Better Software, Better Science

LAPP contribution to Software in EOSC

September 2024, Journées Informatiques IN2P3

Thomas Vuillaume

Who am I (to talk about software quality) ?

Astrophysics background

Turned data scientist

Research Engineer at LAPP since 2021

Interests: data analysis, machine learning & software development

Who am I (to talk about software quality) ?

Astrophysics background

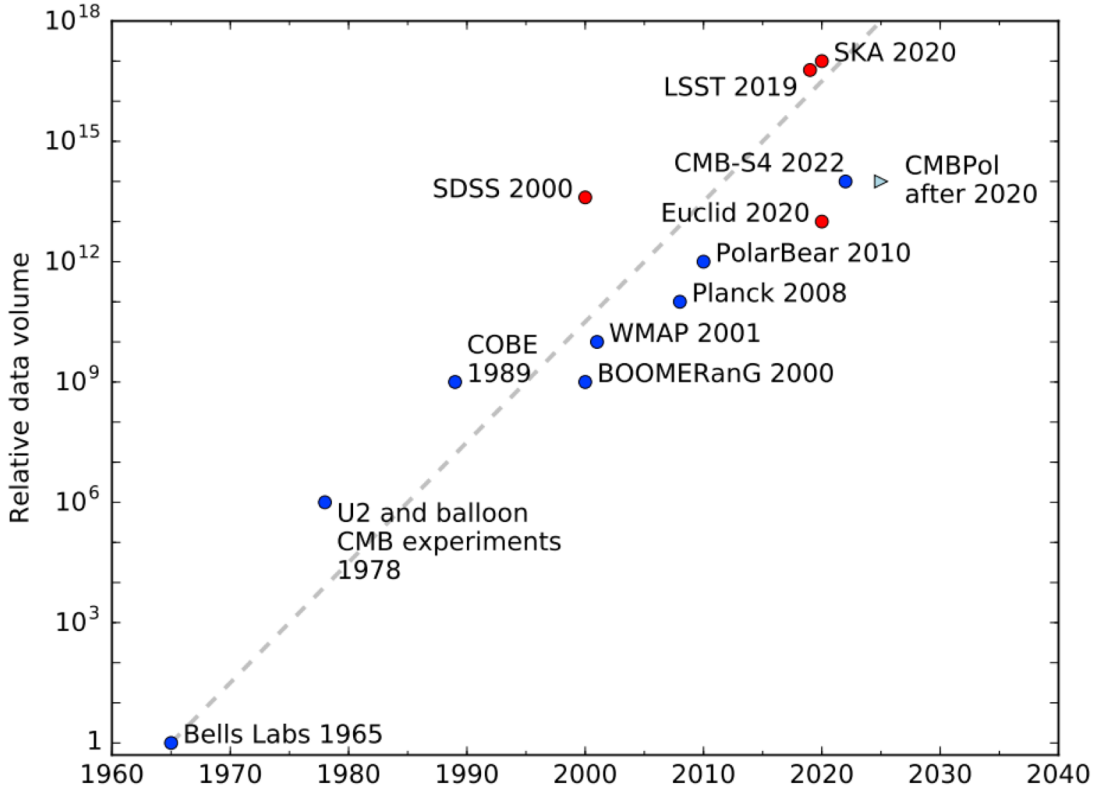
Turned data scientist

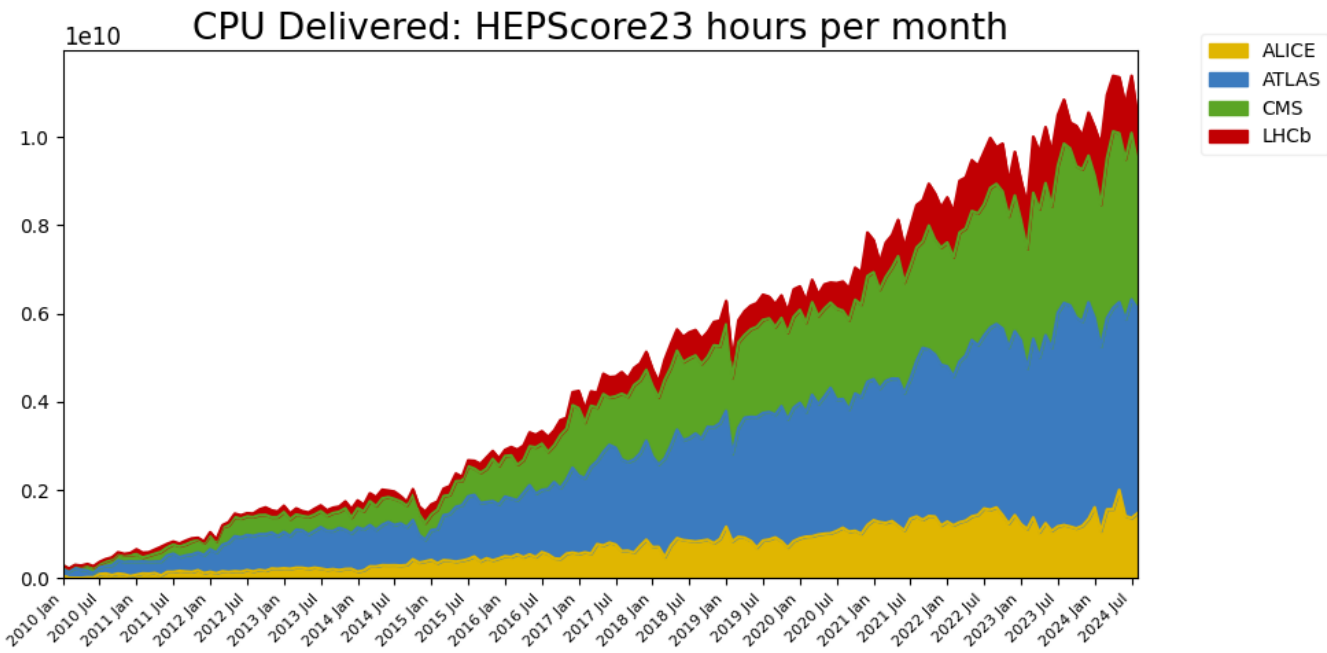
Research Engineer at LAPP since 2021

} I know what bad software is; I write it

Interests: data analysis, machine learning & software development

1. New paradigm: data-driven science



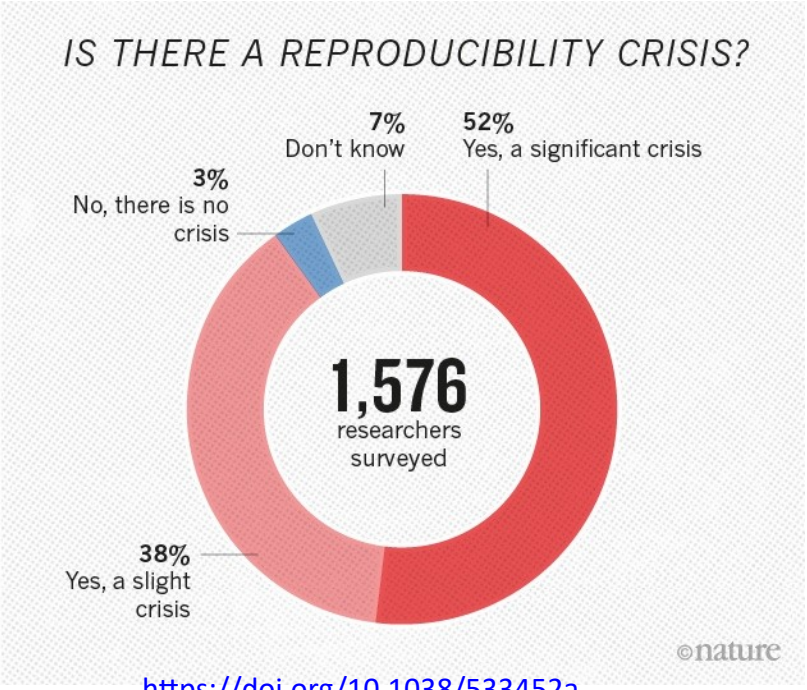


https://wlcg.web.cern.ch/using-wlcg/monitoring-visualisation/monthly-stats

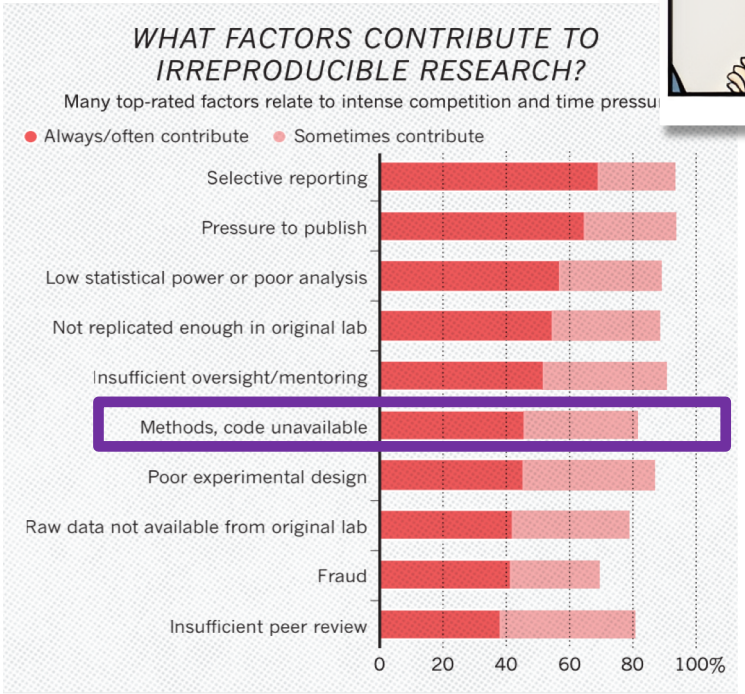
DOI:10.3390/universe2040023

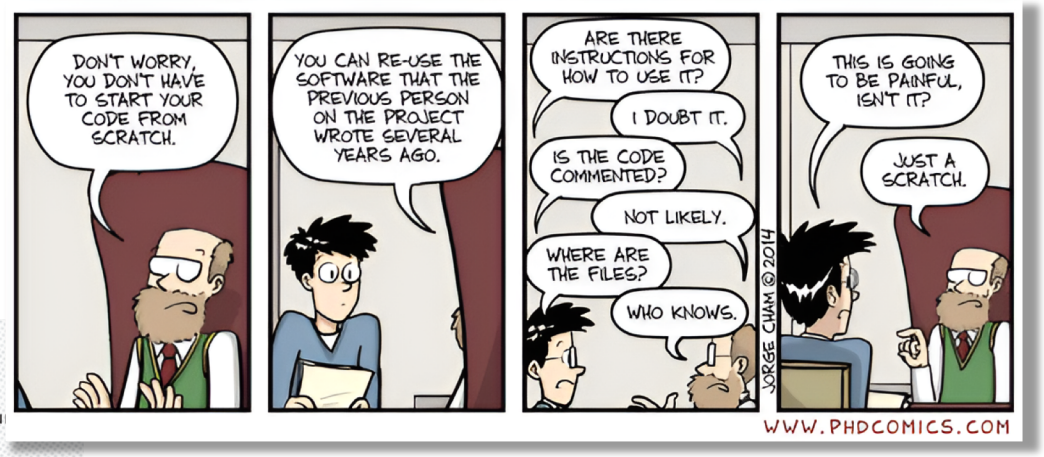
2. Software is not recognized as first-class output

- Software is not shared and reused
  ➔ waste of time, energy and resources
  ➔ **reproducbility crisis**



https://doi.org/10.1038/533452a





https://phdcomics.com/comics/archive.php?comicid=1689

2. Software is not recognized as first-class output

- Research Software Engineers are not recognized
  ➔ short-term contracts often related to specific projects
  ➔ metrics mismatch (they don't publish scientific papers)
  ➔ talents loss to industry

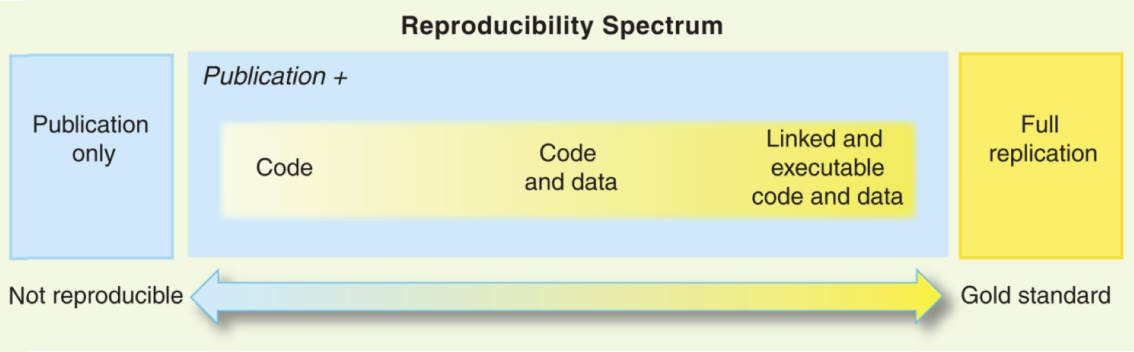
https://www.fz-juelich.de/en/rse/about/what-is-a-research-software-engineer
https://invenia.github.io/blog/2020/07/07/software-engineering/
https://researchcomputing.princeton.edu/news/2021/building-career-path-research-software-engineers
https://www.software.ac.uk/blog/why-research-software-engineers-should-have-permanent-contracts

So, we have two opposing considerations:

1. Software importance is increasing
2. Software importance is not recognized (enough)

What can we do about it ?

1. Publish Open & FAIR software



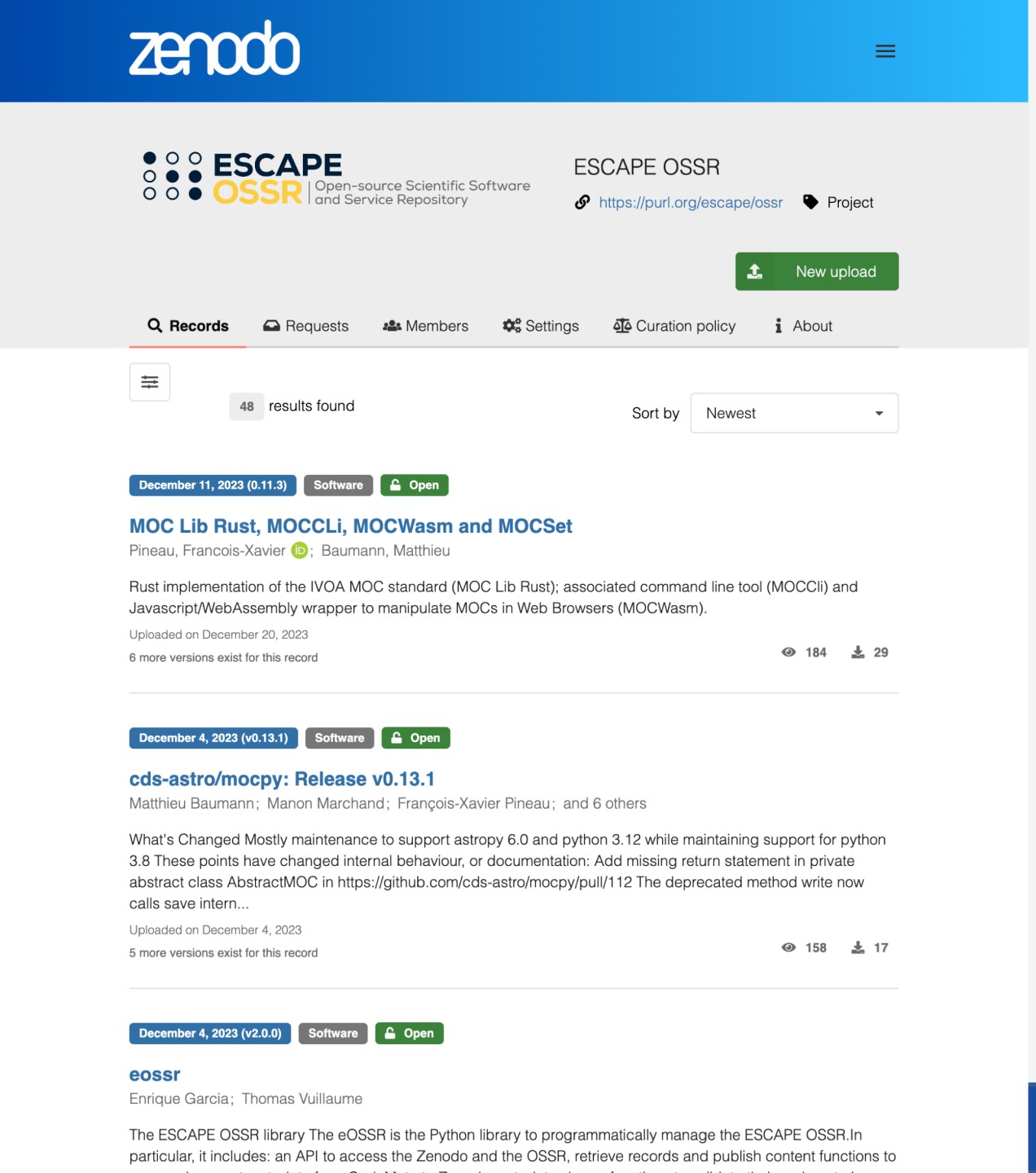
- Findable
- Accessible
- Interoperable
- Reusable

The Turing Way Community, & Scriberia. (2023). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. https://doi.org/10.5281/zenodo.8169292

Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* 9, 622 (2022). https://doi.org/10.1038/s41597-022-01710-x

Open-source Software and Service Repository

- A trusted software repository
- Community Centered
  ESCAPE = Particle Physics and Astronomy Cluster in EOSC
- FAIR
- Long-term
- Curated



Zenodo as 💙

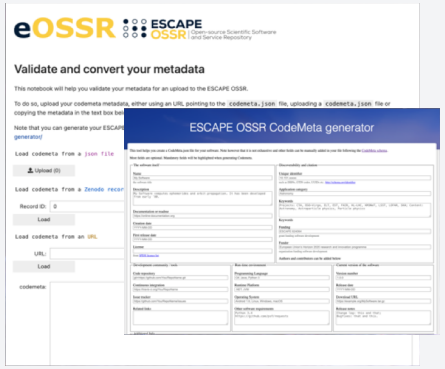
- FAIR centered
- long-term archive
- software citability (DOI)
- widely accepted and used
- don’t reinvent the ⚙️
- integrates with other services
- community management

→ *escape2020* community

The OSSR galaxy

OSSR website
- Information
- Onboarding process

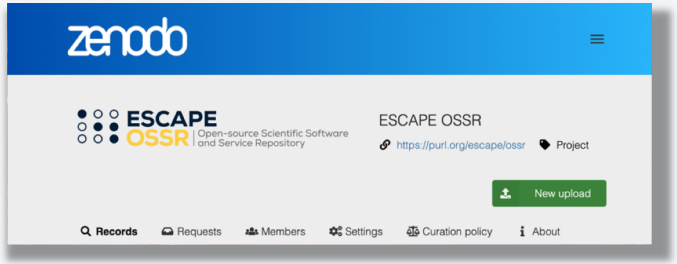
http://purl.org/escape/ossr



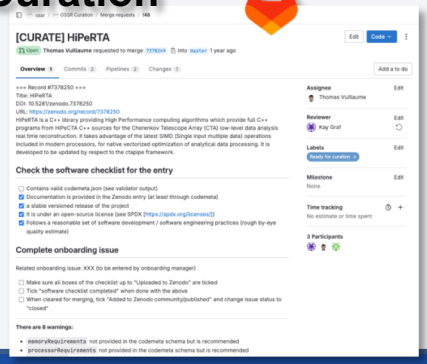
CodeMeta

ESCAPE ESAP ESFRI Science Analysis Platform

OpenAIRE EXPLORE

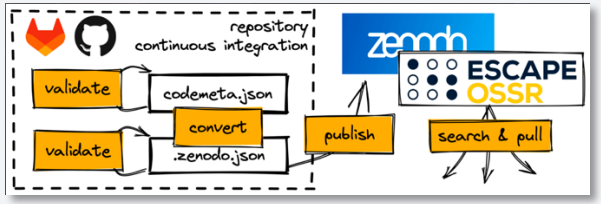


Curation



GitHub GitLab

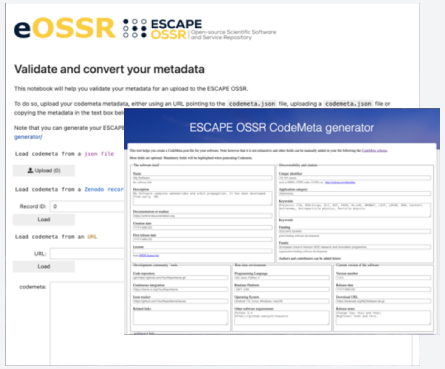
eOSSR



The OSSR galaxy

OSSR website
- Information
- Onboarding process

http://purl.org/escape/ossr

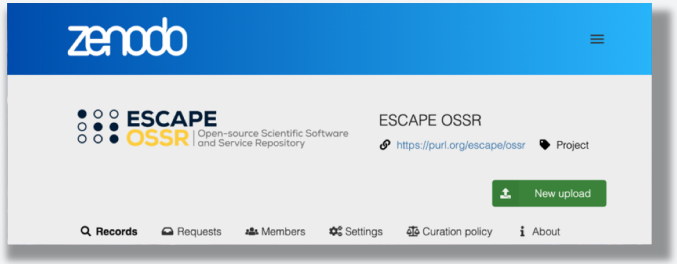


Tools to help RSEs generate the right metadata for their software

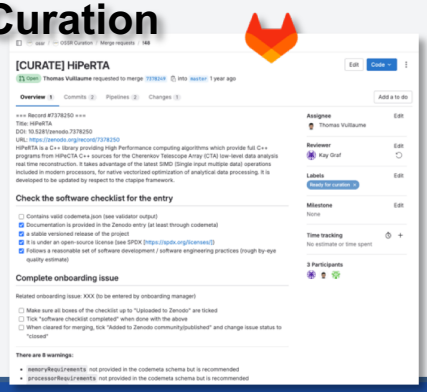
CodeMeta

ESCAPE ESAP ESFRI Science Analysis Platform

OpenAIRE EXPLORE

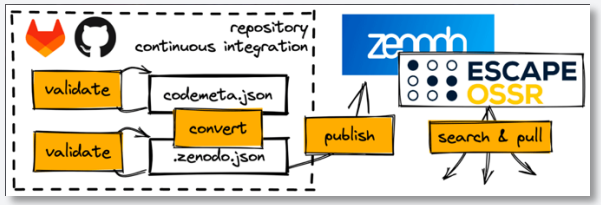


Curation



GitHub GitLab

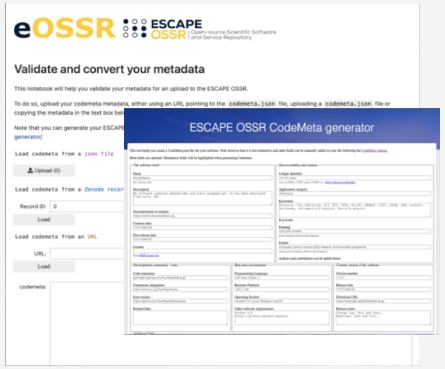
eOSSR



The OSSR galaxy

OSSR website
- Information
- Onboarding process

http://purl.org/escape/ossr

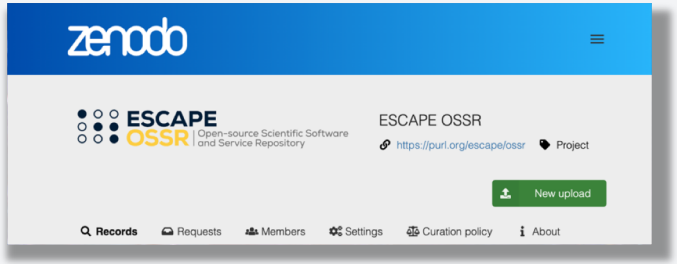


Tools to help RSEs generate the right metadata for their software

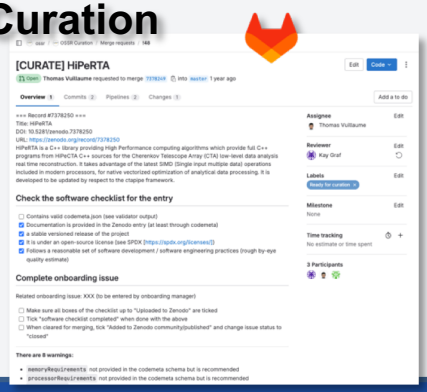
CodeMeta

ESCAPE ESAP ESFRI Science Analysis Platform

OpenAIRE EXPLORE

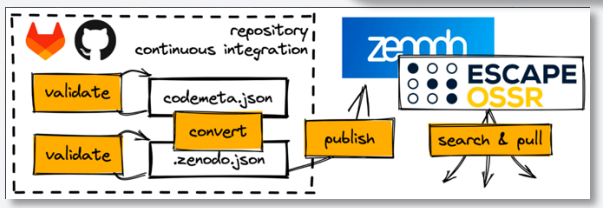


Curation



GitHub GitLab

eOSSR

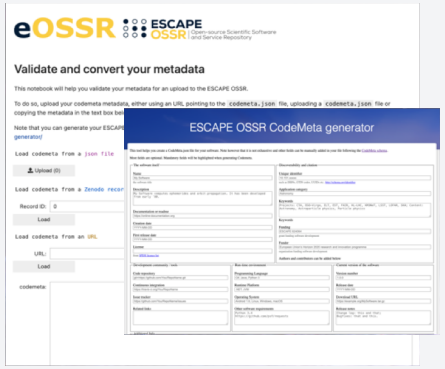


A python library to communicate with the OSSR

The OSSR galaxy

OSSR website
- Information
- Onboarding process

http://purl.org/escape/ossr

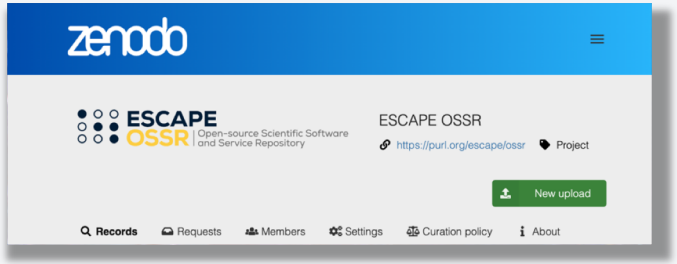


Tools to help RSEs generate the right metadata for their software

CodeMeta

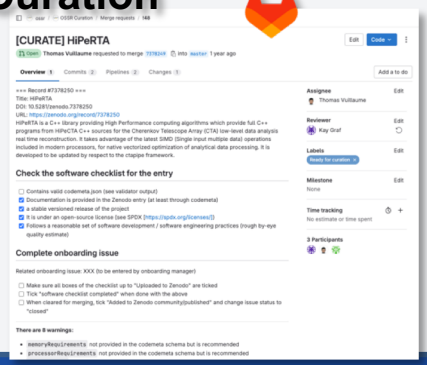
ESCAPE ESAP ESFRI Science Analysis Platform

OpenAIRE EXPLORE



A python library to communicate with the OSSR

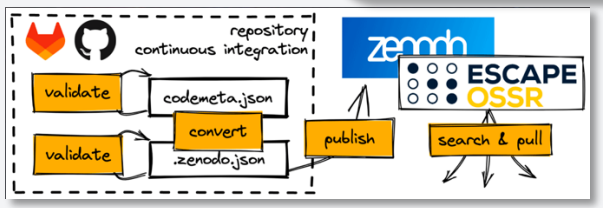
Curation



GitHub GitLab

Tools to automatically publish software from GitHub or GitLab

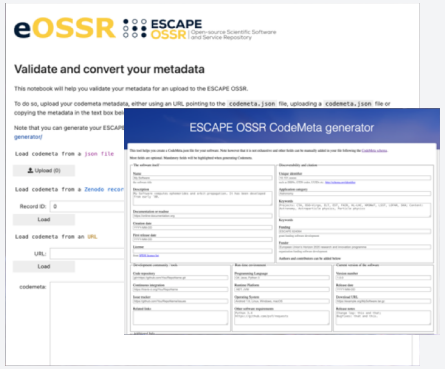
eOSSR



The OSSR galaxy

OSSR website
- Information
- Onboarding process

http://purl.org/escape/ossr

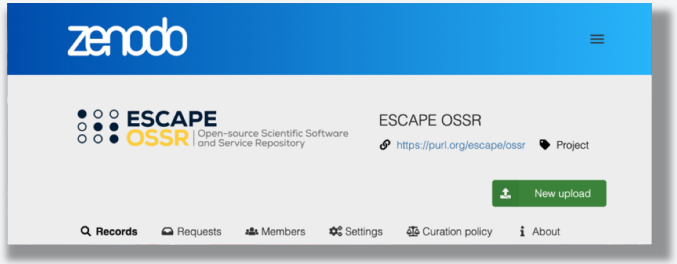


Tools to help RSEs generate the right metadata for their software

CodeMeta

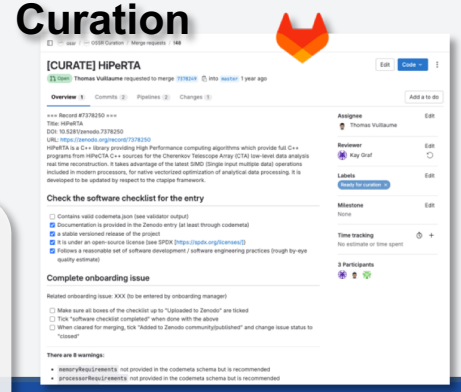
ESCAPE ESAP ESFRI Science Analysis Platform

OpenAIRE EXPLORE



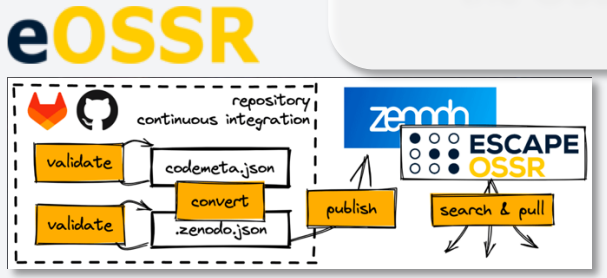
A python library to communicate with the OSSR

Curation



A curation platform to review the requests

GitHub GitLab

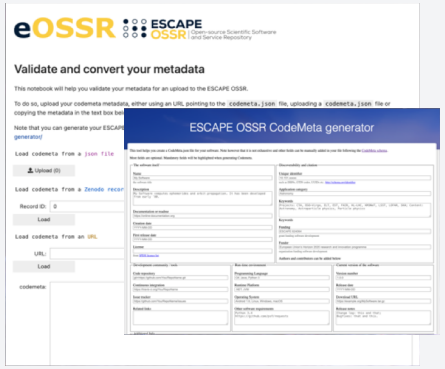


Tools to automatically publish software from GitHub or GitLab

The OSSR galaxy

OSSR website
- Information
- Onboarding process

http://purl.org/escape/ossr



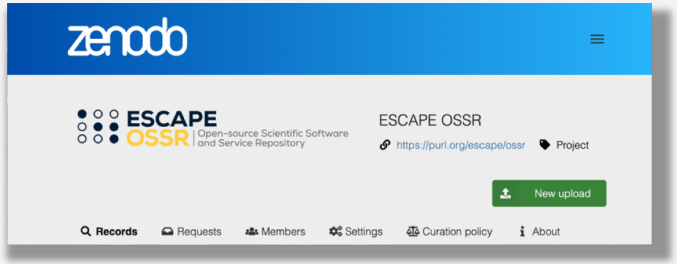
Tools to help RSEs generate the right metadata for their software

CodeMeta

A connection to other services

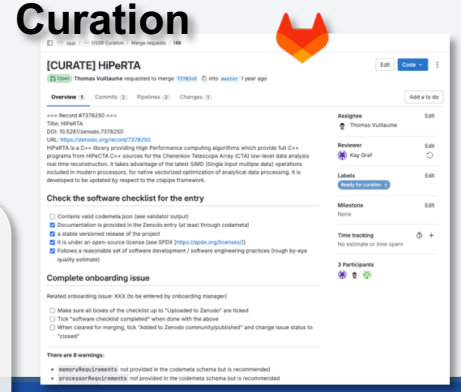
ESCAPE ESAP ESFRI Science Analysis Platform

OpenAIRE EXPLORE

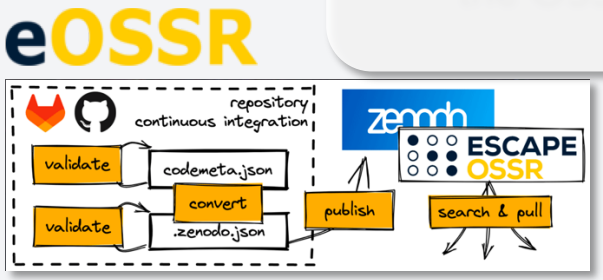


A python library to communicate with the OSSR

Curation



GitHub GitLab



A curation platform to review the requests

Tools to automatically publish software from GitHub or GitLab

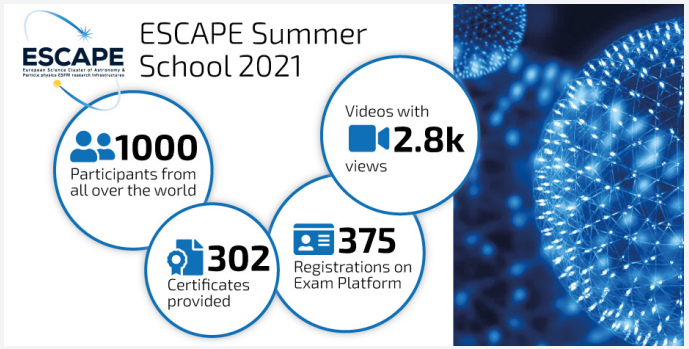
OSSR resources

- http://purl.org/escape/ossr

- Vuillaume T, Al-Turany M, Füßling M et al. The ESCAPE Open-source Software and Service Repository, Open Res Europe 2023, https://doi.org/10.12688/openreseurope.15692.2

2. Improve software quality

- Programming schools at LAPP since 2017



Code development for physicists:

- Coding environment and good code practices
- Version control and collaborative development
- Debugging and profiling
- Python packaging
- Scientific libraries for data science and analysis
- Machine learning

All courses open-source, recorded and available online DOI 10.5281/zenodo.5093909

- HPC
- Heteregeneous architectures
- Code optimisation
- 12 satellite sites

https://indico.in2p3.fr/event/14227/
https://indico.in2p3.fr/event/16864/
https://indico.in2p3.fr/event/18333/

https://escape2020.github.io/school2021/
https://escape2020.github.io/school2022/

https://indico.in2p3.fr/event/30939/

eosc | EVERSE

EVERSE

Paving the way towards a European Virtual Institute for Research Software Excellence

Funded by the European Union

Pilots & Drivers

Environmental Sciences: *Integration of Science Cluster ENVRI through ENVRI-HUB*

- Integrate EVERSE framework into the ENVRI-HUB Knowledge base and Virtual Research Environment
- Apply to the development of the Essential Climate Variable computing program and cloud workflows

Life Sciences: *Integration of Science Cluster EOSC-Life through ELIXIR*

- Make RO-Crate actionable by incorporating the five safes concept into WfExS for secure and federated workflow orchestration
- Use of community-led standards for materialising research software packaged using container technologies and mobilising encrypted data whenever needed

Astronomy and particle physics: *Integration of Science Cluster ESCAPE through the Dark Matter Test Science Project*

- ML for scientific data compression (standalone code, python)
- A Common Tracking Software
- Choose an ATLAS trigger algorithm as an option for the collaboration

Proton and neutron science: *Integration of Science Cluster PaNOSC through LEAPS/LENS*
Transition software to high performance computing (HPC) and heterogeneous computing architectures

Social sciences: *Integration of Science Cluster SSHOC*
Develop a multilanguage textual analysis pipeline of tools that use a combination of open source tools and own code to create an integrated SotA tool capable of deploying locally or as a service

Paving the way towards a European Virtual Institute for Research Software Excellence

EVERSE aims to create a framework for research software and code excellence, collaboratively designed and championed by the research communities, in pursuit of building a European network of Research Software Quality and setting the foundations of a future Virtual Institute for Research Software Excellence

ensure research software curation, quality, preservation and adoption of best practices, by the Communities, for the Communities, build on collaboration with the five EOSC Science Clusters

adopt a three-tier model for research software, i.e., analysis code, prototype tools and research software infrastructure, which captures the varying complexity of research software and its development, and can be used as a basis for research software excellence

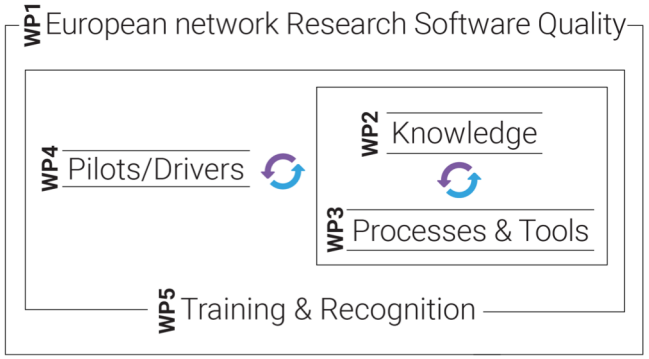
credit and recognition for both developers and software are essential components of our strategy to promote sustainable software practices

Mar/2024 → Feb/2027 (36 months)

15 Beneficiaries, 1 Associated partner & 2 Affiliated entities

Coordinated by CERTH

Project objectives



- ✓ Provide a **framework** that will ensure appropriate **recognition, reward, and career development** for researchers and RSEs who implement research software and code quality assurance practices and policies
- ✓ **Leverage existing tools and resources** to support the evaluation, verification and improvement of research software and code quality, based on **existing practices and standards** across research communities represented by the five EOSC Science Clusters.
- ✓ Establish a **sustainable and collaborative ecosystem of stakeholders** across the research communities associated with the five **EOSC Science Clusters** to ensure research software and code quality assurance and support the advancement of reliable and reproducible research.
- ✓ Build a **collaborative, community-led structure** for evaluating, verifying, and improving the quality of research software and code, by **actively involving** researchers, software developers, and other stakeholders in the research community.

WP1: Network

- Establish the **EVERSE Network of Research Software Quality** (T1.1)
  - A meeting point for a Community of Practice for software quality
    - Open to individuals and organisations internationally
  - Will be an EOSC Expert Group on software
  - Help to shape best practices and get help and advice from like minded partners
  - Will host seminars, webinars and other events to promote and recognise software quality
  - The network is forming now, so we are very interested to work with any *early adopters*
- Reaching out to **Science Communities** (T1.2)
  - Showing the benefits of the EVERSE network and matching specific domain needs
  - Organisation of events such as hackathons to promote best practive and network involvement
- Links to **European, international organisations and industry** (T1.3)
  - Benefit from and reinforce existing policies and practice regarding software quality
  - Align and co-develop shared solutions and host common events

WP2: Best practices

Community-led best practices for developing high-quality research software

Each research community has its own techniques for computing-based research. Work package 2 works towards assessing all these approaches and developing common grounds for best practices applicable to all developers. The Research Software Quality kit will be a knowledge hub that collects existing expertise while ensuring improvement.

The curated best practices are gathered across four pillars: technical quality, FAIR principles, Openness and Software Sustainability. They are used to find a common framework and act as measurables for indicators of software quality. Woven into the RSQkit, these practices will be incorporated into the future Virtual Institute driven by WP1.

CURRENT ACTION: conducting **survey** among communities => **landscaping** of best practices (T2.1)

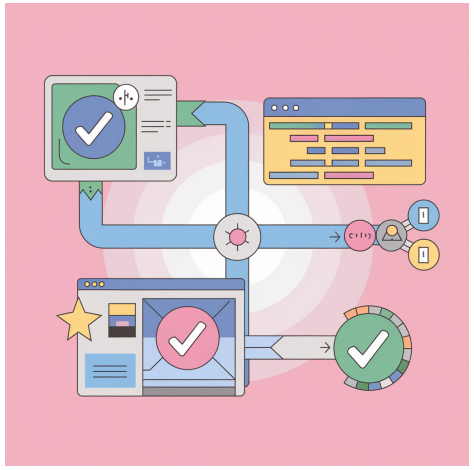
UPCOMING: consolidate/curate best practices in RSQkit

WP3: Tools and services for software quality and FAIRness

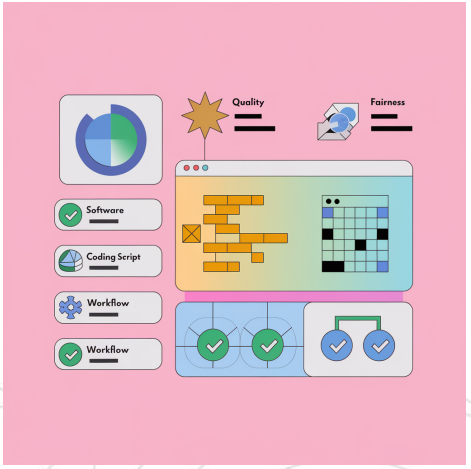
O3.1 (on-going): To establish a **technology watch** identifying and **gathering tools and services** targeting scientific software, code, and workflows **quality and FAIRness**

O3.2 (2nd phase): To assist the Science Clusters in **measuring and improving** software, code, and workflows **quality and FAIRness globally** by combining existing tools and services into common frameworks

Technology Watch for tools and services for softwarequality



Integrated pipelines to measure and improve software quality



Dashboards to measure globally the software quality and its evolution

WP5: Capacity Building and Recognition

- **Collect, curate and enhance training resources**
  - Align training material with best practices in the science clusters
  - Conduct a landscape analysis of existing material and fill the gaps as needed
  - Make training resources available via tools developed by EVERSE partners (e.g. TeSS) and connect those to the RSQKit
- **Build a framework for recognition of trainers and RSEs**
  - Start from tools developed within EVERSE (e.g. Bip!Scholar, Apicuron) and extend as needed
  - Align with EOSC “Research careers, recognition and credit” task force
- **Establish long-term training activities**
  - Incorporate training into existing curricula and make those available (e.g. via a “monthly discovery series”)
  - Develop a network of trainers and establish a process to keep them engaged
  - Provide feedback to universities and schools on needed training for research software engineering

What's next?

- Improve OSSR with EVERSE outcomes
  - improve curation and automated code review
  - provide high-level software quality metrics to developers, reviewers, instances
  - better recognition of software developers
- Extend or bring OSSR framework to other communities
- Continue training scientists (not only in physics) to modern software development
- Think about LLMs and code assistants impact on software development in research
  - Software development will **drastically change** in the coming ~~years~~ months
  - Software quality ? Up or down ?
  - Code generation ? Code reviews ?

CONCLUSION

- LAPP is working to improve research software quality and to promote its importance to instances as well as to individuals that produce it.
- We are involved in several EU projects to reach that goal.
  - ASTERICS (2016-2019)
  - ESCAPE (2019-…) - https://projectescape.eu/
  - EVERSE (2024-…) - https://everse.software/
  - OSCARS (2024-2027) - https://oscars-project.eu/
  - CodeMetaSoft (2025-2027)
- Contact me if you are interested or want to participate.

BACK-UP SLIDES

Software metadata

Software metadata are the implementation of FAIR principles

- Findable, Interoperable
- They should be part of the software and not defined or retained by an external service

OSSR uses CodeMeta

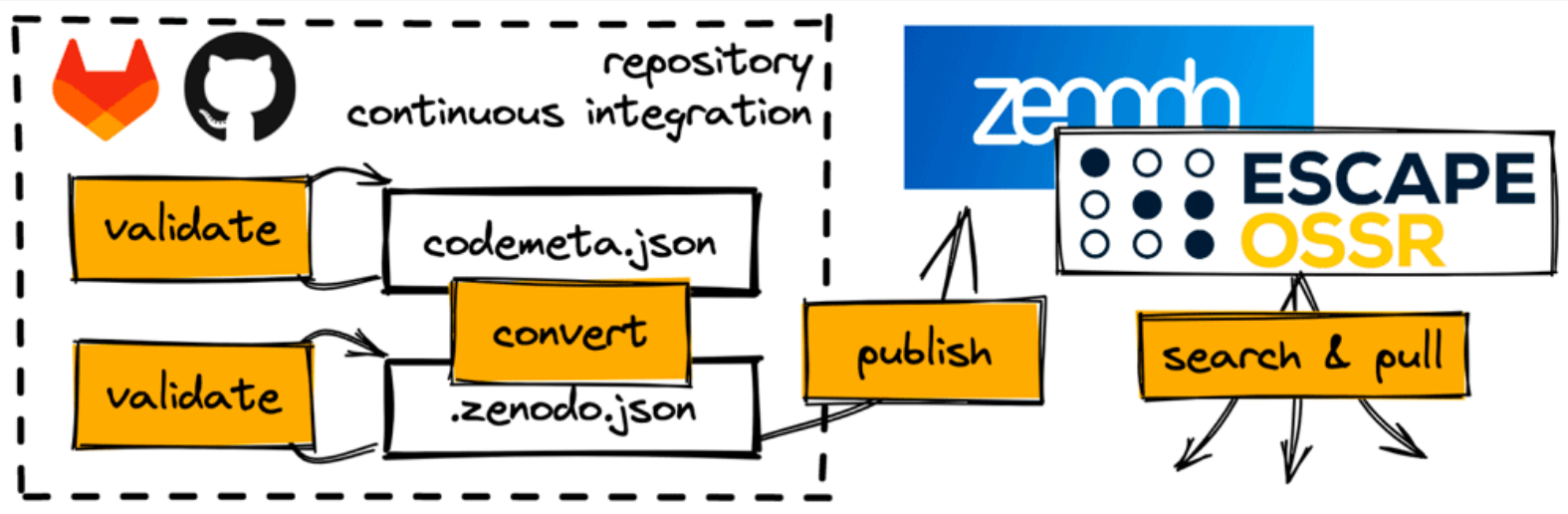
- Universal metadata schema to describe software
- Not limited or linked to a specific service
- Increasing adoption
- Integration with other services

➔ A `codemeta.json` file with a number of required keys is mandatory to submit software to the OSSR. The file comes with the source code, at the root of the repository.

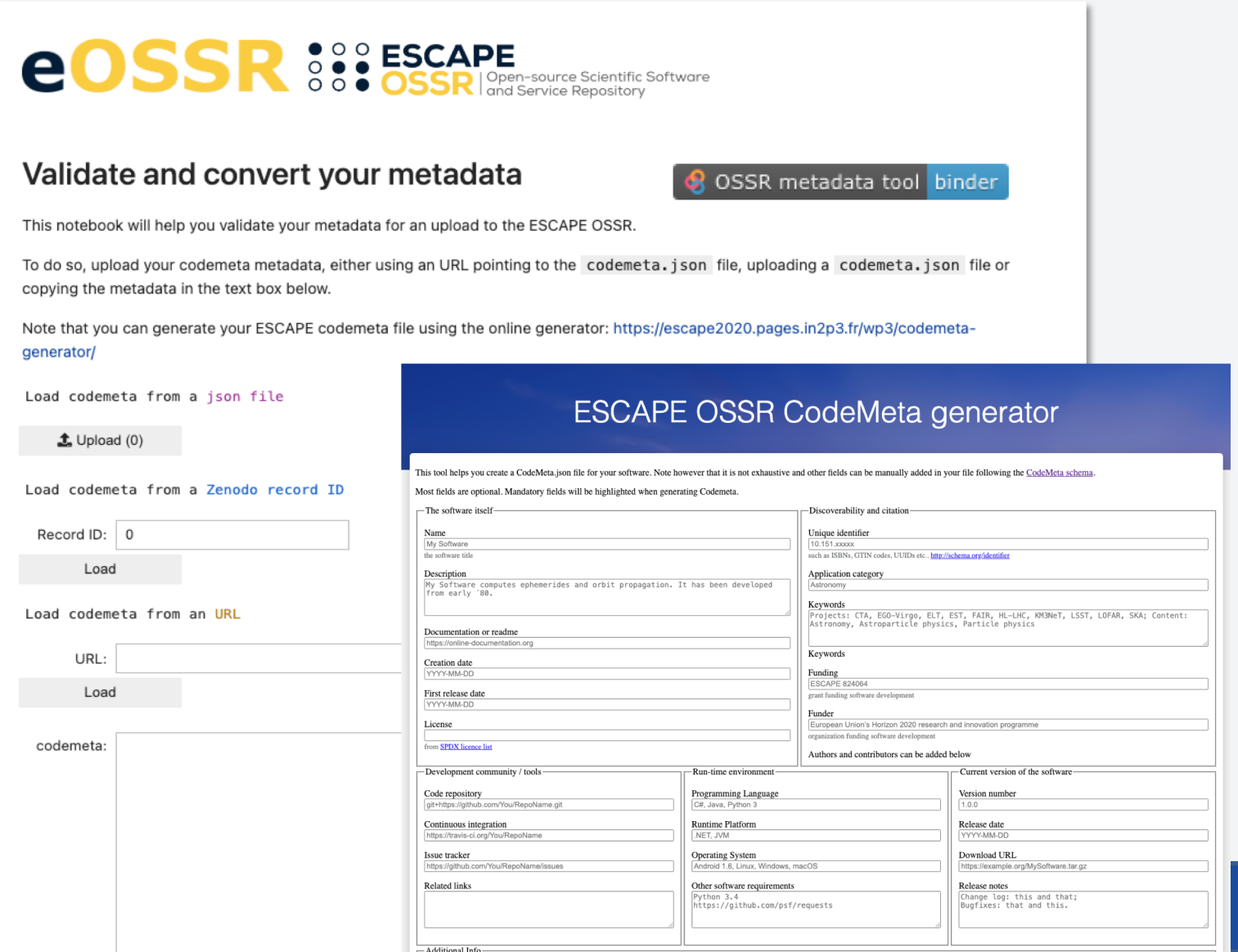
eOSSR

DOI 10.5281/zenodo.5524912 License MIT gitlab eossr

- The eOSSR is the OSSR Python library
  - Connects to Zenodo API to handle:
    - records: search, download, upload, publish, submit…
    - communities: list records, list and handle submissions
  - Handles OSSR metadata:
    - Defines required one
    - Converts from CodeMeta to Zenodo schema
    - Validates `codemeta.json` file

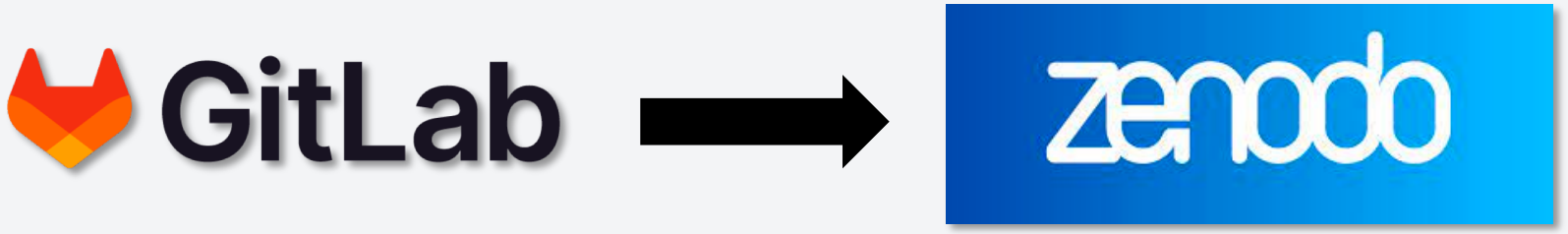


Online tools: metadata generator, converter & validator



- Help software developers to provide valid and complete metadata
- Get that first working version of `codemeta.json`
- Test things out

Gitlab to Zenodo

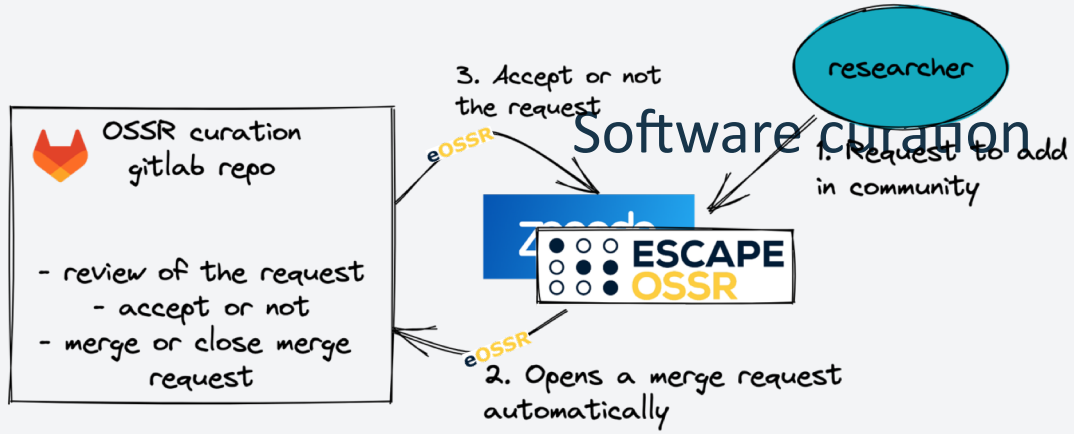


- Zenodo has an efficient GitHub integration, but no GitLab integration
- Many ESFRIs use their own Gitlab instance

➔We provide a simple gitlab-ci snippet

- to publish your software to Zenodo / OSSR, e.g. when making a release in gitlab
- using metadata provided in `codemeta.json`

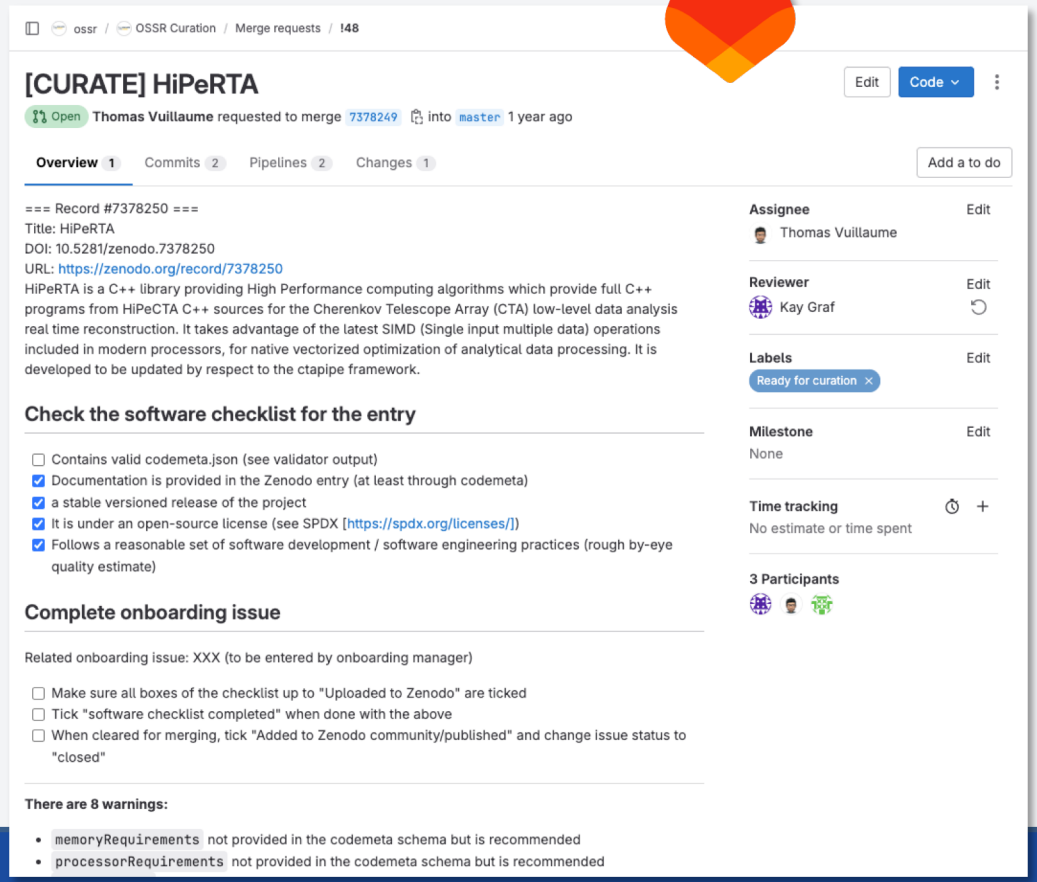
Software curation



- The OSSR is a curated software repository
  - implementation of the FAIR principles
  - good code practices
  - software quality
  - do not review scientific results → science paper

- Curation happens in a dedicated gitlab repository
  - completely open
  - automated checks
  - discussion between reviewers and providers

- Curation provides
  - Trust in the repository and provided content
  - Recognition for software providers



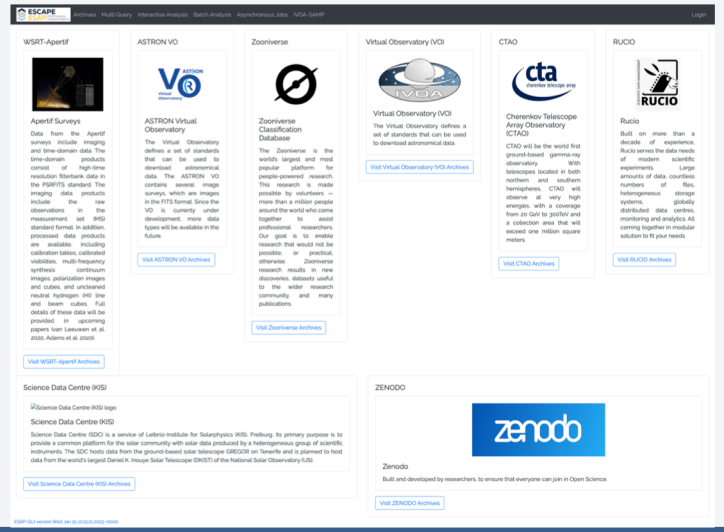
Integration with other services

ESCAPE ESAP | ESFRI Science Analysis Platform

connects to other services

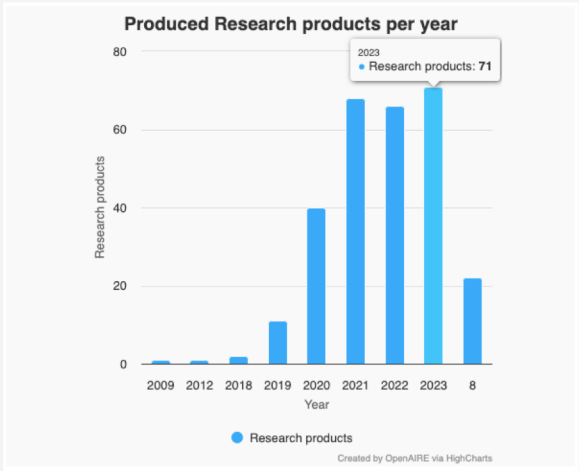
analyze data

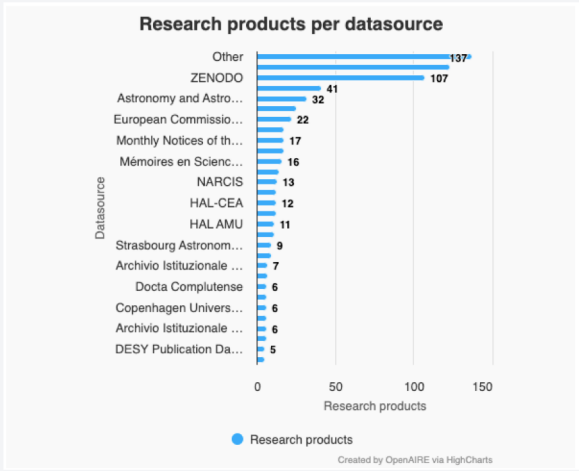
search and pull software from the OSSR



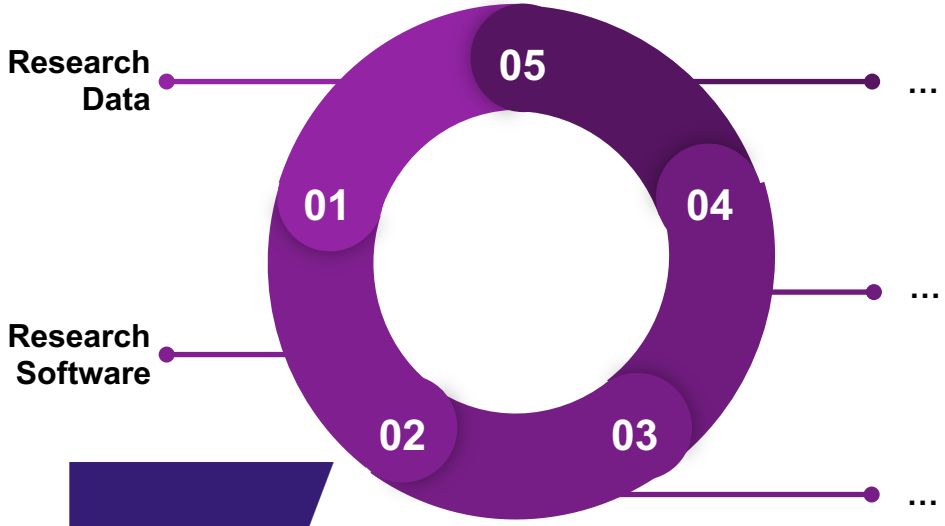
OpenAIRE | EXPLORE

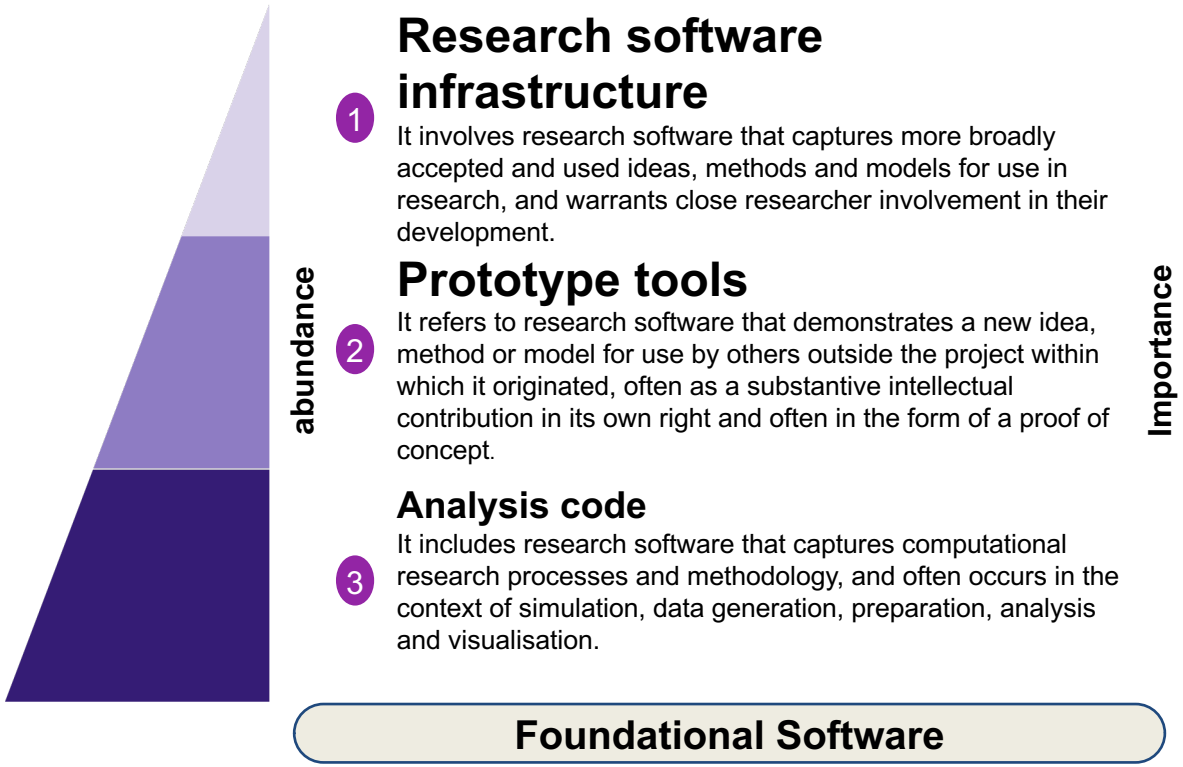
- EOSC integration
- Provides integrated statistics
- Connects with other data sources





Research Software as a first class citizen for the scientific endeavours





1. **Research software infrastructure**
   It involves research software that captures more broadly accepted and used ideas, methods and models for use in research, and warrants close researcher involvement in their development.
2. **Prototype tools**
   It refers to research software that demonstrates a new idea, method or model for use by others outside the project within which it originated, often as a substantive intellectual contribution in its own right and often in the form of a proof of concept.
3. **Analysis code**
   It includes research software that captures computational research processes and methodology, and often occurs in the context of simulation, data generation, preparation, analysis and visualisation.

Foundational Software

Not all software has the same level of importance