



Software Heritage

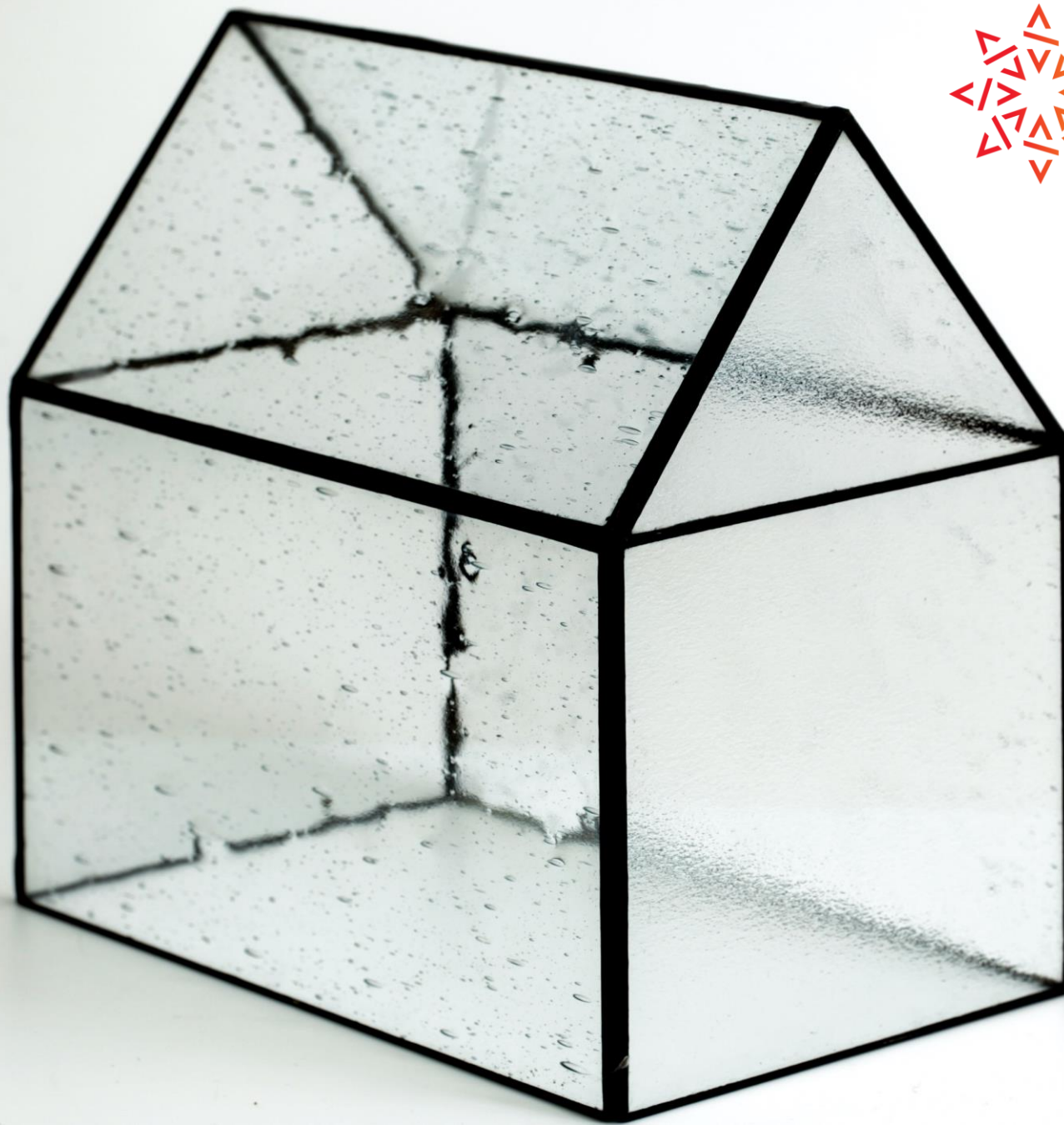
THE GREAT LIBRARY OF SOURCE CODE

Software Heritage :
Préserver le code
source pour une
recherche plus
transparente

MédiaNormandie | 2023-05-12

« Accompagner la préservation et
la diffusion des logiciels dans les
établissements »

sabrina.granger@inria



A propos de Software Heritage (SWH)

Initiative à but non-lucratif

Fondateurs : Roberto di Cosmo, Stefano Zacchiroli

<https://www.softwareheritage.org/mission/approach/>



Collect
Preserve
Share

2016 : ouverture au public de SWH

2017 : accord Inria et Unesco

<https://www.softwareheritage.org/2017/04/04/raising-awareness-for-software-source-code-preservation-and-access/>

2019 : The Paris Call : le code source reconnu comme **patrimoine culturel** <https://en.unesco.org/foss/paris-call-software-source-code>

<https://www.ouvrirlascience.fr/paris-call-software-source-code-as-heritage-for-sustainable-development/>

<https://meshedinsights.com/2019/02/26/software-as-cultural-artefact/>

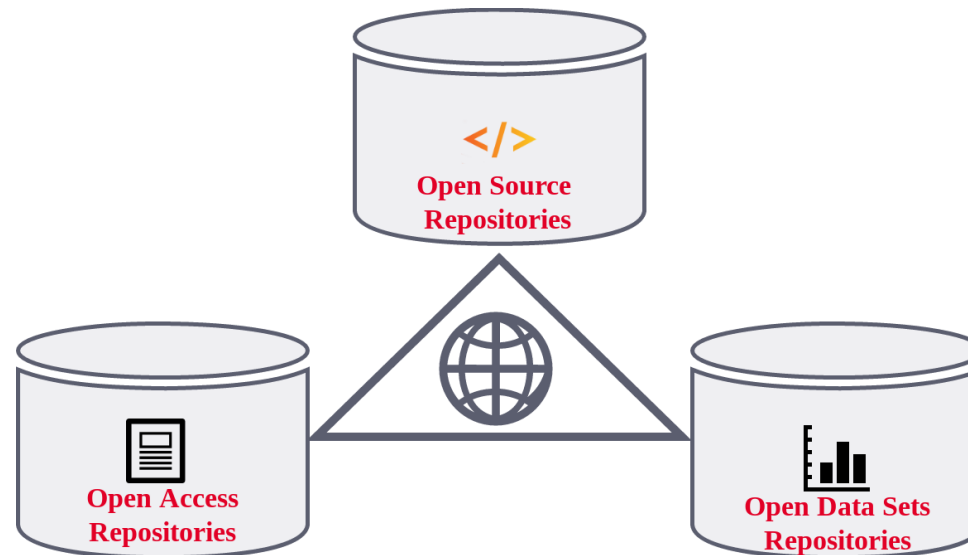


Pourquoi une bibliothèque
dédiée au code source est-elle
indispensable ?

Plus que jamais, les scientifiques ont besoin d'accéder au code source des logiciels, ...

“Digital data are **collected, created, interpreted, and disseminated** via software.
If you don't have the software, you don't have the data”

[Turello, Dan. « How to Think About Data: A Conversation with Christine Borgman | Insights: Scholarly Work at the John W. Kluge Center »](#). Webpage, 7 février 2019.



Quels enjeux scientifiques ?
La panacée existe-t-elle ?

... mais le code source est dispersé et fragilisé par des solutions non pérennes



Fig. 1 : La migration de projet



Les plateformes de développement ne sont pas éternelles



Constituer les fonds de Software Heritage

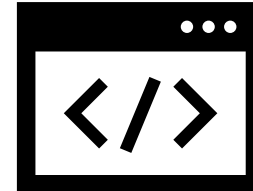
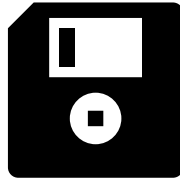
Quels types de contenus ? (1/3)

Que trouve-t-on dans la bibliothèque ?

©Dereck Hard

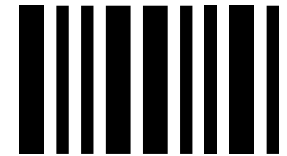


**Le YODA TEST
passer tu dois**



```
# import des données
data_fn=r'./results-
survey398164.xlsx'
df = pd.read_excel(data_fn)
# on supprime les colonnes
inutiles
cols = [c for c in df.columns
if (c[:10] != 'Durée pour'
and c[:11] !='Temps total')]
```

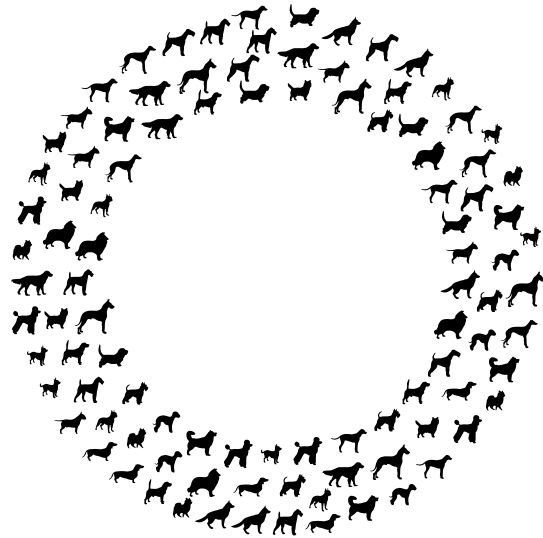
```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40
00
4004ef: b8 00 00 00
00
4004f4: e8 c7 fe ff ff
```



Quels types de contenus ? (2/3)

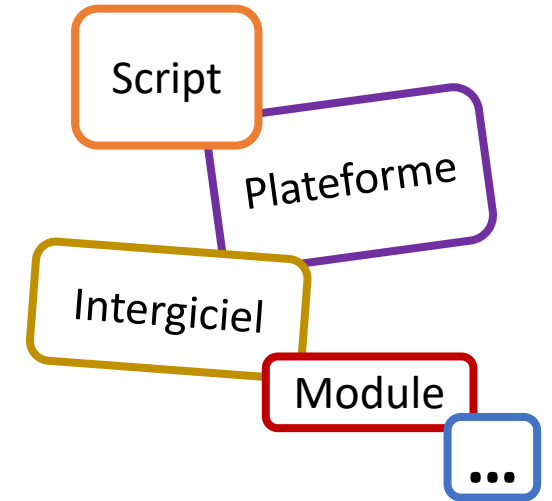
Typologie

CHIEN =



©Karen Arnold

LOGICIEL =



« [Le logiciel de recherche, un pilier de la recherche scientifique ouverte](#) », Comité pour la science ouverte Akhmerov, A., Bazuine, M., Beardsell, J., van den Bogerd, R., Branchett, S., Dunning, A., Keijzer-de Ruijter, M., Cruz, M., Martinez-Lavanchy, P., Schenk, M., Spaargaren, M., & Teperek, M. (2021). *TU Delft Research Software Policy*. <https://doi.org/10.5281/zenodo.4629662>

Quels types de contenus ? (3/3)

Quels critères de politique documentaire ?



Un logiciel stabilisé ?
OU
Un logiciel en cours de développement ?

Périmètre : ESR,
mais pas
seulement



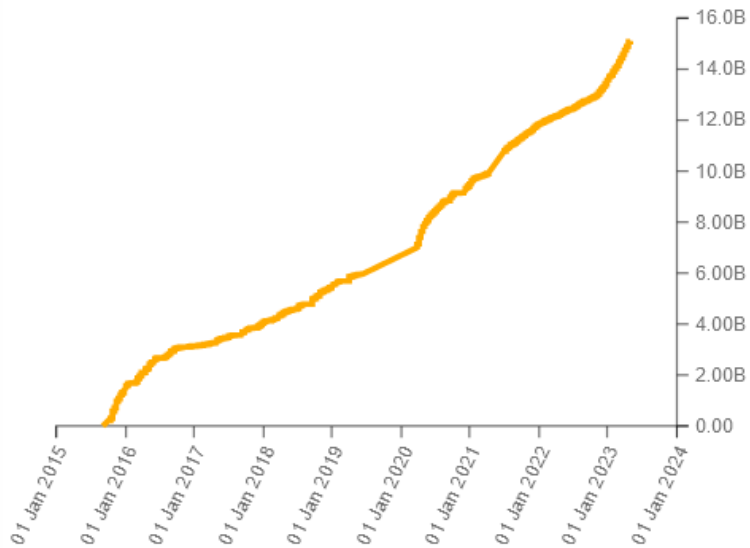
Un logiciel déjà reconnu ?
OU
Un logiciel qui n'a pas encore fait ses preuves ?

C/C:
un matériau au
comportement
très différent de
ce dont nous
avons l'habitude

Quel est le volume des collections SWH ?

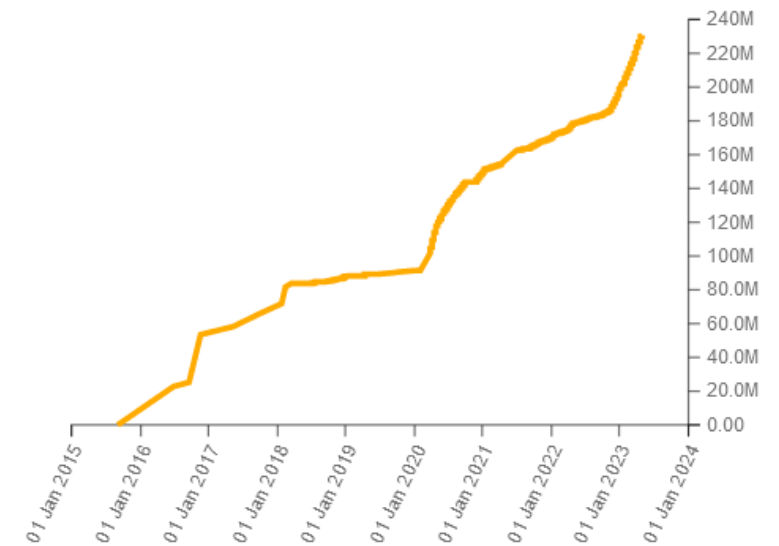
Source files

15 119 413 219



Projects

230 628 683



Des fichiers
uniques

Modalités d'alimentation de la bibliothèque



Des opérations de sauvetage

[En savoir + sur le sauvetage des contenus Gitorious](#)



Bitbucket
(BitBucket :Mercurial)

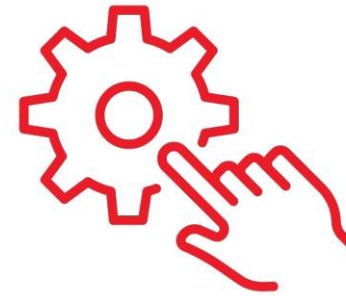


Des partenaires exportent leurs données



Guide dépôt

Guide modération



Du moissonnage automatisé

Bitbucket 2,176,467 origins	git 17,682 origins	R 22,103 origins
debian 132,115 origins	git 8,095 origins	GitHub 173,546,892 origins
GitLab 4,016,994 origins	Guix 13,108 origins	GNU 354 origins
heptapod 1,122 origins	launchpad 420,941 origins	Maven 130,839 origins
NixOS 13,108 origins	npm 2,025,743 origins	Maven 4,083 origins
Phabricator 201 origins	python 460,171 origins	SOURCEFORGE 345,169 origins

Voir si la forge de votre institution est archivée



De l'archivage sur mesure :

- Un projet
 - Une collection de projets (forge)
- Sans création de compte
Service accessible à tous,
pas besoin d'être l'auteur du code*

Des besoins scientifiques majeurs :
quelles réponses documentaires ?
Quels services proposer ?

Questions de reproductibilité : du domaine scientifique au domaine documentaire

Besoin scientifique :

Vérifier/Reproduire/Améliorer des résultats

Traduction documentaire :

- Pas de lien brisé !
- Identifier des ressources

Enjeux

« Est-ce fait correctement ? »
≠ « Est-ce la bonne chose à faire ? »

« Quel est le code source qui a été utilisé ? »



<https://osf.io/preprints/metaarxiv/sjyp3/>

Enjeux

Spécificités du logiciel (≠ données)

- Évolutivité
- Granularité (artefacts logiciels)



Photo de [Noyo creatif sur Unsplash](#)

Référencer et décrire : 2 enjeux différents

Référencer



Photo de [Immo Wegmann](#) sur [Unsplash](#)

=



Référencer un code source

Code source

```
fn handle_event(  
    &mut self,  
    mut _data: StateData<'_, GameData<'_, '_>>,  
    event: StateEvent,  
) -> SimpleTrans {  
    if let StateEvent::Window(event) = &event {  
        // Check if the window should be closed  
        if is_close_requested(&event) || is_key_down(&event, VirtualKeyCode::Escape) {  
            return Trans::Quit;  
        }  
    }  
}
```

Quelles différences avec d'autres types d'identifiants pérennes ? (ex. DOI)

SWHID du code source

[swh:1:cnt:8819df653b79a79ead8b0dd291e78c8547708e5f;origin=https://github.com/davidB/amethyst-starter-2d;visit=swh:1:snp:a137d958a3259c32e547d7d810feaabee1f3cfd1;anchor=swh:1:rev:a85801125959375c49738deb182d76999715d64e;path=/src/state.rs;lines=34-43](https://swid.world/1/snp:a137d958a3259c32e547d7d810feaabee1f3cfd1/anchor=swh:1:rev:a85801125959375c49738deb182d76999715d64e/path=/src/state.rs;lines=34-43)



```
10 use log::info;  
11  
12 pub struct MyState;  
13  
14 impl SimpleState for MyState {  
15     // On start will run when this state is initialized. For more  
16     // state lifecycle hooks, see:  
17     // https://book.amethyst.rs/stable/concepts/state.html#life-cycle  
18     fn on_start(&mut self, data: StateData<'_, GameData<'_, '_>>) {  
19         let world = data.world;  
20  
21         // Get the screen dimensions so we can initialize the camera and  
22         // place our sprites correctly later. We'll clone this since we'll  
23         // pass the world mutably to the following functions.  
24         let dimensions = world.read_resource::<ScreenDimensions>().clone();  
25  
26         // Place the camera  
27         init_camera(world, &dimensions);  
28  
29         // Load our sprites and display them  
30         let sprites = load_sprites(world);  
31         init_sprites(world, &sprites, &dimensions);  
32     }  
33  
34     fn handle_event(  
35         &mut self,  
36         mut _data: StateData<'_, GameData<'_, '_>>,  
37         event: StateEvent,  
38     ) -> SimpleTrans {  
39         if let StateEvent::Window(event) = &event {  
40             // Check if the window should be closed  
41             if is_close_requested(&event) || is_key_down(&event, VirtualKeyCode::Escape) {  
42                 return Trans::Quit;  
43             }  
44  
45             // Listen to any key events  
46             if let Some(event) = get_key(&event) {  
47                 info!("handling key event: {:?}", event);  
48             }  
49         }  
50     }  
51 }
```

Référencer et décrire : 2 enjeux différents

Décrire



=

Biographie	
Naissance	29 mai 1830 ✎ Vroncourt-la-Côte ✎
Décès	9 janvier 1905 ✎ (à 74 ans) Marseille ✎
Sépulture	Cimetière de Levallois-Perret (depuis le 22 janvier 1905) ✎
Nom de naissance	Louise Michel ✎
Pseudonymes	Enjolras, La Vierge Rouge ✎
Nationalité	française ✎
Activités	Enseignante, communarde, anarchiste, femme politique, franc- maçonne, journaliste, pédagogue, femme de lettres, poétesse, écrivaine ✎

Décrire un logiciel

README.md

Parmap in a nutshell

Parmap is a minimalistic library allowing to exploit multicore architecture for OCaml programs with minimal modifications: if you want to use your many cores to accelerate an operation which happens to be a map, fold or map/fold (map-reduce), just use Parmap's `parmap`, `parfold` and `parmapfold` primitives in place of the standard `List.map` and friends, and specify the number of subprocesses to use by the optional parameter `~ncores`.

See the `example` directory for a couple of running programs.

DO'S and DONT'S

Parmap is *not* meant to be a replacement for a full fledged implementation of parallelism skeletons (map, reduce, pipe, and the many others described in the scientific literature since the end of the 1980's, much earlier than the specific implementation by Google engineers that popularised them). It is meant, instead, to allow you to quickly leverage the idle processing power of your extra cores, when handling some head

The principle of parmap is very simple: write your map, fold, and mapfold, your O(n) operations, and each subprocesses (you choose the n), and each child processes the 1/n of the data, in chunks of a size you choose, through a shared memory area to the parent process. When all the children have terminated, and the d

AUTHORS

```
1 This code started as a two day exercise done at Irill in July 2011, by
2 Marco Danelutto <marcod@di.unipi.it> and Roberto Di Cosmo <roberto@dicosmo.org>,
3 that produced the architecture now contained in simplemapper.
4
5 It has evolved significantly after, and it now contains a particularly efficient
6 version of the map code on float arrays, incorporating ideas from Paul Vernaza
7 <pvernaza@andrew.cmu.edu>, and that works thanks to smart code and significant
8 insight from Jerome Vouillon <Jerome.Vouillon@pps.jussieu.fr>.
9
10 Pietro Abate <Pietro.Abate@pps.jussieu.fr> provided autoconf and ocamlbuild support.
11
12 Detailed contributor history is contained in the git repository available at
13
14 https://github.com/rdicosmo/parmap
15
```



2 modes
de dépôts

- SWHID
- source



Format du dépôt	Fichier
Type de document	Logiciel
Titre	en Parmap
Résumé	en Parmap is a minimalistic library allowing to exploit multicore architecture for OCaml programs with minimal modifications: if you want to use your many cores to accelerate an operation which happens to be a map, fold or map/fold (map-reduce), just use Parmap's parmap, parfold and parmapfold primitives in place of the standard List.map and friends, and specify the number of subprocesses to use by the optional parameter ~ncores.
Auteur(s)	Roberto Di Cosmo ^{1,2} , Marco Danelutto ³ 1 DGD-I - Direction générale déléguée à l'innovation (5000000000) • Institut National de Recherche en Informatique et en Automatique 2 - Software Heritage (498515) - France 3 - University of Pisa - Università di Pisa (366408) - Lur
Code Repository	https://github.com/rdicosmo/parmap.git
Date de production/écriture	2011-07-18
Version	1.2.4
Etat du développement	active
Licences	• GNU Library General Public License v2 only
Platform/OS	• Linux • MacOS
Domaine	• Informatique (cs)/Langage de programmation (cs.PL)
Langage de programmation	• OCaml
Classification ACM 2012	• acm2012_ACM2012/Computing methodologies/Parallel computing
Publications associées	• 10.1016/j.procs.2012.04.202
Mots-clés	en parallel computing, functional programming, OCaml
SWHID	swh:1:dir:ec88e5b901c034d5a91aa133e824d65cff3788a3;0 anchor=swh:1:rev:db44dc9cf7a6af7b56d8ebda8c75be3375

Cas d'usage : une personne qui utilise une plateforme de développement

DEVELOPPER

Plateforme collaborative (forge)

- ❑ AMELIORER la description du code en ajoutant des fichiers : README, AUTHORS, LICENSE, [CodeMeta](#)

ARCHIVER, REFERENCER

Software Heritage



- ❑ VERIFIER si le code est archivé :
 - Oui 🙌
 - Pas encore, besoin de MAJ ?
 - 👉 « Save code now » :
 - [Depuis SWH](#)
 - [Depuis sa plateforme](#)
 - [GitHub Action](#)
 - [API](#)
 - ❑ COPIER un [SWHID](#) pour pointer vers la version exacte d'un artefact (release, directory, fichier, lignes de code source)
- #Reproductibilité #Traçabilité

DECRIRE, CITER

HAL



- ❑ DEPOSER en copiant-collant un [SWHID](#)
#Authorship #AttribuerLesCrédits

- ❑ EXPORTER les MD de HAL pour citer un logiciel
#Authorship

Pour les utilisateurs de LaTeX

- ❑ INSTALLER le [biblatex-software](#) package

BESOINS RECHERCHE

OUTILS

SOLUTIONS DOCUMENTAIRES

Quelle complémentarité ?

- ✓ Articulation HAL et SWH
- ☐ Zenodo et SWH : différences ? 🤔

FAIR pour tous, tous pour FAIR?

Le cas des logiciels

Différences entre jeux de données et logiciels:

- Architecture
- Comportement

Principe FAIR (données)	Implications dans le domaine du logiciel ?
FINDABLE	Des métadonnées et des identifiants pérennes Enjeux de granularité et d'évolution
ACCESSIBLE	Archiver sur le long terme
INTEROPERABLE	= Echanger des données Importance des API (Une API permet d'exposer les fonctionnalités qu'on rend accessibles // menu d'un restaurant. Ex. de fonction d'API : appeler toutes les origines incluant le terme « X »)
REUSABLE	= 2 traductions possibles - Réutiliser = ré-exécuter - Ré-utiliser = modifier



Ligne d'arrivée :
open + high-quality + robust + FAIR + citable + sustained

Katz, D. (2021, June 28). FAIR is not the end goal. *Daniel S. Katz's Blog*.
<https://danielskatzblog.wordpress.com/2021/06/28/fair-is-not-the-end-goal/>

Ménager les
usagers :
gains/coûts ?

FAIR pour le
logiciel :
un point de
départ

Le programme des ambassadeurs

“**Policy** describes what is required, desired, and incentivised;
infrastructure determines what is possible;
but the **community** determines how things are done in practice.”

Brinkman, L., Schettino, A., Zurita-Milla, R., Eerland, A., Imming, M., Zormpa, E., Schwamm, H., Heininga, V. E., Tsang, E., & van 't Veer, A. (2020). *Open Science Community Starter Kit* (A. Blanchard, Trans.). International Network of Open Science & Scholarship Communities (INOSC). <https://osf.io/7vez3>

Le réseau des ambassadeurs en mai 2023



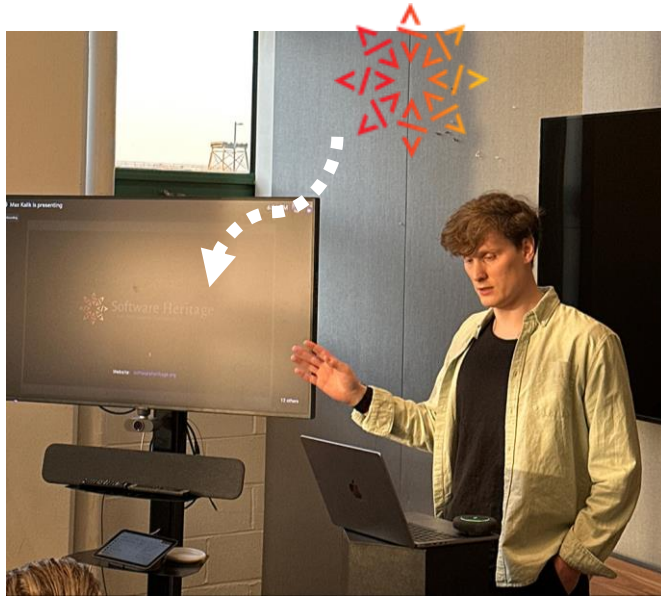
- 28 ambassadeurs
- Les profils représentés
- 4 domaines
- Un réseau international

<https://www.softwareheritage.org/2023/04/14/2023-community-workshop/>

Solliciter les ambassadeurs :
avantages,
démarche

Les modalités de
collaboration
ambassadeurs/SWH

Promouvoir SWH : les initiatives des ambassadeurs



© Max Kalik

Max Kalik au iOSoho meetup (2023)



“Services should assign PIDs to data descriptions, data and complementary materials (e.g., digital object identifiers (DOI)), **software (DOI, Software Heritage ID (SWHID)¹⁴**, authors (open researcher and contributor IDs (ORCID)) and associated research resources (RRIDs¹⁵).”

Sandström, Malin, Abrams, M., Bjaalie, J. G., Hicks, M., Kennedy, D. N., Kumar, A., Poline, J.-B., Roy, P. K., Tiesinga, P., Wachtler, T., & Goscinski, W. J. (2022). **Recommendations for repositories and scientific gateways from a neuroscience perspective**. *Scientific Data*, 9(1), Article 1. <https://doi.org/10.1038/s41597-022-01334-1>



Et aussi :
réseaux sociaux, formations,
groupes de travail,
échanges informels, etc.

Pour conclure

Quelques messages clés

Photo de [Wilhelm Gunkel](#) sur [Unsplash](#)



Logiciel \neq jeux de données

- Granularité
- Architecture
- Cycle de vie
- Des cas d'usage différents

Des solutions adaptées :

- SWHID
- Archive spécifique (\neq entrepôt généraliste)

Des collaborations

Une meilleure reconnaissance

Passer à l'action (liste non exhaustive, à classer selon vos priorités)

- Expliquer qu'une forge n'est pas une archive 🙋🙋
- Collaborer avec d'autres services pour identifier les forges de son institution
- Vérifier si les forges institutionnelles sont archivées sur SWH
- [Demander l'archivage d'une forge](#)
- Diffuser les [bonnes pratiques \(description, archivage\) pour les logiciels de recherche](#)
- [Promouvoir le dépôt de logiciel dans HAL : c'est simple comme un copier-coller avec le dépôt SWHID](#)
- Apprendre à modérer un dépôt logiciel dans HAL : hal.support@ccsd.cnrs.fr
- Faire intervenir les ambassadeurs SWH dans son établissement
- Exploiter le [guide « Codes et logiciels » du Comité pour la science ouverte](#) : formations internes, formations aux usagers, etc.
- Sensibiliser les décideurs au rôle du logiciel en recherche : feuille de route institutionnelle science ouverte ; ex. : [schéma directeur UGA](#) ; [Institut Pasteur](#))
- Mettre en œuvre les fonctionnalités dédiées au logiciel du Baromètre pour la Science Ouverte : [blog du CCSD](#)
- Diffuser [le tutoriel pour citer un logiciel](#) avec BibLaTeX
- Promouvoir la citation de logiciel dans les formations, au même titre que les autres productions académiques
- Expliquer la complémentarité et la différence entre outils, plateformes : ex. Zenodo et SWH, HAL et SWH, DOI et SWHID, etc.
- Diffuser et/ou produire des ressources sur les licences pour les logiciels ; ex. : [UGA](#) ; [site web ReUse](#) ; [diapos T. Gomez-Diaz](#)

Vos prochaines étapes ?

Par quoi souhaiteriez-vous commencer ?

Developing Guidelines for Metadata Collection and Curation for Research Software

10.30 - 11.30 CEST
23 May 2023 *Online*



FAIR-IMPACT
Expanding FAIR solutions across EOSC



FAIRCORE4EOSC
Core Components Supporting a FAIR EOSC

Une prochaine
étape clé en main :
Contribuer aux
préconisations
européennes

S'inscrire au webinaire :
<https://fair-impact.eu/events/fairimpact-events/developing-guidelines-metadata-collection-and-curation-research-software>



Une prochaine étape clé en main :

Participer à l'atelier 3 organisé lors des journées CasuHAL

« Software Heritage et HAL : deux plateformes, une mission - faciliter le dépôt de vos logiciels et codes source »

Sur place, à Toulouse

14-16 juin 2023

S'inscrire, voir le programme :

<https://casuhal2023.sciencesconf.org/>

Annexes

Liens à partager avec vos communautés : les collègues, les usagers, les décideurs

S'informer, échanger sur SWH

Archiver, référencer, décrire, citer : 4 enjeux majeurs pour les logiciels

Le millefeuille des vocabulaires pour décrire le logiciel

Des ressources sur les principes FAIR et le logiciel

Des alliés pour les data stewards : les research software engineers. Domaines d'intervention

Liens à partager avec vos communautés : les collègues, les usagers, les décideurs

Photo de [Nerene Grobler](#) sur [Unsplash](#)



- SWH en 1'40 : <https://youtu.be/opvGCXUqU4M>
- Faire découvrir les fonctionnalités :
<https://docs.softwareheritage.org/>
- Interroger SWH :
<https://docs.softwareheritage.org/#landing- browse>
- "Open Science tutorial: source code deposit"
#HAL :
<https://youtube.com/playlist?list=PLD2VqrZz2-u3bOWtoCoBlh5Flt6iYXsq3>

S'informer, échanger sur SWH

Photo de Jac Alexandru sur [Unsplash](#)



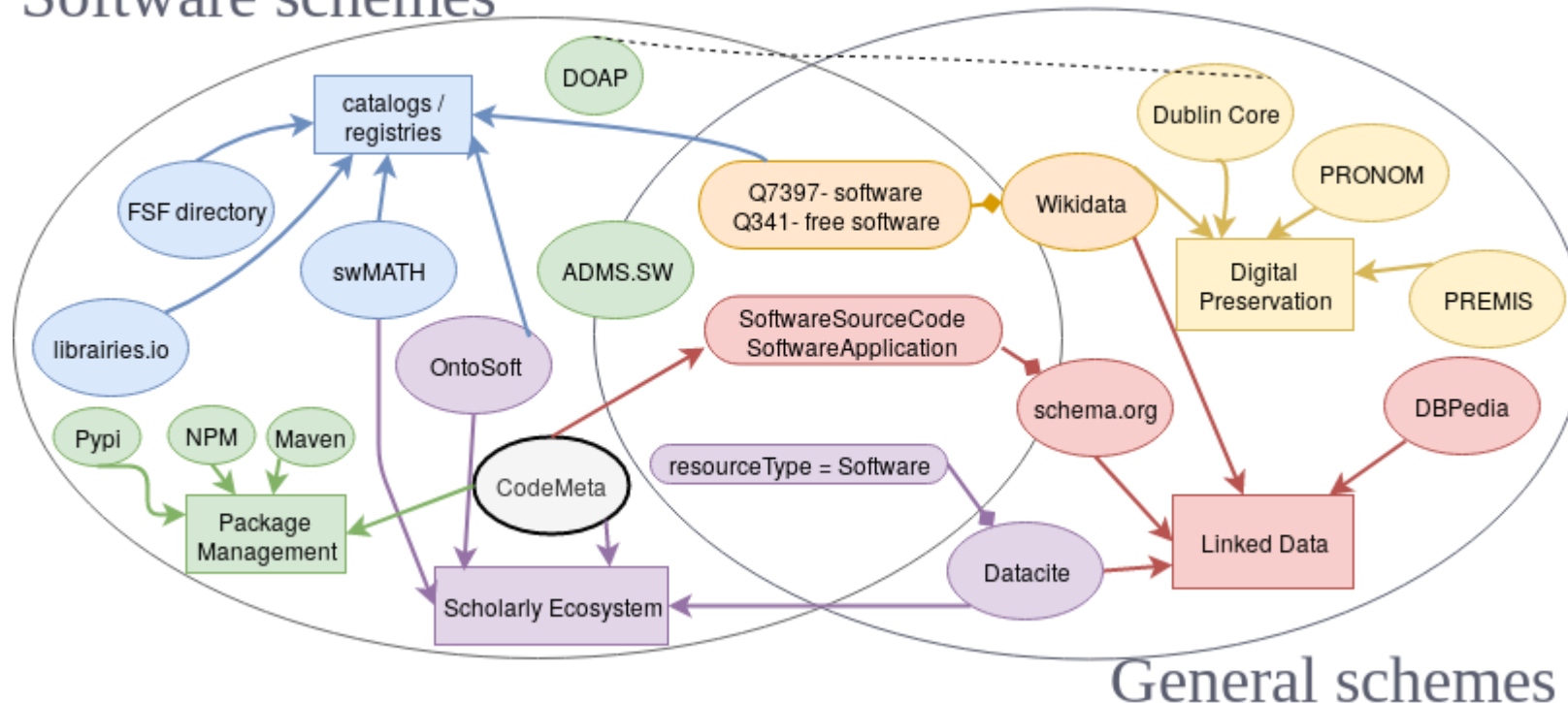
- <https://twitter.com/SWHeritage>
- <https://twitter.com/rdicosmo>
- <https://twitter.com/moraneottilia>
- <https://www.linkedin.com/company/swheritage/>
- <https://sympa.inria.fr/sympa/info/swh-science>
- <https://www.softwareheritage.org/newsletter/>
- <https://www.softwareheritage.org/blog>

Archiver, référencer, décrire, citer : 4 enjeux majeurs pour les logiciels

	POUR ...	AVEC ...
ARCHIVER	Rendre la recherche plus reproductible Aider à la vérification des résultats	- Des entrepôts académiques - SWH, la bibliothèque de code source
REFERENCER		- Des identifiants intrinsèques (SWHID) - Des identifiants extrinsèques (DOI)
DECRIRE	Favoriser la découverte des artefacts logiciels	- Des MD intrinsèques et extrinsèques - Des MD lisibles par les humains et les machines
CITER	Créditer les auteurs et les contributeurs de leurs travaux	- Une taxonomie des types de contribution (cf. rôles dans HAL, dépôt logiciel) - Une adaptation des pratiques de citation

Le millefeuille des vocabulaires pour décrire le logiciel

Software schemes



“As yet, there is no single ontology or metadata vocabulary for software description, or indeed agreed-upon taxonomies for these and other possible sets of criteria.”

Morrissey, S. M. (2020). *Preserving Software: Motivations, Challenges and Approaches*. Digital Preservation Coalition.

<https://doi.org/10.7207/twgn20-02>

[CodeMeta](#), un pivot entre les vocabulaires

Des ressources sur les principes FAIR et le logiciel

- Katz, D. S., Barker, M., Chue Hong, N. P., Garcia-Castro, L. J., Gruenpeter, M., Harrow, J., Martinez, C., Martinez, P. A., & Psomopoulos, F. E. (2022, March 9). *The Overlap Between FAIR for Research Software and Open Science*. <https://doi.org/10.5281/zenodo.6340732>
- Katz, D. S., Gruenpeter, M., & Honeyman, T. (2021). Taking a fresh look at FAIR for research software. *Patterns*, 2(3), 100222. <https://doi.org/10.1016/j.patter.2021.100222>
- Gruenpeter, M. & RDA France (Directors). (2021, June 28). *FAIR for Research Software*. <https://www.canal-u.tv/chaines/rda/fair-for-research-software-0>

N'agissez pas seul.e ! (1/2)

Cartographie des acteurs et outils impliqués dans le logiciel de recherche



Research Data Alliance/FORCE11 Software Source Code Identification WG, Allen, A., Bandrowski, A., Chan, P., di Cosmo, R., Fenner, M., Garcia, L., Gruenpeter, M., Jones, C. M., Katz, D. S., Kunze, J., Schubotz, M., & Todorov, I. T. (2020). *Software Source Code Identification Use cases and identifier schemes for persistent software source code identification*. <https://doi.org/10.15497/RDA00053>

N'agissez pas seul.e ! (2/2)

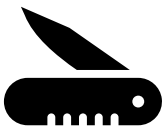
Zoom sur les research software engineers (RSE), des alliés pour les data stewards



IDENTITE

“Fundamentally, RSEs build software to support scientific research. They generally don’t have research questions of their own — they develop the computer tools to help other people to do cool things.”

Woolston, C. (2022). Why science needs more research software engineers. *Nature*. <https://doi.org/10.1038/d41586-022-01516-2>



CHAMPS
D’ACTION

Area 1: research implementation | Area 2: research analysis | Area 3: software project management | Area 4: knowledge sharing | Area 5: system, software, and data maintenance and support

Jetten, M., Grootveld, M., Mordant, A., Jansen, M., Bloemers, M., Miedema, M., & Gelder, C. W. G. van. (2021). *Professionalising data stewardship in the Netherlands. Competences, training and education. Dutch roadmap towards national implementation of FAIR data stewardship*. Zenodo. <https://doi.org/10.5281/zenodo.4623713>



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Merci pour votre
attention !

<https://archive.softwareheritage.org/>

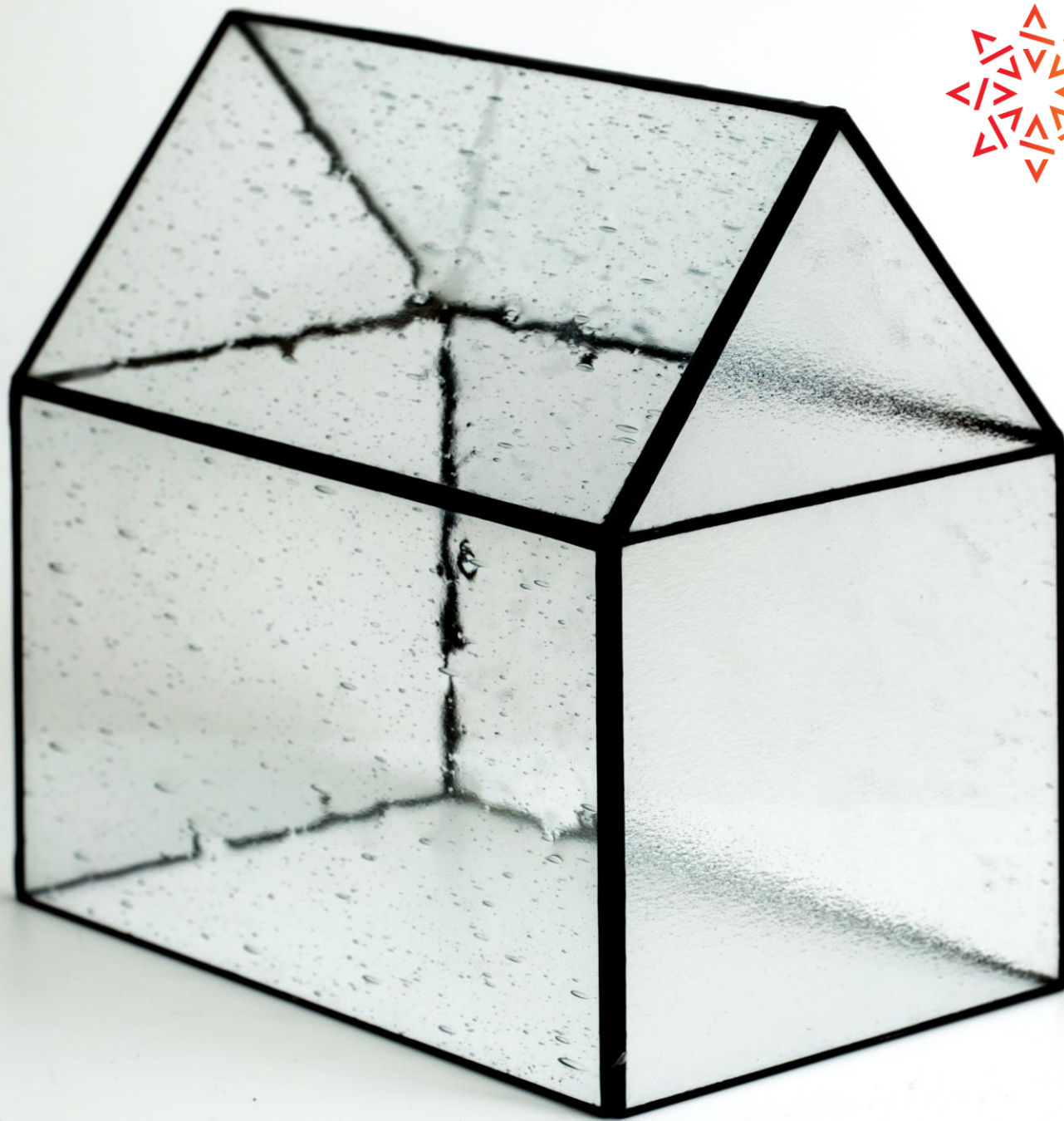


Photo de [Wilhelm Gunkel](#) sur [Unsplash](#)

