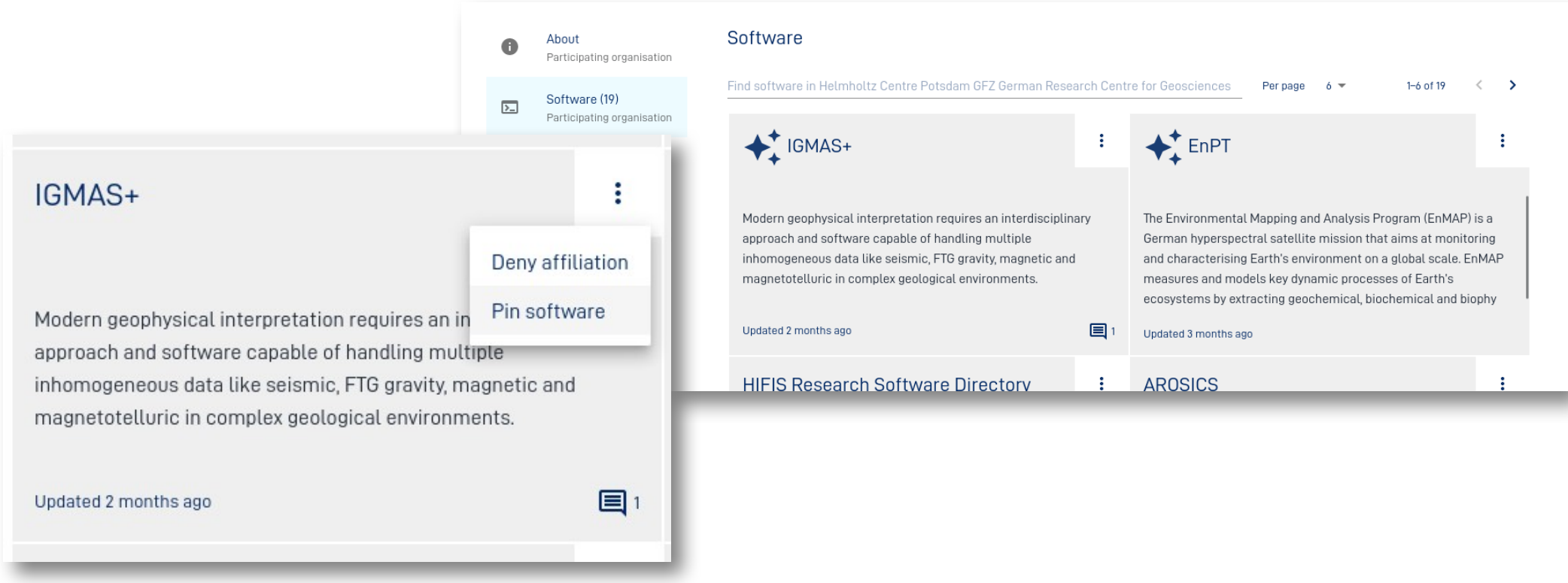
Maintaining organisations
About section

- Important information the visitor should learn about



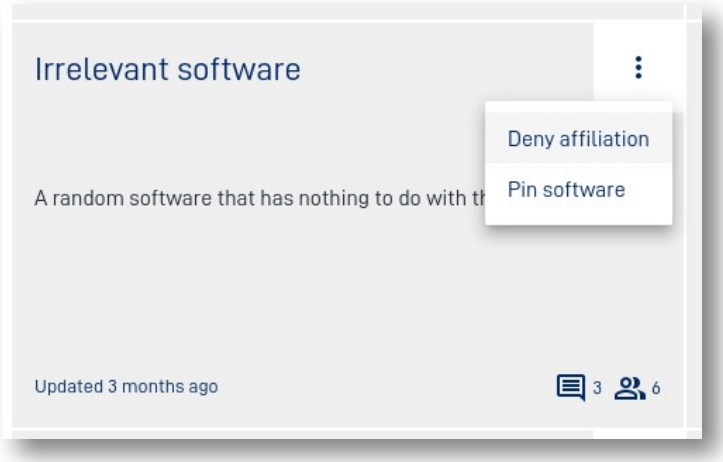
Maintaining organisations
Pinning software

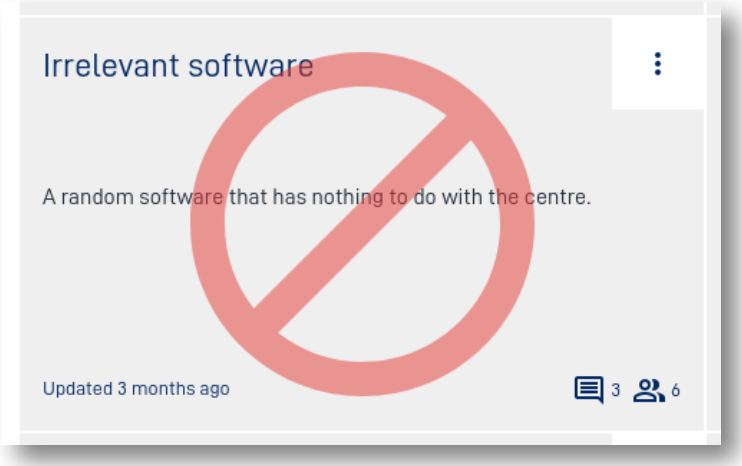
Highlight important software by pinning it to the start of the list



Maintaining organisations
Approving affiliations

- Deny affiliations with software or projects that are not related to the organisation
- After denying: software/project not shown in the publicly visible list
- Relation still visible for organisation maintainers until software maintainer removes link





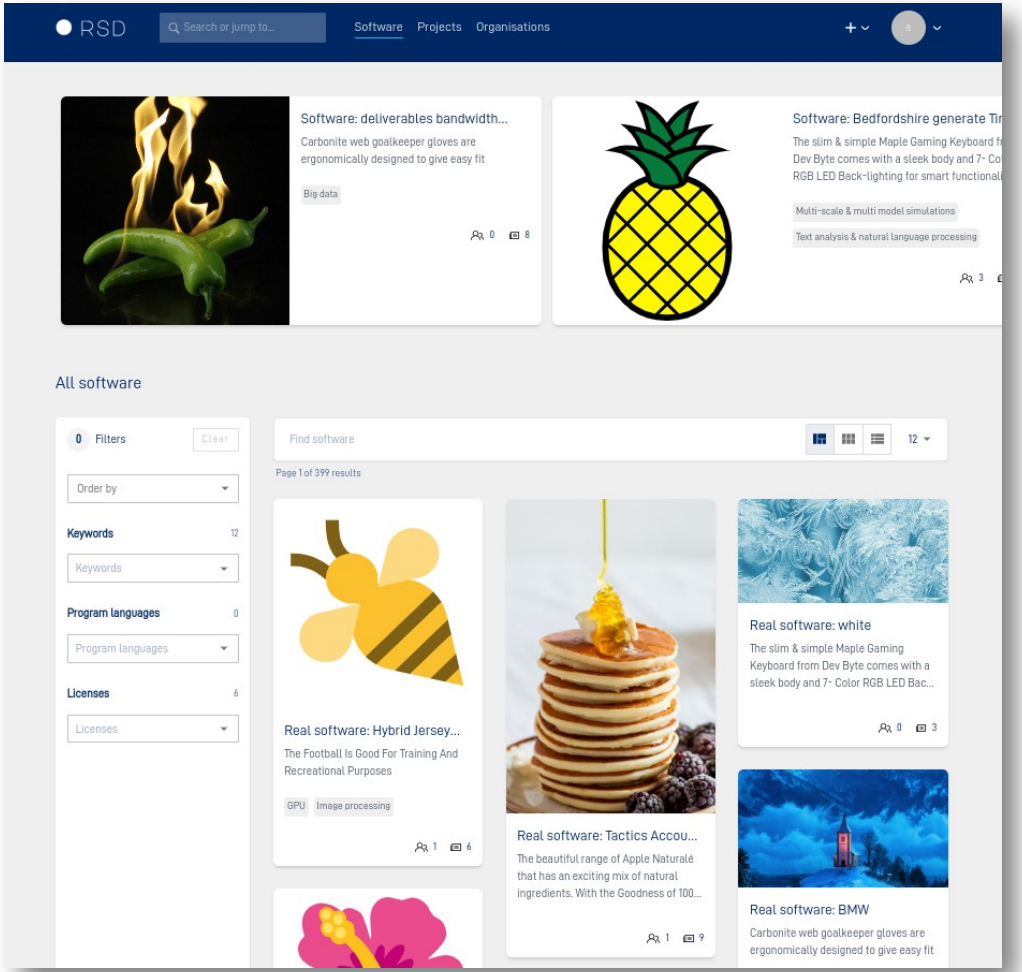
Call for organisation maintainers

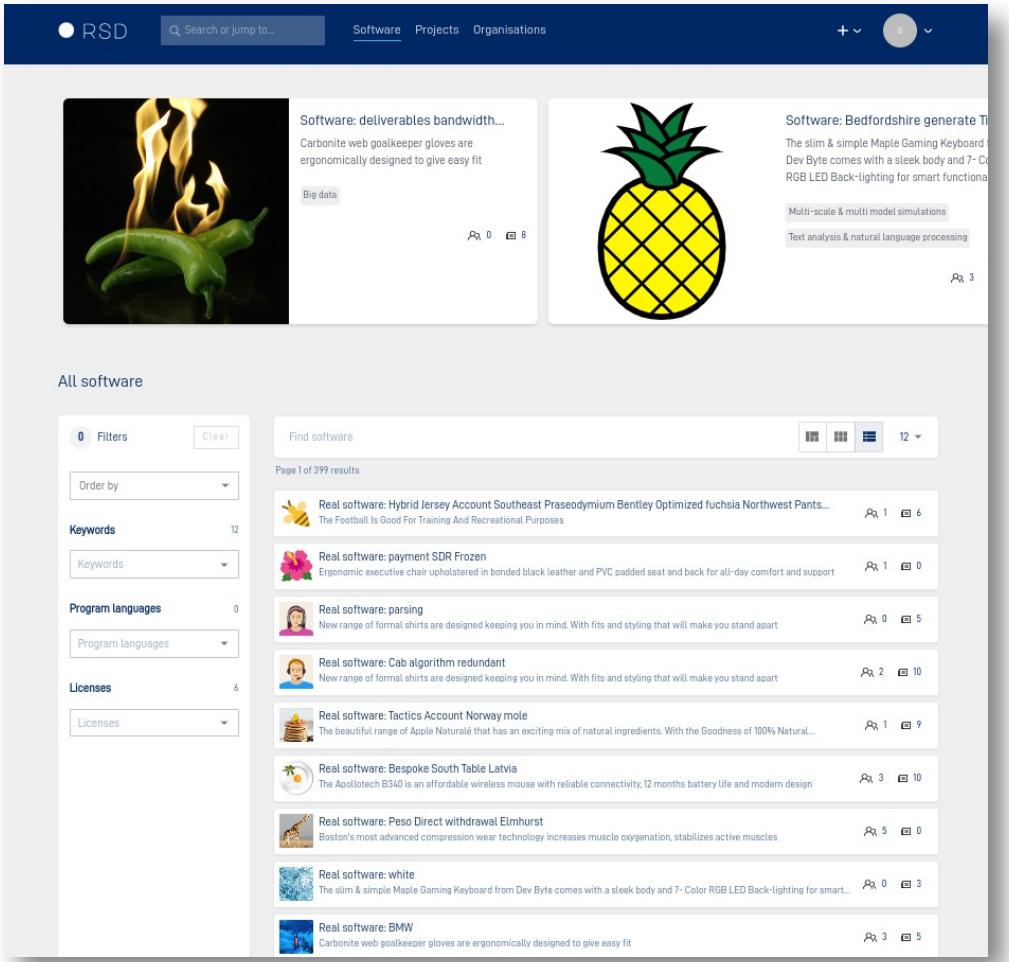
Dear centres, please nominate a maintainer for their organisation in the RSD.

Contact us via

support@hifis.net

Sneak preview – Software list update





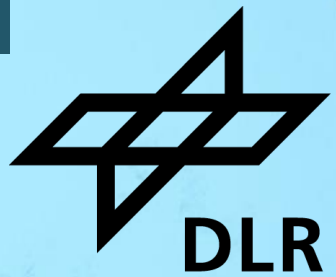
Appendix 6 — Software Licensing

(Tobias Schlauch, DLR)

SOFTWARE LICENSING

Helmholtz Open Science Forum: Research Software Policies, 23.05.2023, Berlin-Adlershof

Tobias Schlauch <Tobias.Schlauch@DLR.de>
Institute for Software Technology
German Aerospace Center (DLR)
http://www.dlr.de/sc

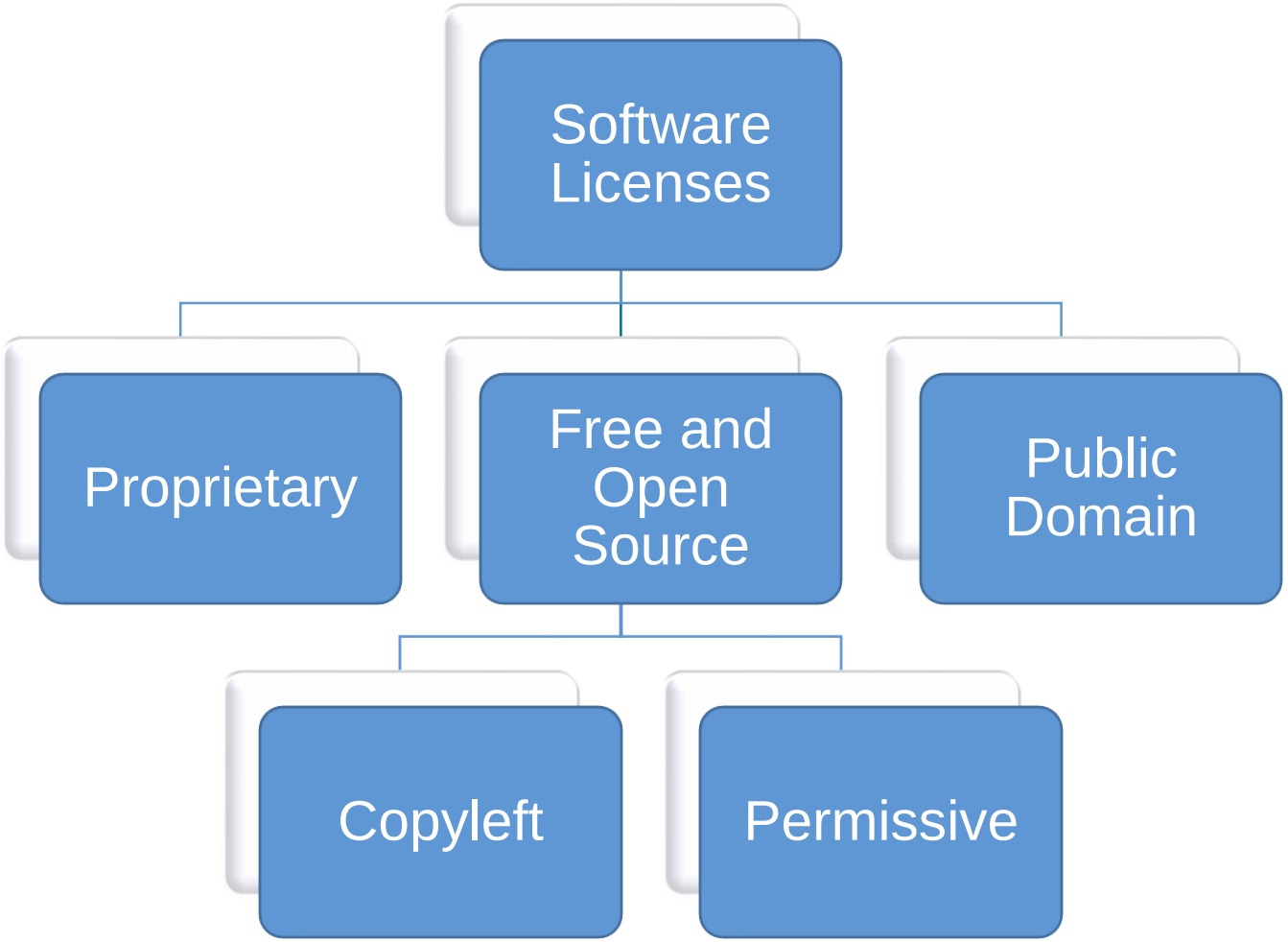


Copyright Basics

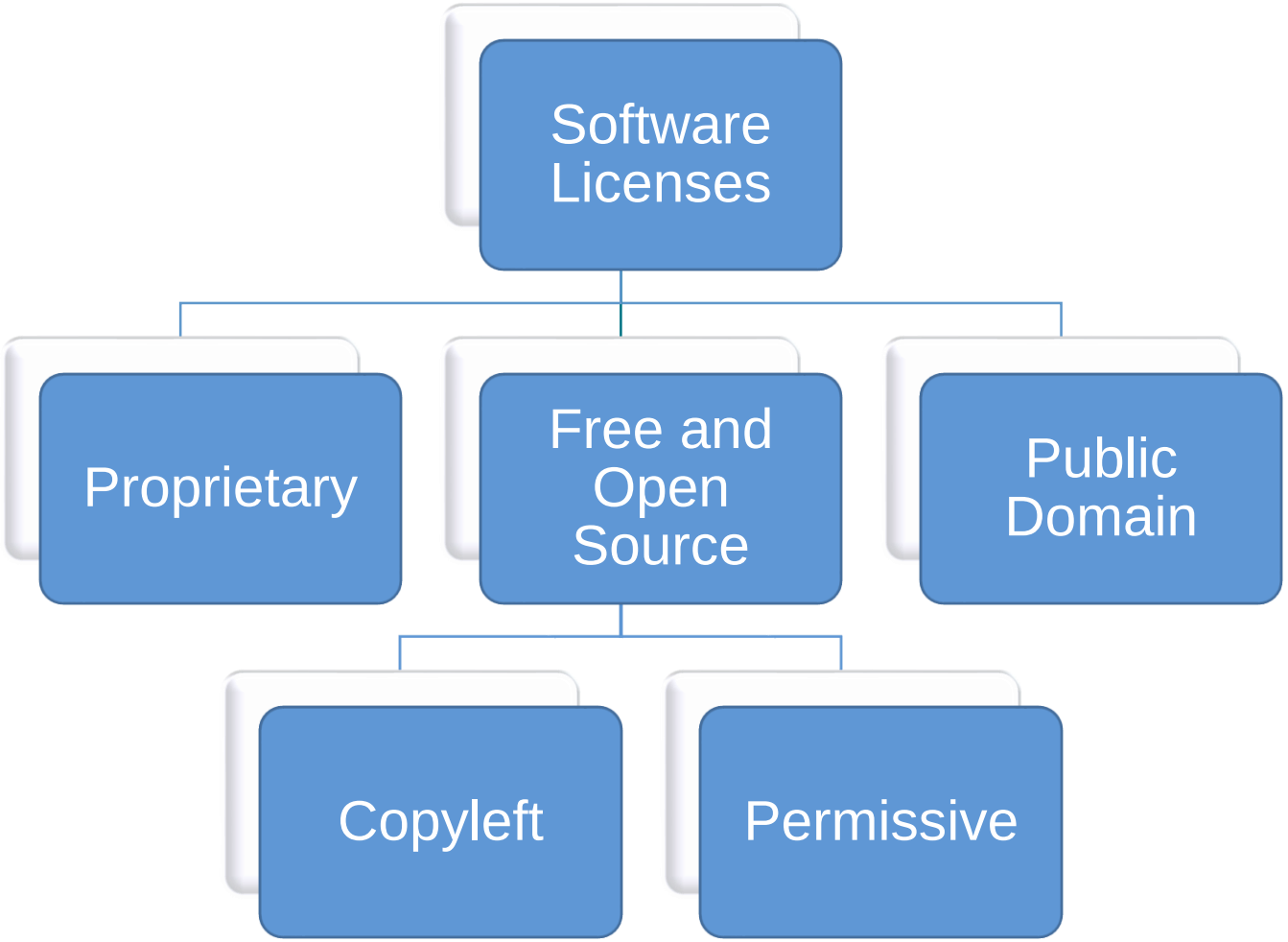
- **Copyright**
  - Software is protected by copyright.
  - Copyright protects the expression of an idea.
  - Copyright grants exclusive rights to the copyright holder.

- **Who is the copyright holder of a software?**
  - All contributors are considered as copyright holders and jointly exercise the rights granted by copyright.
  - A company paying an employed developer obtains most of the exclusive rights.

Software License Types



Software License Types



Make sure that every code that you use is covered by a license!

Take care when combining code under different licenses!

Minimal License Checklist

1. Choose a license

- Consider strategical implications
- Comply with licenses of third-party dependencies

2. Ask your boss for permission

3. Add copyright holder and license information

Minimal License Checklist

1. Choose a license

- Consider strategical implications
- Comply with licenses of third-party dependencies

2. Ask your boss for permission

3. Add copyright holder and license information

Find out about your organizational processes!

Ask for legal advice if you are unsure!

Minimal License Checklist

1. Choose a license

- Consider strategical implications
- Comply with licenses of third-party dependencies

2. Ask your boss for permission

3. Add copyright holder and license information

- **Please note:**
  - This is a simplified checklist assuming that you are the only creator of the software and there are no other legal aspects in the way!
  - DLR Open Source Brochure (German only) provides further detailed information.

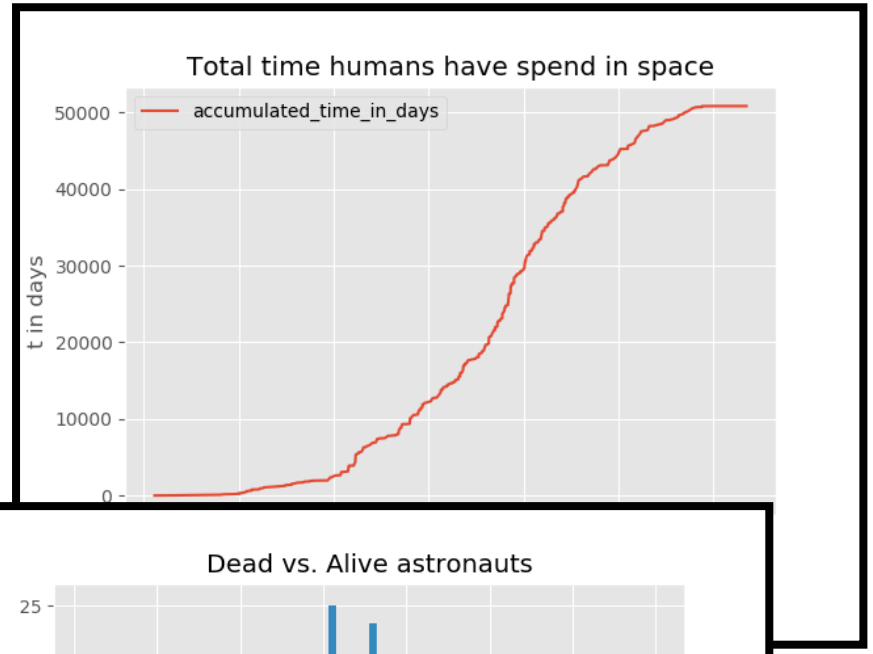
Find out about your organizational processes!

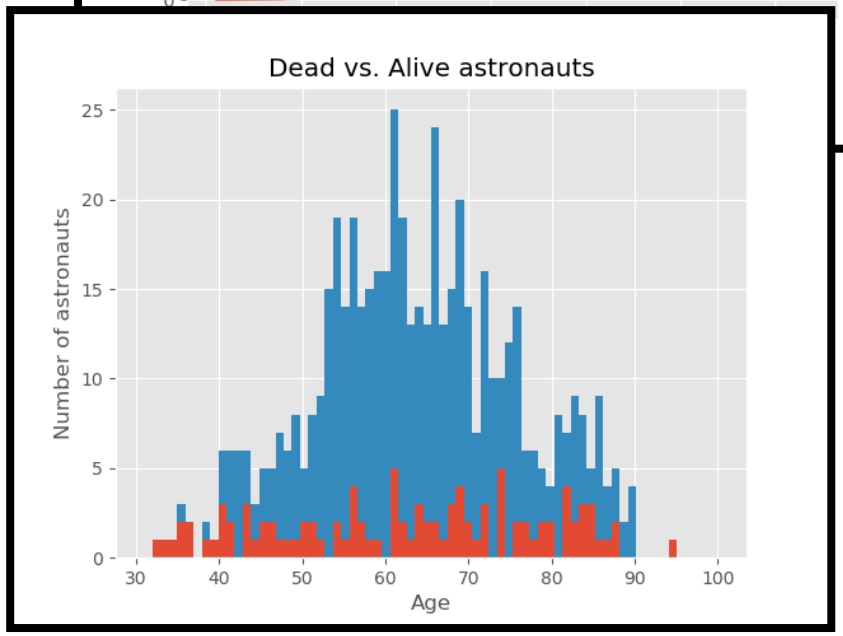
Ask for legal advice if you are unsure!

EXAMPLE

Example: Astronaut Analysis

- **Astronauts Analysis is a data publication consisting of:**
  - Data set
  - Analysis script written in Python using pandas and matplotlib
  - Result plots
- **Scenario:**
  - I created it on my own as part of my job.
  - I want to publish it with my research paper.
  - I want to make its reuse as easy as possible and make it available under an open source license.





Choose a License

- After checking the recommendation from https://choosealicense.com/, I want to use the MIT License. But do the licenses of my dependencies fit?

Choose a License

- After checking the recommendation from https://choosealicense.com/, I want to use the MIT License. But do the licenses of my dependencies fit?
- Let us check the dependencies of the analysis script:
  - You can do this check manually or using a tool such as liccheck or pip-licenses.

Choose a License

- After checking the recommendation from https://choosealicense.com/, I want to use the MIT License. But do the licenses of my dependencies fit?

- Let us check the dependencies of the analysis script:
  - You can do this check manually or using a tool such as liccheck or pip-licenses.
  - Results look good ☺ Both libraries are licensed under permissive licenses!

```
$ liccheck -s liccheck.ini -r requirements.txt --no-deps
gathering licenses...
3 packages.
check unknown packages...
3 packages.
    flake8 (3.9.2): ['MIT']
    matplotlib (3.4.2): ['Python Software Foundation']
    pandas (1.2.4): ['BSD']
```

Choose a License

- After checking the recommendation from https://choosealicense.com/, I want to use the MIT License. But do the licenses of my dependencies fit?

- Let us check the dependencies of the analysis script:
  - You can do this check manually or using a tool such as liccheck or pip-licenses.
  - Results look good ☺ Both libraries are licensed under permissive licenses!

```
$ liccheck -s liccheck.ini -r requirements.txt --no-deps
gathering licenses...
3 packages.
check unknown packages...
3 packages.
    flake8 (3.9.2): ['MIT']
    matplotlib (3.4.2): ['Python Software Foundation']
    pandas (1.2.4): ['BSD']
```

But what about the non-code artifacts?

Choose a License
Finalize Decisions

- Although the license decision for our code is fine,
  I do not want to use the MIT license for all content!

Choose a License
Finalize Decisions

- Although the license decision for our code is fine,
  I do not want to use the MIT license for all content!

- **Final copyright and license decisions:**
  - Copyright holder: German Aerospace Center
  - Source code: MIT
  - Data set: CC0-1.0
  - Docs and plots: CC-BY-4.0
  - Insignificant files: CC0-1.0

- **Finally, I need the approval from my boss…**

Choose a License
Finalize Decisions

- Although the license decision for our code is fine,
  I do not want to use the MIT license for all content!

- **Final copyright and license decisions:**
  - Copyright holder: German Aerospace Center
  - Source code: MIT
  - Data set: CC0-1.0
  - Docs and plots: CC-BY-4.0
  - Insignificant files: CC0-1.0

- **Finally, I need the approval from my boss…**

My boss is fine with it ☺ But how do I annotate this information “correctly”?

Add Copyright Holder and License Information
REUSE SOFTWARE

- **Goal:** Make it easy for humans and machines to determine what license a file is licensed under and who owns the copyright!
- Heavily builds on SPDX: https://spdx.dev/
- Provides the reuse helper tool for annotation, validation, and more: https://git.fsfe.org/reuse/tool
- For more information: Tutorial, FAQ, Specification

Add Copyright Holder and License Information
Step 1: Add License Files

- Add all required license files to the `LICENSES` directory
- You can provide them manually or via `reuse download`:

```
$ reuse download MIT CC-BY-4.0 CC0-1.0
Successfully downloaded LICENSES\MIT.txt.
Successfully downloaded LICENSES\CC-BY-4.0.txt.
Successfully downloaded LICENSES\CC0-1.0.txt.
```

- **Notes:**
  - `reuse download` supports all licenses from the SPDX License List Data repository
  - Other licenses can be used as well
  - `reuse download -a` can download all referenced but missing license files

Add Copyright Holder and License Information
Step 2: Add Copyright and Licensing Information

- Add copyright and license information to all files in the code repository
- You can add them manually or via `reuse annotate`:

```
$ reuse annotate --copyright="German Aerospace Center" --license="MIT" code/*
Successfully changed header of code\requirements.txt
Successfully changed header of code\test.sh
Successfully changed header of code\astronaut-analysis.py
```

```
# SPDX-FileCopyrightText: 2023 German Aerospace Center
#
# SPDX-License-Identifier: MIT

""" This script analysis the astronaut data set and creates different plots as result. """
```

- **Notes:**
  - `reuse annotate` handles comment styles and uncommentable files automatically
  - You can have multiple copyright and license statements in each file
  - License identifiers are standardized (see also: SPDX expressions)

Add Copyright Holder and License Information
Step 3: Check Compliance

- You can check whether everything is fine via reuse lint:

```
$ reuse lint
# SUMMARY

* Bad licenses:
* Deprecated licenses:
* Licenses without file extension:
* Missing licenses:
* Unused licenses:
* Used licenses: CC-BY-4.0, CC0-1.0, MIT
* Read errors: 0
* Files with copyright information: 12 / 12
* Files with license information: 12 / 12

Congratulations! Your project is compliant with version 3.0 of
the REUSE Specification :-)
```

- You can use the linter in pre-commit hooks or in your CI/CD workflow.

Summary

- Copyright protects software and grants exclusive rights to the copyright holder(s).
- Software licenses grant rights but also come with obligations:
  - Make sure that every code that you use is covered by a license!
  - Take care when combining code under different licenses!
- Software publication processes depend on your organization:
  - Find out about your process!
  - Ask for legal advice if you are unsure with licensing or copyright aspects!
- REUSE SOFTWARE helps you to annotate copyright and licensing information properly and to validate it continuously.

Discussion: What aspects should decided/provided at Helmholtz center level?

- Publication process
  - Which pre-conditions need to be checked before: copyright holders, project constraints, export control, trademarks, …
  - Define who can give permission to publish a software
  - …

- Practical support
  - Make researchers aware of copyright and licenses issues
  - Provide recommendations for software licenses including the reasoning
  - Provide decisions support and support in case of questions
  - Provide recommendations to document copyright holder and license information
  - …

Copyright and License Information

Thank you!

What are your Questions?

Email: Tobias.Schlauch@dlr.de
Mastodon: https://norden.social/@schlauch
HIFIS Mattermost: @schlauch

HELMHOLTZ

Open Science