

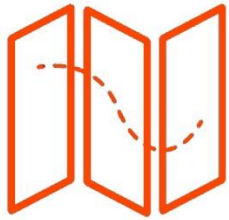
Software Heritage et HAL : deux plateformes, une mission

Faciliter le dépôt des codes sources de logiciels

Atelier proposé pour les Journées CasuHAL 2023

Hélène Bégnis – Sébastien Mazzaresse (CCSD)

Au programme



Atelier d'1h30

- 1 : Des éléments de contexte
- 2 : Les enjeux du partage de codes sources
- 3 : Les deux voies du dépôt
- 4 : Les grands principes de la modération
- 5 : Des cas d'usage



Comment ?

- Un quizz
- Quelques apports théoriques
- Une réflexion collective
- Des démos et une mise en application

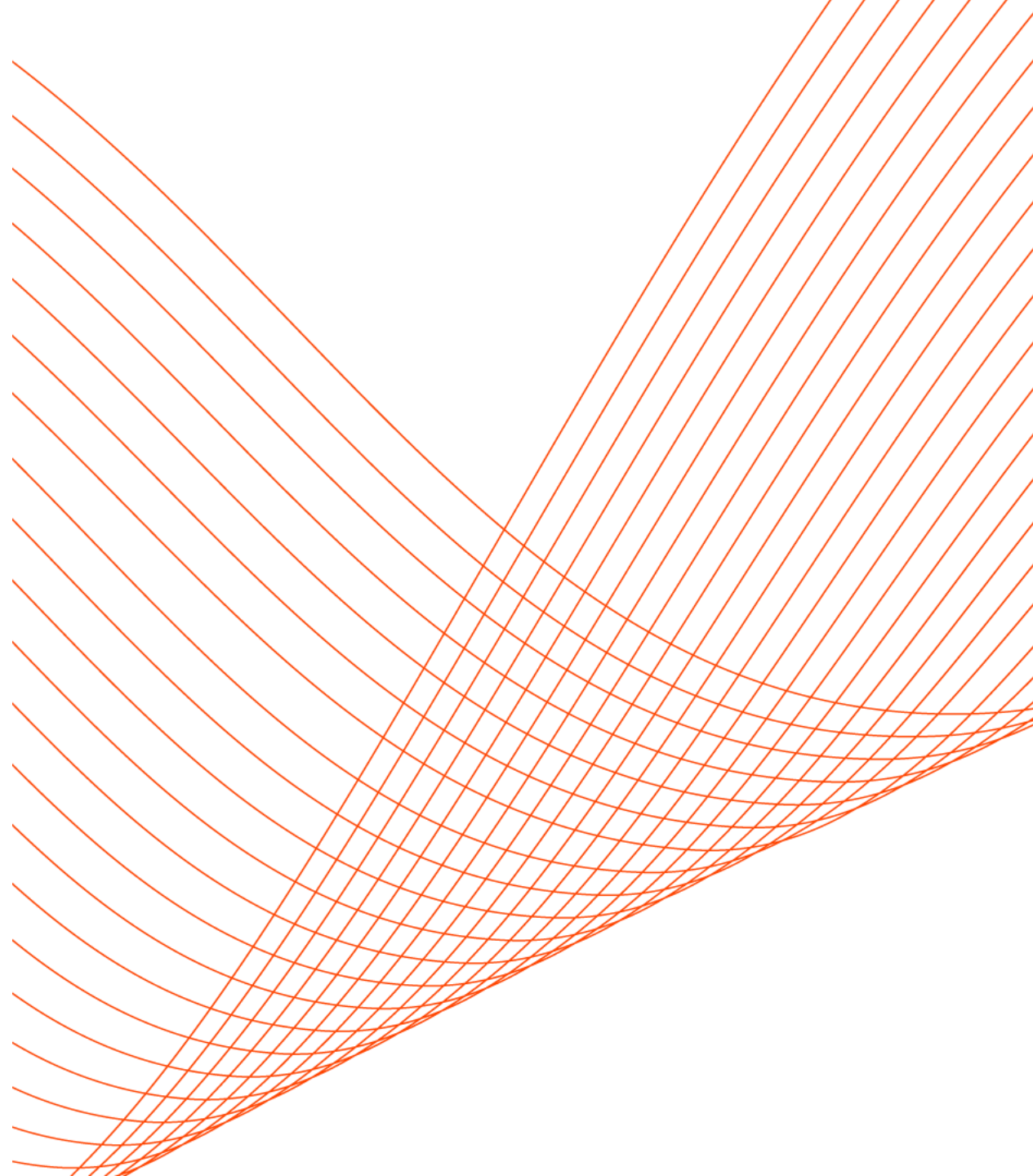


Objectif de l'atelier

- Être sensibilisé.e aux enjeux scientifiques et techniques du partage de codes sources
- Appréhender les différentes modalités du dépôt de logiciels dans HAL

01

Des éléments de contexte et de définition



Testons nos connaissances !

Réfléchissons ensemble à ce qu'est un code source de logiciel... et à ce qu'il n'est pas !

1. Qu'est-ce qu'un code source ? *

- ☐ Une copie du logiciel prête à être installée
- ☐ Les instructions compréhensibles par les humains qui décrivent le fonctionnement d'un logiciel
- ☐ Le nom donné à un logiciel open source

2. À quoi sert le code source d'un logiciel ? *

- ☐ À exécuter le logiciel sur un ordinateur
- ☐ À fournir une interface utilisateur conviviale
- ☐ À permettre aux développeurs de comprendre, modifier et améliorer le logiciel

3. Quelle est la différence entre le code source et le logiciel ? *

- ☐ Le code source est une version plus ancienne du logiciel
- ☐ Le code source est une partie du logiciel qui gère les fonctionnalités avancées
- ☐ Le code source est le texte contenant les instructions du logiciel, tandis que le logiciel est le produit exécutable obtenu après compilation.



5 min

Le logiciel issu de la recherche

Une nature hybride

Quels rôles ?

- Un **outil**

➔ Il permet le traitement de divers types de données

- Un **résultat** de recherche

➔ Il constitue en lui-même la solution à un problème donné

- Un **objet** de recherche

➔ Il est le « Quoi » sur lequel une communauté scientifique investigue

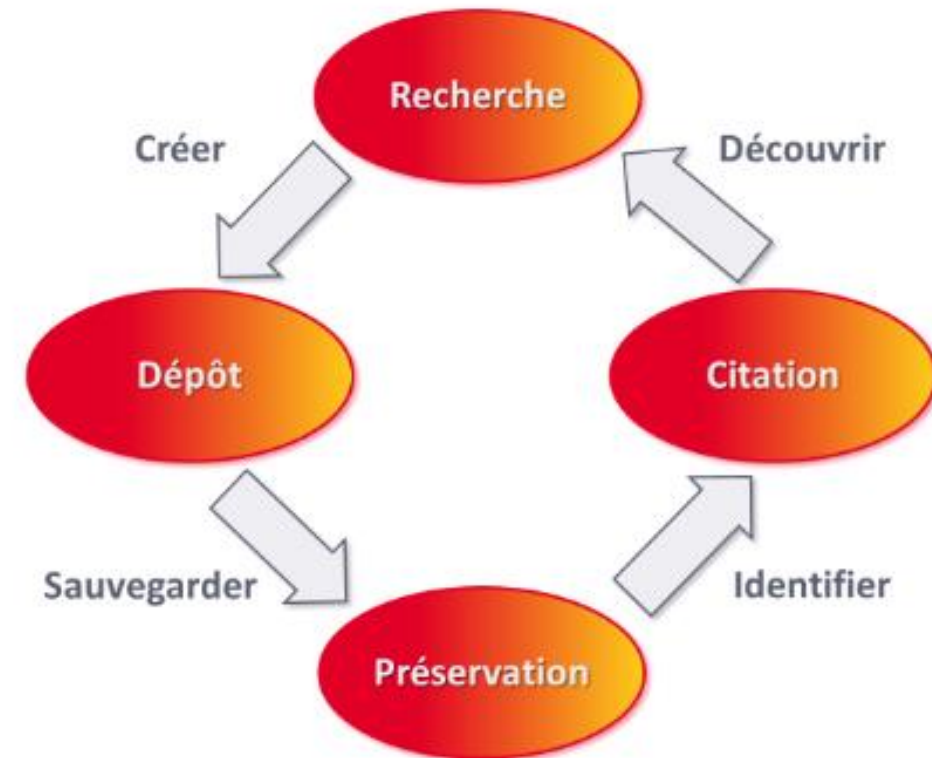
« Les scientifiques ne produisent plus seulement des articles de recherche synthétisant leurs résultats, mais aussi des logiciels venant en appui ou en démonstration de ceux-ci. »

Source : Mélanie Clément-Fontaine, Roberto Di Cosmo, Bastien Guerry, Patrick Moreau, François Pellegrini. Note d'opportunité sur la valorisation des logiciels issus de la recherche. [Rapport de recherche] Comité pour la science ouverte. 2019, 6 p. (<https://hal.science/hal-03606374>), consulté le 25/05/2023

Préserver et partager les logiciels : un enjeu scientifique

Une série de constats :

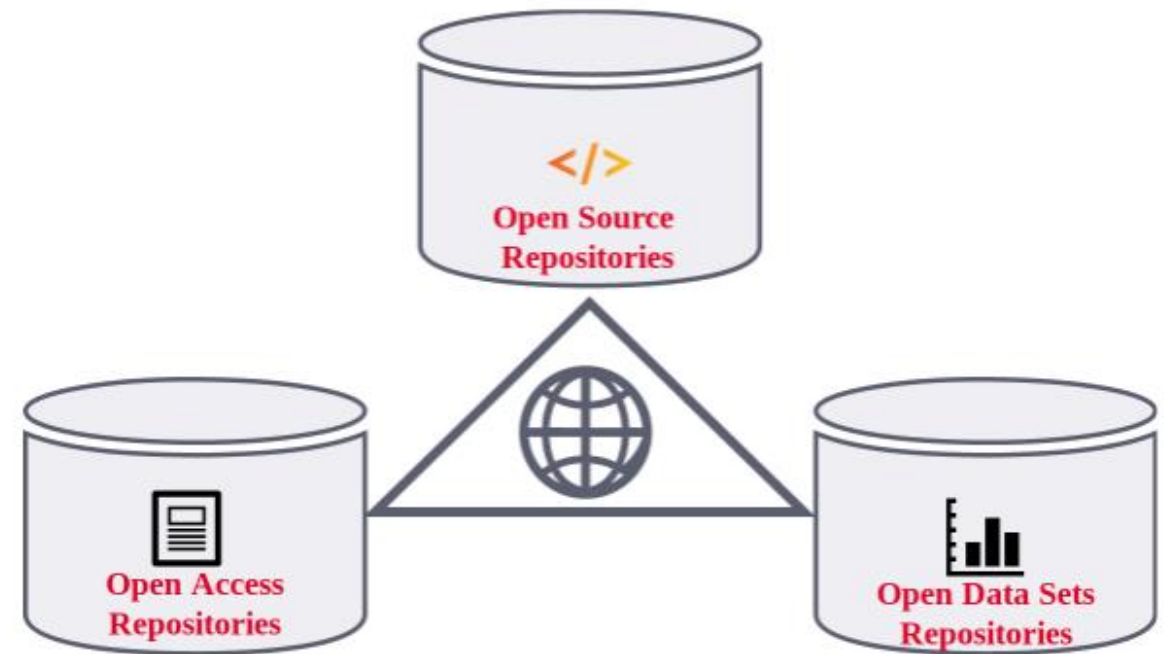
- Les logiciels sont devenus le support indissociable des connaissances techniques et scientifiques de l'humanité tout entière
- Ils constituent un véritable socle pour les sciences
- L'utilisation des logiciels intervient à toutes les étapes de la recherche, dans tous les domaines scientifiques
- La préservation des logiciels est à la base de la reproductibilité
- ➔ Offrir à la communauté scientifique les moyens de reproduire les conditions expérimentales ayant conduit à l'obtention de résultats



Source : Le cycle de vie du logiciel de recherche in Yannick Barborini, Roberto Di Cosmo, Antoine R. Dumont, Morane Gruenpeter, Bruno P Marmol, et al.. La création du nouveau type de dépôt scientifique - Le logiciel. JSO 2018 - 7es journées Science Ouverte Couperin : 100 % open access : initiatives pour une transition réussie, Jan 2018, Paris, France, 2018. <https://hal.science/hal-01688726>, consulté le 25/05/2023

Préserver et partager les logiciels : un enjeu scientifique

- Dans la quête pour la reproductibilité des résultats scientifiques et la transmission des connaissances scientifiques, trois piliers principaux :
 - o des **articles** scientifiques décrivant les résultats
 - o les ensembles de **données**
 - o les **logiciels** utilisés ou produits



Source : Roberto Di Cosmo, Morane Gruenpeter, Bruno P Marmol, Alain Monteil, Laurent Romary, et al.. Curated Archiving of Research Software Artifacts : lessons learned from the French open archive (HAL). IDCC 2020 - International Digital Curation Conference, Feb 2020, Dublin, Ireland. <10.2218/ijdc.v15i1.698>. <https://inria.hal.science/hal-02475835>, consulté le 25/05/2023

Objet

Organiser les modalités techniques et méthodologiques d'un nouveau type de dépôt dans HAL

Les acteurs

Software Heritage

CCSD

INRIA



Software Heritage



Les principales étapes

2017 : début de la collaboration

Mars 2018 : Phase de tests sur HAL-INRIA

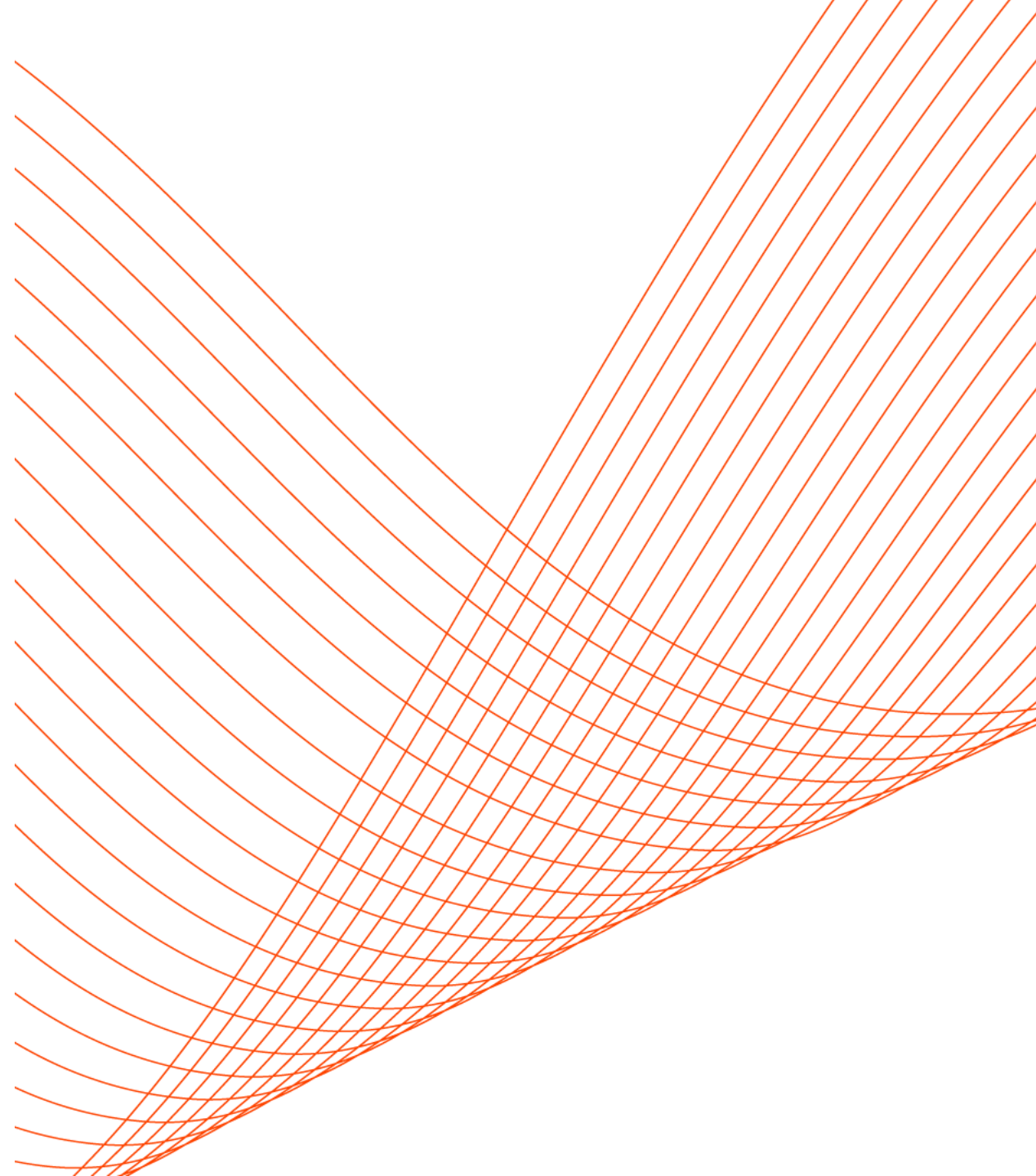
Septembre 2018 : Ouverture sur HAL

Avril 2020 : BibLaTeX @software

2023 : Nouvelle modalité de dépôt de logiciel et de code source dans HAL (Dépôt SWHID)

02

Enjeux du partage de codes sources dans HAL et SWH



Réflexion collective

Pourquoi partager des codes sources de logiciels dans HAL et Software Heritage ?

A quels besoins – scientifiques, techniques, documentaires – ce partage répond-il ?



10 min



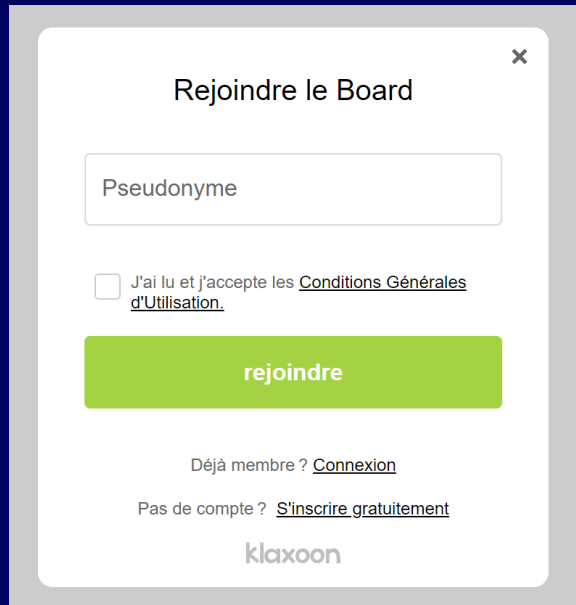
Seul ou en binômes



Klaxoon et restitution à l'oral

Se connecter au Board Klaxoon

Étape 1



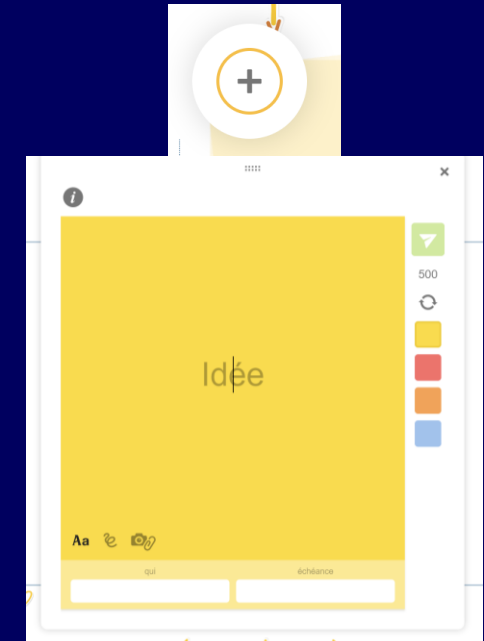
Saisissez un pseudonyme,
cochez la case des CGU
Et cliquez sur « rejoindre »

Étape 2



Cliquez sur « continuer »

Étape 3



Cliquez sur l'icône « + » pour ajouter de
nouvelles idées
Ajoutez-en autant que vous voulez !

Pourquoi partager dans HAL et Software Heritage ?

Les quatre piliers du partage de code source dans un contexte de science ouverte

Archiver

Garantir une préservation à long terme des logiciels de recherche

Référencer / Identifier

Assurer une identification sans ambiguïté :

- des artefacts logiciels (éventuellement dans leur contexte)
- des métadonnées associées

Décrire

Soutenir la découvrabilité des artefacts logiciels

- Métadonnées, vocabulaires et ontologies
- Outils pour créer, éditer, valider et convertir les métadonnées
- Répertoires pour stocker les métadonnées

Créditer

Créditer les auteurs de logiciels de recherche

Archiver

Préservation à long terme du code source

Enjeux

- Pérennité de l'archivage du code source sur SWH
- Garantie d'accès à long terme aux dépôts sur HAL (notices, identifiants, métadonnées, citation) et au code source sur SWH

➔ La préservation du code source est au fondement de la reproductibilité, de la transmission des connaissances aux générations futures et de leur réutilisation

Référencer

Identification du code source et de sa notice descriptive

Enjeux

- Identifier tous les artefacts numériques issus du processus de développement logiciel : un fragment de code, la version précise d'un projet, une révision...
 - Référencer dans HAL la notice descriptive d'un projet logiciel de recherche
- ➔ L'identification des objets logiciels est nécessaire pour garantir une traçabilité pérenne à travers les futures évolutions du développement logiciel et pour améliorer leur citabilité

Référencer

Le SWHID et le HAL-ID

Le SWHID

- Attribué par SWH
- Identifie un objet logiciel
- Est une empreinte digitale d'une version spécifique du logiciel

Objectifs : Spécifier - Archiver

Le HAL-ID

- Attribué par HAL
- Identifie un dépôt dans HAL
- Permet d'accéder aux métadonnées du projet logiciel

Objectifs : Référencer - Créditer

Décrire

Description du logiciel avec des métadonnées de qualité

Enjeux

Des métadonnées pour bien décrire le logiciel sont disponibles sur le formulaire de dépôt HAL et sont préservées avec le logiciel dans l'archive SWH.

- Métadonnées **obligatoires** (Nom, Auteurs, Domaine, Licence)
- Métadonnées **générales** (Identifiants, Mots clés, Financements, Date de production...)
- Métadonnées **spécifiques** (Langage de programmation, Code Repository, OS, Version...)
- Avant mise en ligne, la **cohérence** et l'**exactitude** des métadonnées est vérifiée par des documentalistes

➔ Assurer la **qualité** des métadonnées et une grande **visibilité** aux logiciels dans une démarche de science ouverte

Citer

Citation complète avec des métadonnées contrôlées dans différents formats

Enjeux

On peut citer un logiciel même s'il n'est pas déposé dans HAL, mais les services de HAL facilitent cet usage pour les publiants :

- La citation contient des métadonnées obligatoires soumises avec le logiciel ainsi que les identifiants SWHID et HAL-ID
- HAL propose différents formats d'export (BibTeX, TEI, codemeta.json, Dublin Core...)
- Le suivi de la production logicielle est facilité : pages CV, exports automatiques pour les rapports d'activité, pages web HAL, etc.
- L'ensemble des auteurs sont crédités

➔ La citation est indispensable pour promouvoir le rôle du logiciel dans la recherche et s'assurer que les développeurs de logiciels de recherche voient leurs contributions reconnues

Le SWHID et le HAL-ID composent la citation

Le SWHID fourni par SWH

To reference or cite the objects present in the Software Heritage archive, permalinks based on SoftWare Heritage p

Select below a type of object currently browsed in order to display its associated SWHID and permalink.

☒ directory
 ☐ revision
 ☐ snapshot

☒ archived repository
 ☐ archived `swh:1:dir:60866808c3fe54782fa93a8ebf7c849584b5e8ea`

```
swh:1:dir:60866808c3fe54782fa93a8ebf7c849584b5e8ea;
origin=https://github.com/frmichel/sparql-micro-service;
visit=swh:1:snp:a942f5053178ab76149aa4cfc3ff95b1d33a24f1;
anchor=swh:1:rev:b138e412f40a4add298d3dab32c106fffd86d62a
```

☒ Add contextual information

Sources : Software Heritage. Franck Michel. SPARQL Micro-Services. https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://github.com/frmichel/sparql-micro-service ; Franck Michel. SPARQL Micro-Services. 2018, <https://hal.science/hal-04115815>, consulté le 25/05/2023

Le format de citation dans HAL

Citer

Franck Michel. SPARQL Micro-Services. 2018, (swh:1:dir:60866808c3fe54782fa93a8ebf7c849584b5e8ea;origin=https://github.com/frmichel/sparql-micro-service;visit=swh:1:snp:a942f5053178ab76149aa4cfc3ff95b1d33a24f1;anchor=swh:1:rev:b138e412f40a4add298d3dab32c106fffd86d62a). (hal-04115815)

Export BibTeX dans HAL

```
@softwareversion{michel:hal-04115815v1,
  TITLE = {{SPARQL Micro-Services}},
  AUTHOR = {Michel, Franck},
  URL = {https://hal.science/hal-04115815},
  NOTE = {},
  YEAR = {2018},
  MONTH = Jun,
  DOI = {10.5281/zenodo.7756248},
  SWHID = {swh:1:dir:60866808c3fe54782fa93a8ebf7c849584b5e8ea;origin=https://github.com/frmichel/sparql-micro-service},
  VERSION = {0.5.5},
  REPOSITORY = {https://github.com/frmichel/sparql-micro-service.git},
  LICENSE = {https://spdx.org/licenses/Apache-2.0},
  KEYWORDS = {web API ; SPARQL ; RDF ; mapping ; mediator},
  HAL_ID = {hal-04115815},
  HAL_VERSION = {v1},
}
```

Valoriser

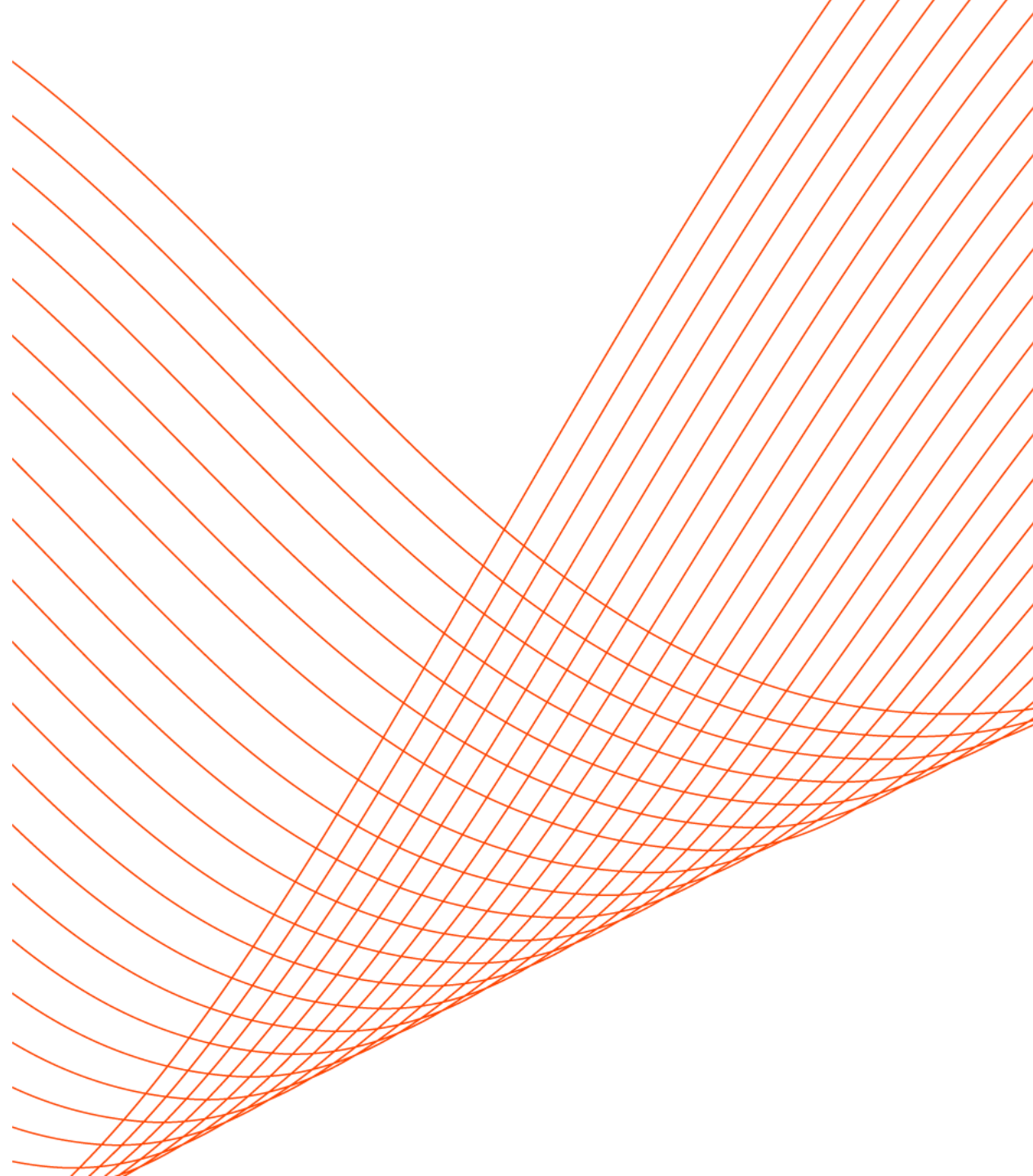
Promotion et accès ouvert à la production logicielle

Enjeux

- Accroître la visibilité de la production logicielle
- Partager les travaux dans des CV HAL, des rapports d'activité, des sites web d'unités, des collections...
- Mettre en relation Publications, Données et Codes

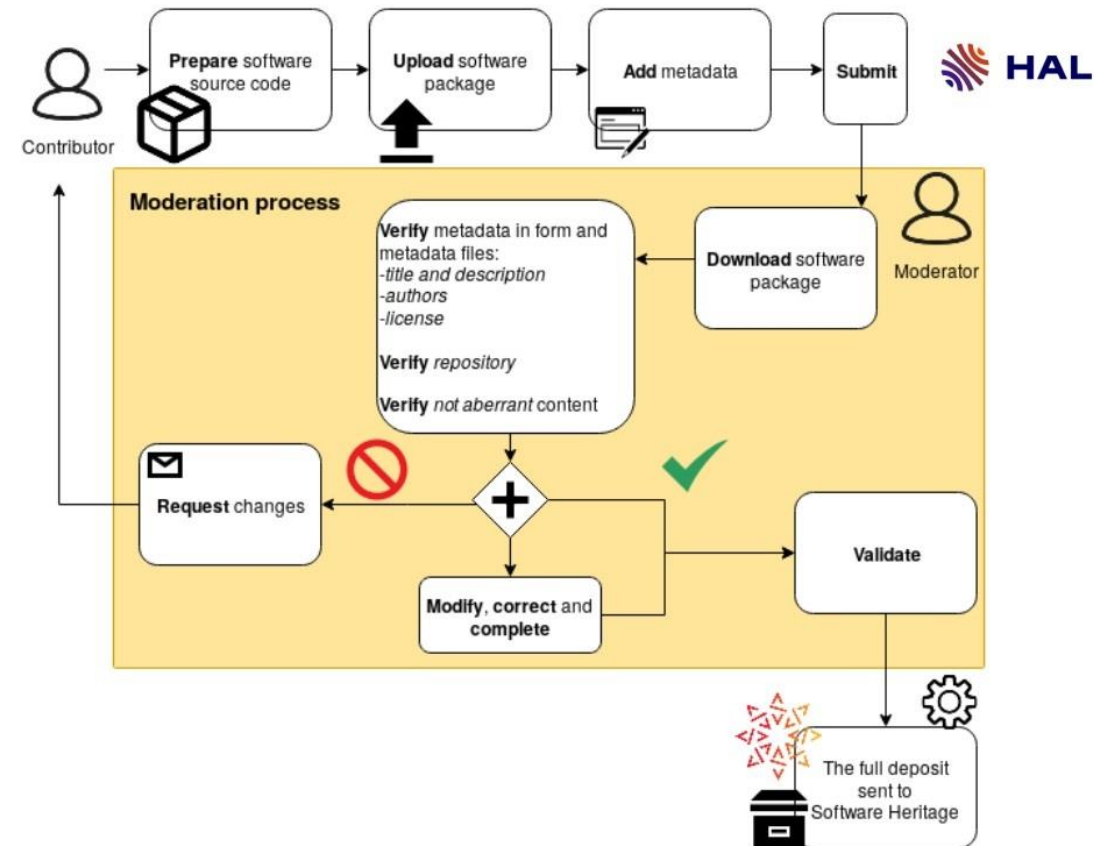
03

Les deux voies du dépôt dans HAL



Le dépôt, étape par étape

- En amont du dépôt
 - Préparation du code source
- Dépôt dans HAL
 - Choix de la méthode : Dépôt source / Dépôt SWHID
 - Ajout des métadonnées
- Validation du formulaire de dépôt par le déposant
- Dépôt en attente de modération
 - Dialogue entre déposant et modérateur
- Validation du dépôt par le modérateur
- Dépôt mis en ligne sur HAL et transfert vers SWH
- Export dans différents formats
 - codemeta.json, BibTeX, TEI, etc.



Source : Roberto DI COSMO, Morane GRUENPETER, Bruno MARMOL, et al. « Curated Archiving of Research Software Artifacts : Lessons Learned from the French Open Archive (HAL) », IJDC. 2020, vol. 15, no 1. En ligne : <https://doi.org/10.2218/ijdc.v15i1.698>, consulté le 25/05/2023

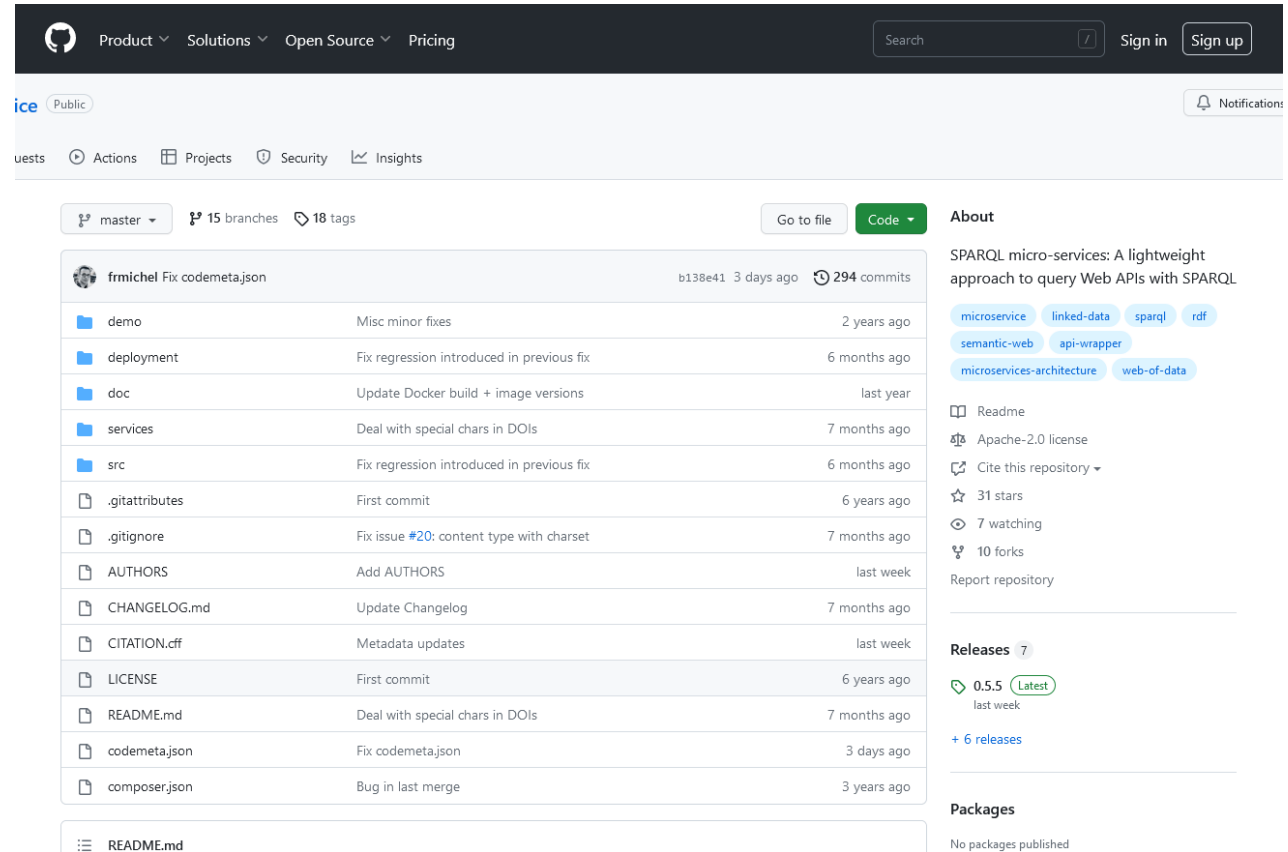
La préparation du code source

➤ Fichiers recommandés pour documenter le code source :

- README
- AUTHORS
- LICENSE
- codemeta.json

➤ Où ?

- Soit localement
- Soit sur un code repository (GitHub, GitLab, Bitbucket...)



The screenshot shows the GitHub interface for the repository 'frmichel/sparql-micro-service'. The repository is public and has 15 branches and 18 tags. The main branch is 'master'. The repository description is 'SPARQL micro-services: A lightweight approach to query Web APIs with SPARQL'. The repository has 31 stars, 7 watchers, and 10 forks. The latest release is 0.5.5, published last week. The repository includes a README, a LICENSE, and a codemeta.json file. The file list shows various folders and files, including 'demo', 'deployment', 'doc', 'services', 'src', '.gitattributes', '.gitignore', 'AUTHORS', 'CHANGELOG.md', 'CITATION.cff', 'LICENSE', 'README.md', 'codemeta.json', and 'composer.json'.

Source : Projet SPARQL micro-services: A lightweight approach to query Web APIs with SPARQL Franck Michel, <https://github.com/frmichel/sparql-micro-service>, consulté le 25/05/2023

Un README, c'est quoi ?

- WHAT ?

Un fichier texte qui présente et explique un projet

- WHY ?

Répondre aux questions sur la façon d'installer et d'utiliser un logiciel

- WHO ?

Toute personne travaillant sur un projet de programmation

- WHEN ?

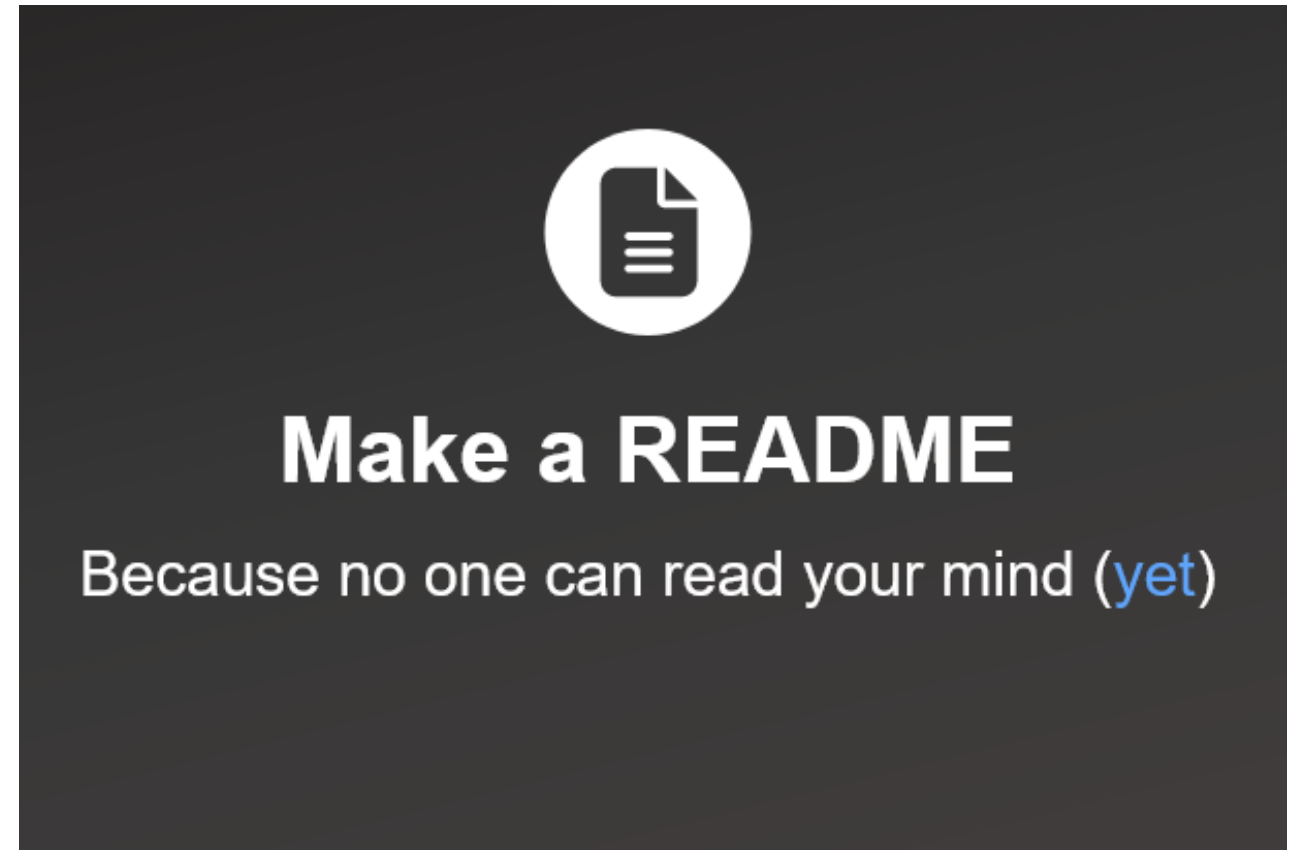
Avant de montrer un projet à d'autres personnes ou de le rendre public

- WHERE ?

Dans le répertoire de niveau supérieur du projet

- HOW ?

Dans n'importe quel format de fichier texte (le plus courant : Markdown)



Source : <https://www.makeareadme.com/>, consulté le 25/05/2023

Un README, quelles informations ?

Suggestions pour un « bon » README

- Nom du projet
- Descriptif du logiciel
- Environnement de développement
- Instructions d'installation et d'exécution du code
- Contact / Assistance
- Lien vers la documentation / Site web
- Usage possible du logiciel
- Statut du projet
- Historique des versions
- Feuille de route
- Visuels
- ...

```
README.md
1 # SPARQL Micro-Services
2
3 The SPARQL Micro-Service architecture [1, 3] is meant to allow the combination of Linked Data with data from Web APIs. It enables querying non-RDF Web APIs
4
5 Each SPARQL micro-service is a lightweight, dedicated SPARQL endpoint that typically provides access to a small, resource-centric graph. The graph produces
6
7 This project is a PHP implementation for JSON-based Web APIs. It comes with multiple configuration options to fit most specific APIs (e.g. add specific HTTP headers)
8
9 ## Examples and Demo
10
11 You can check out some services we published at [https://sparql-micro-services.org/](https://sparql-micro-services.org/).
12 An HTML documentation and test interface is generated dynamically from the micro-service description, that embeds http://schema.org/Dataset markup data to
13
14 A demo(http://sparql-micro-services.org/demo-sms?param=Delphinapterus+leucas) showcases the use of SPARQL micro-services to integrate, within a single SPARQL
15
16 This project comes with several example SPARQL micro-services, allowing for instance to search photos matching some tags on [Flickr](https://www.flickr.com/),
17 Other services are designed to query major biodiversity data sources such as the [GBIF](https://www.biodiversitylibrary.org/), [BHL](https://www.biodiversitylibrary.org/),
18 See the services available in this in [this repository](services/) as well as the [TaxrefWeb](https://github.com/frmichel/taxrefweb/tree/master/sparql-micro-services)
19
20
21
22 ## Documentation
23
24 - [Usage of SPARQL micro-services](doc/01-usage.md)
25 - [Configuration of a SPARQL micro-service](doc/02-config.md)
26 - [Installation, configuration and deployment](doc/04-install.md)
27 - [Docker deployment](deployment/docker/)
28 - [Dynamic HTML documentation](doc/03-html-doc.md)
29 - [Provenance information](doc/05-prov.md)
30
31
32
33 ## Typical use case
34
35 The query below illustrates a common usage of SPARQL micro-services that builds a mashup of Linked Data and data from Web APIs.
36 It first retrieves the URI of the common dolphin species (Delphinus delphis) from TAXREF-LD, a biodiversity RDF dataset [2]. Then, it enriches this descriptive
37
38 The SPARQL endpoint as well as the two SPARQL micro-services are invoked within dedicated SERVICE clauses.
39
40
```

Source : fichier README du projet SPARQL Micro-Services,
https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://github.com/frmichel/sparql-micro-service, consulté le 25/05/2023

Le fichier AUTHORS, quelles informations ?

- Le fichier AUTHORS identifie les personnes à créditer comme auteurs du code source
- Ces auteurs doivent être renseignés dans le dépôt HAL
 - Au moment de la modération, vérification de la cohérence des informations entre le fichier Authors et les métadonnées du dépôt HAL
- En complément des auteurs, d'autres contributeurs peuvent être mentionnés (par exemple, dans un fichier CONTRIBUTORS)
- Noms de fichiers attendus : AUTHORS, CONTRIBUTORS, CREDITS, CITATION

NB : Les personnes renseignées dans HAL et sur le fichier AUTHORS ont les droits de création sur le code source

AUTHORS

```
1 The wonderland team (1953-1960)
2 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
3 Captain Hook <c.hook@wonderland.org>
4 The Crocodile/Tic Toc <croco@wonderland.org>
5 Peter Pan <peter.pan@wonderland.org>
6 Wendy Darling <w.darling@wonderland.org>
7 Morane Gruenpeter <morane.gruenpeter@inria.fr>
8
9
```

Source : fichier AUTHORS du projet The wonderland App,
https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://github.com/moraneegg/deposit-template, consulté le 25/05/2023

Le fichier LICENSE, quelles informations ?

- Le fichier LICENSE décrit les droits d'utilisations du code source déposé
 - A choisir en collaboration avec les services de valorisation des organismes de rattachement des auteur
- Les informations concernant la licence doivent être renseignées dans le dépôt HAL
 - Au moment de la modération, vérification de la cohérence des informations entre le fichier LICENSE et les métadonnées du dépôt HAL

LICENSE	
1	GNU GENERAL PUBLIC LICENSE
2	Version 3, 29 June 2007
3	
4	Copyright (C) 2007 Free Software Foundation, Inc. < http://fsf.org/ >
5	Everyone is permitted to copy and distribute verbatim copies
6	of this license document, but changing it is not allowed.
7	
8	Preamble
9	
10	The GNU General Public License is a free, copyleft license for
11	software and other kinds of works.
12	
13	The licenses for most software and other practical works are designed
14	to take away your freedom to share and change the works. By contrast,
15	the GNU General Public License is intended to guarantee your freedom to
16	share and change all versions of a program--to make sure it remains free
17	software for all its users. We, the Free Software Foundation, use the
18	GNU General Public License for most of our software; it applies also to
19	any other work released this way by its authors. You can apply it to
20	your programs, too.
21	
22	When we speak of free software, we are referring to freedom, not
23	price. Our General Public Licenses are designed to make sure that you
24	have the freedom to distribute copies of free software (and charge for
25	them if you wish), that you receive source code or can get it if you
26	want it, that you can change the software or use pieces of it in new
27	free programs, and that you know you can do these things.
28	
29	To protect your rights, we need to prevent others from denying you
30	these rights or asking you to surrender the rights. Therefore, you have
31	certain responsibilities if you distribute copies of the software, or if
32	you modify it: responsibilities to respect the freedom of others.
33	

Source : fichier LICENSE du projet AffectationRO,
https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://github.com/moraneegg/AffectationRO, consulté le 25/05/2023

Le fichier codemeta.json ?

- Un schéma de métadonnées pour décrire spécifiquement les logiciels de recherche et leur code source
- Un vocabulaire pouvant être utilisé pour normaliser les échanges de métadonnées entre les organisations (archives, plateformes de développement, etc.)
- Au format JSON
- CodeMeta generator : un outil pour générer un fichier codemeta.json (<https://codemeta.github.io/codemeta-generator/>)

codemeta.json

```
{
  "@context": "https://doi.org/10.5063/schema/codemeta-2.0",
  "@type": "SoftwareSourceCode",
  "license": "https://spdx.org/licenses/AGPL-3.0",
  "codeRepository": "https://github.com/moranegg/AffectationRO",
  "dateCreated": "2014-01-01",
  "dateModified": "2019-06-26",
  "name": "AffectationRO- The assignment problem",
  "version": "1.0.0",
  "description": "A java implementation for The Assignment Problem  a distributed system as a set of processors that can perform tasks (or processes) in p",
  "applicationCategory": "info",
  "releaseNotes": "First release with GLPK, in Beta testing",
  "developmentStatus": "concept",
  "keywords": [
    "distributed systems",
    "glpk",
    "optimisation",
    "OR"
  ],
  "programmingLanguage": [
    "Java"
  ],
  "author": [
    {
      "@type": "Person",
      "@id": "https://orcid.org/0000-0002-9777-5560",
      "givenName": "Morane",
      "familyName": "Gruenpeter",
      "email": "morane.gg@gmail.com",
      "affiliation": {
        "@type": "Organization",
        "name": "Software Heritage"
      }
    }
  ]
}
```

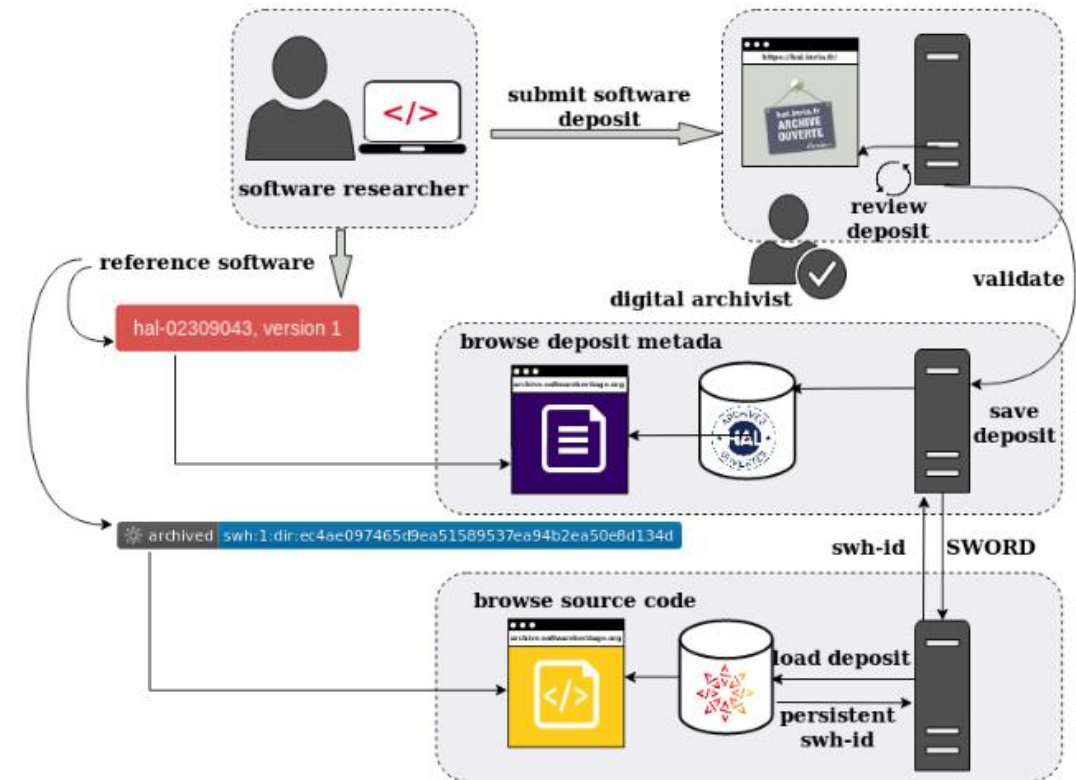
Source : fichier codemeta.json du projet AffectationRO,
https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://github.com/moranegg/AffectationRO,
consulté le 25/05/2023

03.1

Le dépôt source

Modèle général

- Création et dépôt dans HAL d'une archive compressée contenant tous les fichiers du code source (.zip, .tar.gz)
- Renseignement des métadonnées du formulaire de dépôt
- Validation du dépôt avec option de transfert vers SWH
- Modération avant mise en ligne
- Récupération de l'identifiant SWHID dans la notice HAL
- Archivage du code source dans SWH



Source : Roberto DI COSMO, Morane GRUENPETER, Bruno MARMOL, et al. « Curated Archiving of Research Software Artifacts : Lessons Learned from the French Open Archive (HAL) », IJDC. 2020, vol. 15, no 1. En ligne : <https://doi.org/10.2218/ijdc.v15i1.698>, consulté le 25/05/2023

Démo

Nous allons montrer pas à pas comment réaliser un dépôt source de logiciel dans HAL

Dépôt de ce code source dans l'environnement test de HAL :

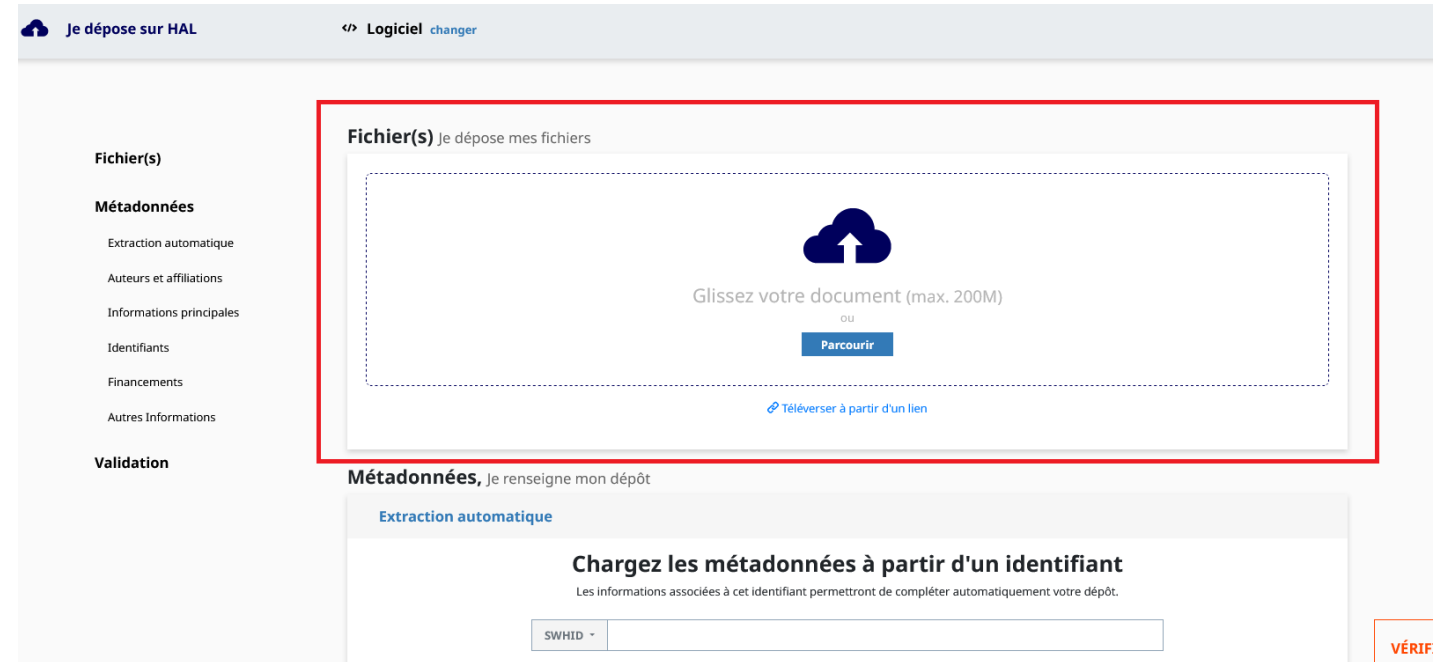
Morane Gruenpeter. AffectationRO- The assignment problem. 2014



5 min

Le dépôt des fichiers

- Se connecter à son compte utilisateur HAL et cliquer sur le bouton « Déposer »
- Sélectionner le type de document « Logiciel »
- Déposer le fichier compressé
 - Par Glisser / Déposer
 - Par téléchargement
- Quelques règles :
 - Un seul fichier compressé par dépôt
 - L'archive (.zip, .tar.gz) doit contenir tous les fichiers qui ne pourront être décompressés avant validation du formulaire de dépôt
 - L'archive (.zip, .tar.gz) ne peut contenir une autre archive (.zip, .tar.gz)



Je dépose sur HAL

Logiciel changer

Fichier(s)

Métadonnées

Extraction automatique

Auteurs et affiliations

Informations principales

Identifiants

Financements

Autres Informations

Validation

Fichier(s) Je dépose mes fichiers

Glissez votre document (max. 200M)

ou

Parcourir

Téléverser à partir d'un lien

Métadonnées, Je renseigne mon dépôt

Extraction automatique

Chargez les métadonnées à partir d'un identifiant

Les informations associées à cet identifiant permettront de compléter automatiquement votre dépôt.

SWHID

VÉRIF

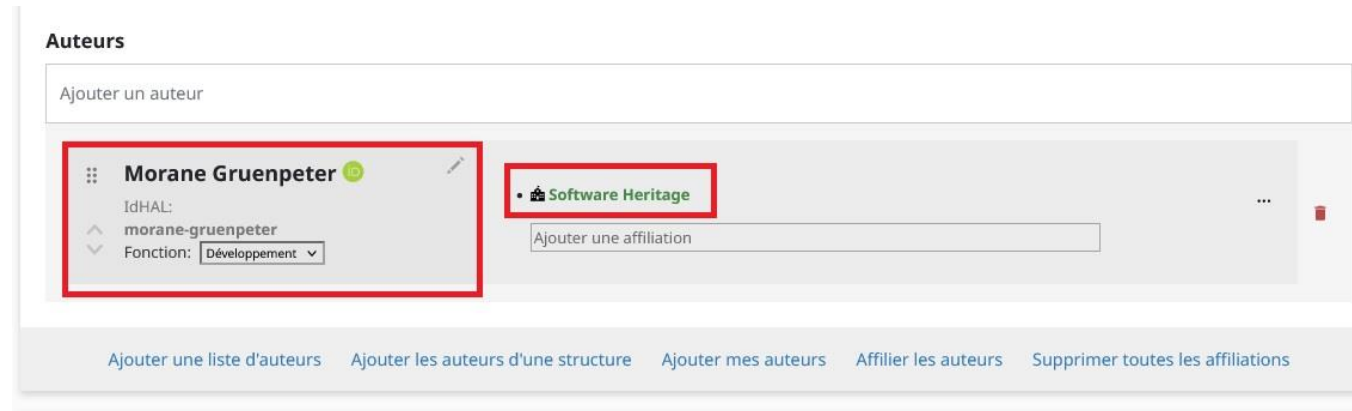
Source : Formulaire de dépôt HAL, consulté le 25/05/2023

Informations sur les auteurs

- Ajouter le(s) auteur(s)
 - Tous les auteurs du fichier AUTHORS
 - Possibilité de renseigner le rôle de chaque auteur


- Ajouter une affiliation pour chaque auteur
 - Au moins un auteur du dépôt doit être affilié

NB : Possibilité d'ajouter dans l'archive compressée un fichier CONTRIBUTORS pour indiquer les personnes ayant contribué à la création du logiciel sans être considérés comme les auteurs



Auteurs

Ajouter un auteur

Morane Gruenpeter 

IdHAL: morane-gruenpeter

Fonction: Développement

Software Heritage

Ajouter une affiliation

Ajouter une liste d'auteurs Ajouter les auteurs d'une structure Ajouter mes auteurs Affilier les auteurs Supprimer toutes les affiliations

Source : Formulaire de dépôt HAL, Morane Gruenpeter. AffectationRO- The assignment problem. 2014, [\[swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d;origin=https://github.com/moranegg/AffectationRO;visit=swh:1:snp:3d73c454047658280dbbf7a5fbc3c25f788d4a37;anchor=swh:1:rev:44c5ef119c5af14193b5a9322d34c34c395e4c8b\]](https://hal.science/hal-03566866), <https://hal.science/hal-03566866>, consulté le 25/05/2023

Métadonnées générales et spécifiques

➤ Compléter les métadonnées

o Métadonnées générales

- Nom *
- Domaine*
- Description
- Mots clés
- Identifiants
- Données / Publications associées
- Date de production
- Financement
- Autres informations

o Métadonnées spécifiques

- Licences* (sur la base de la liste de référence SPDX)
- Langage de programmation
- Code Repository
- Plateforme / OS
- Version
- Etat du développement
- Outils de développement

Informations principales

Informations nécessaires pour la citation

Nom *

Vous pouvez renseigner le titre en plusieurs langues : choisir la langue et cliquer sur + pour ajouter une nouvelle langue

Anglais ▾

AffectationRO- The assignment problem

Supprimer

Français ▾

+ Ajouter

Domaine *

Discipline principale

Informatique [cs] ▾

Le domaine que vous mettez en premier, sera considéré comme la discipline principale du dépôt (article, logiciel ou autre). L'ordre des domaines peut être changé par glissé/déposé.

Description

Cliquez sur + pour ajouter une nouvelle langue

Anglais ▾

A java implementation for The Assignment Problem a distributed system as a set of processors that can perform tasks (or processes) in parallel. We therefore consider a set of m processors, each equipped with a certain amount of random access memory (RAM). We associate a cost to pay to perform this task on this processor, and each pair of tasks is associated with a communication cost. The Assignment problem works on minimizing the cost and

Source : Formulaire de dépôt HAL, Morane Gruenpeter. AffectationRO- The assignment problem. 2014, {swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d;origin=https://github.com/moranegg/AffectationRO;visit=swh:1:snp:3d73c454047658280dbbf7a5fbc3c25f788d4a37;anchor=swh:1:rev:44c5ef119c5af14193b5a9322d34c34c395e4c8b}, <https://hal.science/hal-03566866>, consulté le 25/05/2023

Validation du dépôt et transfert vers SWH

- Transférer son dépôt
 - Optionnel
 - Conditions
 - Les fichiers déposés doivent être sous une licence libre
 - Les fichiers ne peuvent pas être sous embargo
 - L'accès au dépôt sur SWH est disponible après un délai de traitement (7-30 jours)
- Accepter les conditions de HAL et déposer
- Le dépôt est modéré avant sa mise en ligne

Validation, Je vérifie mes informations et j'accepte les conditions

Je vérifie les informations de la citation

Logiciel

Morane Gruenpeter. AffectationRO- The assignment problem. 2023, {swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d;origin=https://github.com/moranegg/AffectationRO;visit=swh:1:snp:3d73c454047658280dbbf7a5fbc3c25f788d4a37;anchor=swh:1:rev:44c5ef119c5af14193b5a9322d34c34c395e4c8b}

Je transfère mon dépôt



Software Heritage a pour objectif de collecter, préserver, et rendre accessible, à tous, le code source de tous les logiciels disponibles. Pour pouvoir transférer votre logiciel dans la plus grande archive de code source, votre dépôt doit satisfaire les conditions suivantes:

- ✓ Les fichiers déposés doivent être sous une licence libre.
- ✓ Les fichiers ne peuvent pas être sous embargo.

L'accès à votre dépôt sur Software Heritage sera disponible dans un délai de 7-30 jours (le temps de traitement de votre dépôt)

☒ Je transfère vers Software Heritage

J'accepte les conditions de HAL

En déposant ce document, le contributeur (je) accorde la licence suivante à HAL :

- J'autorise HAL à mettre en ligne et à distribuer cet article ;
- Je reconnais avoir pris connaissance que les dépôts ne peuvent pas être supprimés, une fois acceptés ;
- Je comprends que HAL se réserve le droit de reclasser ou de rejeter tout dépôt.

☒ J'accepte ces conditions

DÉPOSER

Source : Formulaire de dépôt HAL, Morane Gruenpeter. AffectationRO- The assignment problem. 2014, {swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d;origin=https://github.com/moranegg/AffectationRO;visit=swh:1:snp:3d73c454047658280dbbf7a5fbc3c25f788d4a37;anchor=swh:1:rev:44c5ef119c5af14193b5a9322d34c34c395e4c8b}, <https://hal.science/hal-03566866>, consulté le 25/05/2023

Dépôt en ligne

- Si option de transfert vers SWH validée :
 - Archivage du code source sur SWH
 - Un identifiant direct au code source (SWHID) est inclus dans la notice HAL et dans le format de citation


➤ Consulter sur SWH

➤ Citer

➤ Exporter

➤ Partager

NB : Une fois le dépôt validé il est poussé grâce au protocole SWORD sur Software Heritage. SWH procède à l'injection du logiciel sur sa plateforme d'archivage et génère l'identifiant intrinsèque - le SWHID. Hal revient récupérer le SWHID qui est mis à disposition sur la page du dépôt et dans le format de citation.

Consulter sur  Software Heritage

Dates et versions
hal-03566866, **version 1** (11-02-2022)

Identifiants
HAL Id : hal-03566866, version 1
SWHID : swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d;origin=https://github.com/moranegg/AffectationRO;visit=swh:1:snp:3d73c454047658280dbbf7a5fbc3c25f788d4a37;anchor=swh:1:rev:44c5ef119c5af14193b5a9322d34c34c395e4c8b

Métadonnées

- version : 1.0.0
- Licences : <https://spdx.org/licenses/AGPL-3.0>
- Langage de programmation : java
- Code Repository : <https://github.com/moranegg/AffectationRO>

Citer
Morane Gruenpeter. AffectationRO- The assignment problem. 2014, (swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d;origin=https://github.com/moranegg/AffectationRO;visit=swh:1:snp:3d73c454047658280dbbf7a5fbc3c25f788d4a37;anchor=swh:1:rev:44c5ef119c5af14193b5a9322d34c34c395e4c8b), (hal-03566866)

Exporter


CodeMeta BibTeX TEI

Dublin Core DC Terms EndNote

Datacite

Logiciel Année :

AffectationRO- The assignment problem
Morane Gruenpeter (1)
Afficher plus de détails

 1 Software Heritage

Résumé en
A java implementation for The Assignment Problem a distributed system as a set of processors that can perform tasks (or processes) in parallel. We therefore consider a set of m processors, each equipped with a certain amount of random access memory (RAM). We associate a cost to pay to perform this task on this processor, and each pair of tasks is associated with a communication cost. The Assignment problem works on minimizing the cost and maximizing the tasks performed.

Mots clés en
distributed systems gipk optimisation OR

Domaines
Informatique [cs]

Liste complète des métadonnées

Morane Gruenpeter : Connectez-vous pour contacter le contributeur
<https://inria.hal.science/hal-03566866>
Soumis le : vendredi 11 février 2022-17:30:54
Dernière modification le : mardi 17 janvier 2023-10:33:57

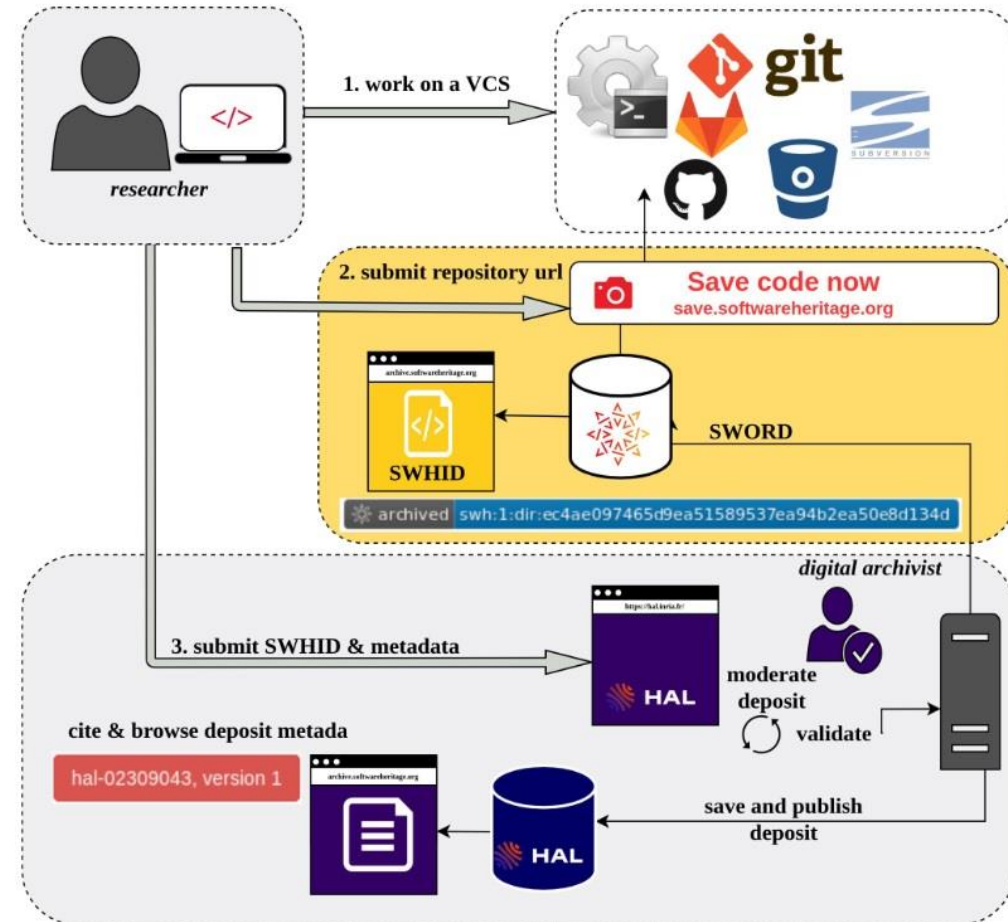
Source : Morane Gruenpeter. AffectationRO- The assignment problem. 2014, (swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d;origin=https://github.com/moranegg/AffectationRO;visit=swh:1:snp:3d73c454047658280dbbf7a5fbc3c25f788d4a37;anchor=swh:1:rev:44c5ef119c5af14193b5a9322d34c34c395e4c8b), <https://hal.science/hal-03566866>, consulté le 25/05/2023

03.2

Le dépôt SWHID

Modèle général

- Dépôt dans HAL avec le SWHID, identifiant attribué par Software Heritage au code source archivé
- Renseignement des métadonnées du formulaire de dépôt
 - Les métadonnées peuvent être récupérées automatiquement à partir du fichier codemata.json présent dans le code source
- Validation du dépôt par le déposant
- Modération avant mise en ligne
- L'identifiant SWHID est intégré à la notice HAL
- Le code source reste archivé dans SWH



Source : Morane Gruenpeter, Jozefina Sadowska, Estelle Nivault, Alain Monteil. Create software deposit in HAL: User guide and best practices. [Technical Report] Inria; CCSD; Software Heritage. 2022. <https://hal.science/hal-01872189v2>, consulté le 25/05/2023

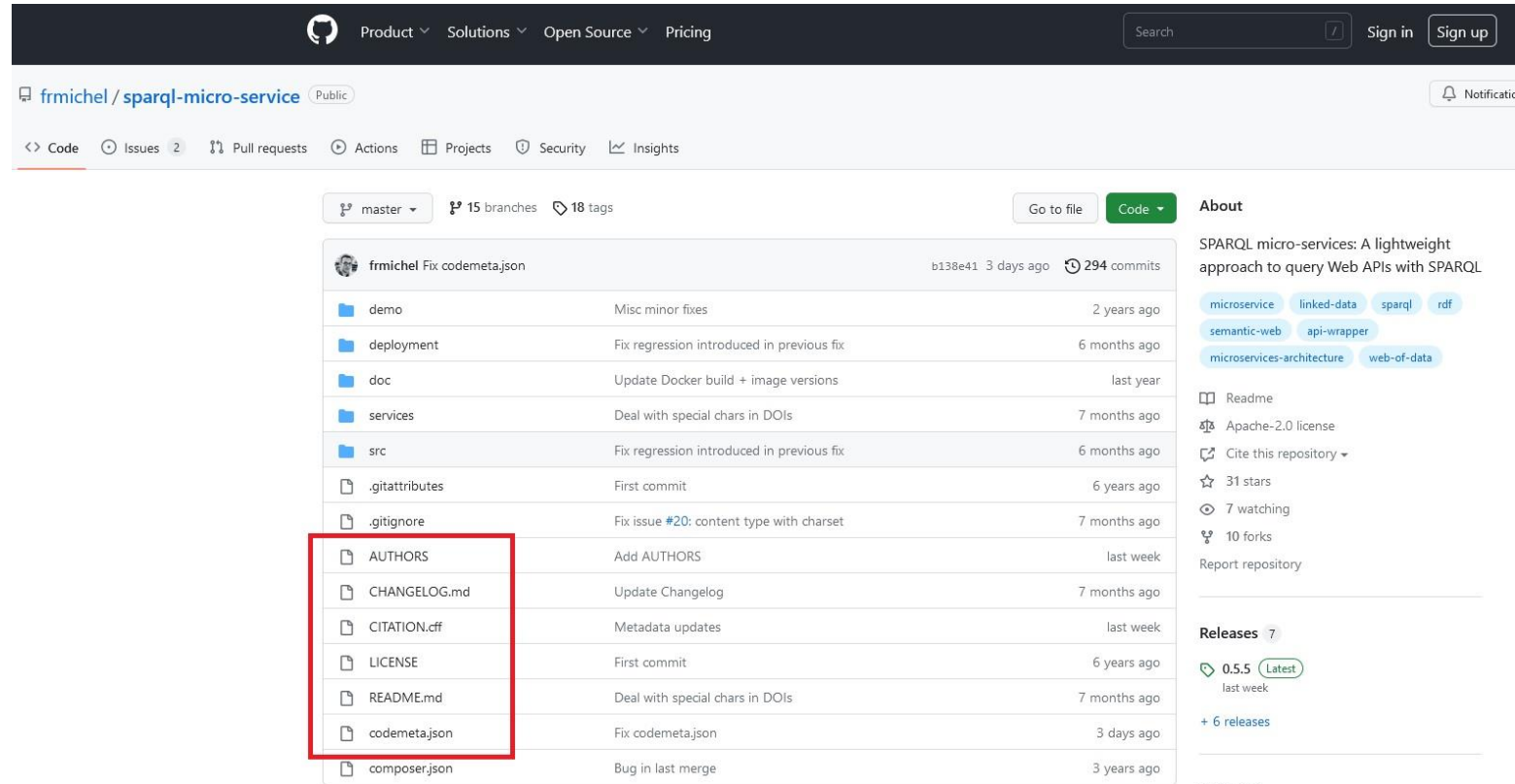
En amont du dépôt

➤ Préparation du code source du logiciel sur la plateforme de développement

➤ Fichiers recommandés :

- README
- AUTHORS
- LICENSE

NB : dans le cadre du dépôt SWHID, un fichier codemeta.json est utile car il permet la récupération automatique des métadonnées dans formulaire de dépôt HAL



The screenshot shows the GitHub repository page for 'frmichel / sparql-micro-service'. The repository is public and has 15 branches and 18 tags. The file list on the left shows the following files and their commit history:


File	Commit Message	Commit Date
demo	Misc minor fixes	2 years ago
deployment	Fix regression introduced in previous fix	6 months ago
doc	Update Docker build + image versions	last year
services	Deal with special chars in DOIs	7 months ago
src	Fix regression introduced in previous fix	6 months ago
.gitattributes	First commit	6 years ago
.gitignore	Fix issue #20: content type with charset	7 months ago
AUTHORS	Add AUTHORS	last week
CHANGELOG.md	Update Changelog	7 months ago
CITATION.cff	Metadata updates	last week
LICENSE	First commit	6 years ago
README.md	Deal with special chars in DOIs	7 months ago
codemeta.json	Fix codemeta.json	3 days ago
composer.json	Bug in last merge	3 years ago

The 'AUTHORS', 'CHANGELOG.md', 'CITATION.cff', 'LICENSE', and 'README.md' files are highlighted with a red box. The repository also has 294 commits and 31 stars.

Source : Github. Franck Michel. SPARQL micro-services: A lightweight approach to query Web APIs with SPARQL. <https://github.com/frmichel/sparql-micro-service>, consulté le 25/05/2023

En amont du dépôt

- Archivage du code source dans Software Heritage (<https://archive.softwareheritage.org/save/>)
- Fonctionnalité « Save code now »
 - Origin type = Type de VCS (Version Control System)
 - Origin url = Url du code sur le code repository
 - Submit = Soumettre


Software Heritage
Archive

Save code now

Features

Search

Downloads

Save code now

Add forge now

Help

Help

Browse save requests

You can contribute to extend the content of the Software Heritage archive by submitting an origin save request. To do so, fill the required info in the form below:

Origin type

git

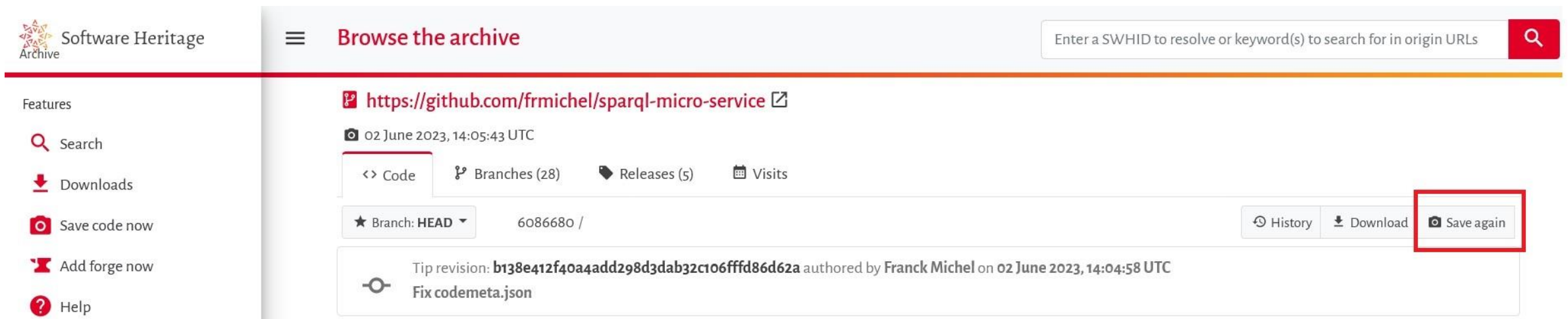
Origin url

Submit

A "Save code now" request takes the following parameters:

En amont du dépôt

- Si le code est déjà archivé dans Software Heritage, fonctionnalité « Save again » pour mise à jour de l'archive

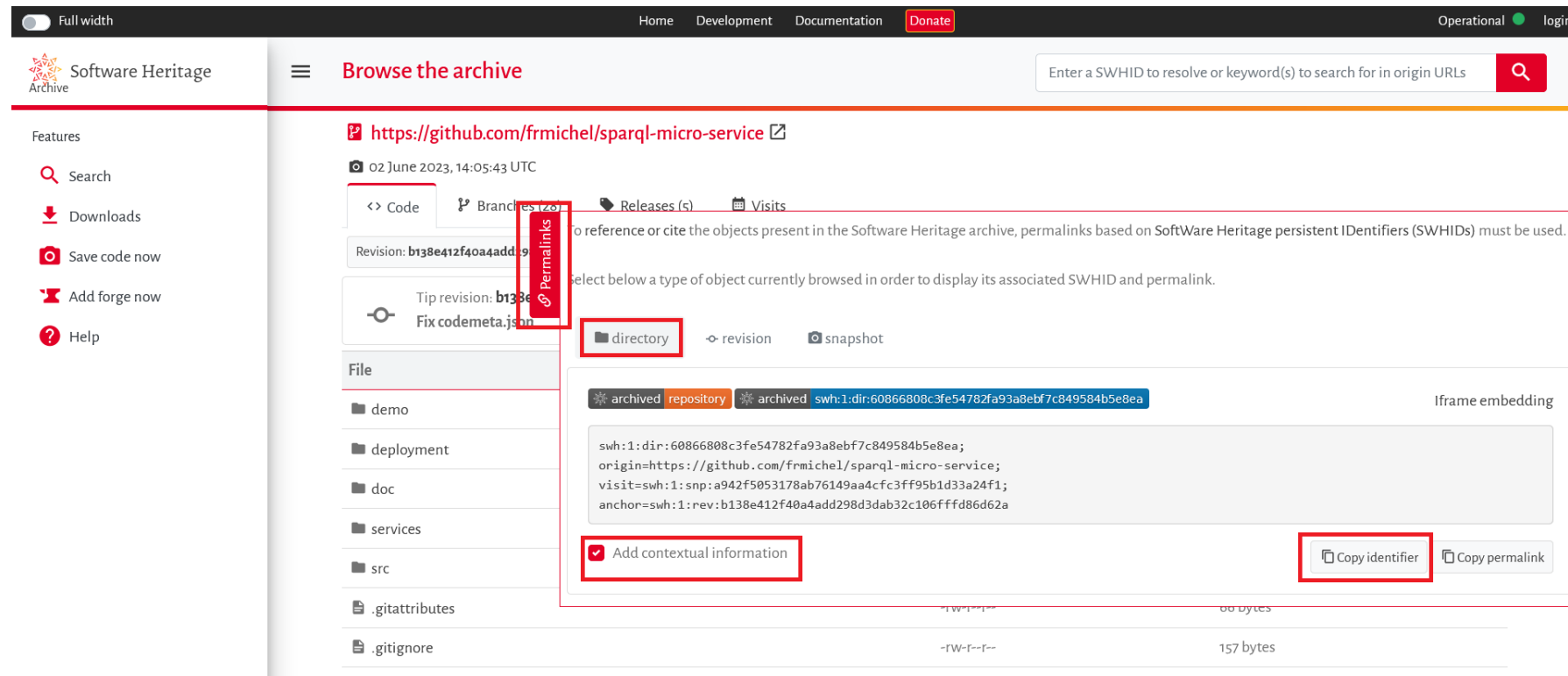


The screenshot shows the Software Heritage 'Browse the archive' page. The left sidebar contains a 'Features' menu with options: Search, Downloads, Save code now, Add forge now, and Help. The main content area displays details for a repository: `https://github.com/frmichel/sparql-micro-service`, dated 02 June 2023, 14:05:43 UTC. Below this, there are tabs for Code, Branches (28), Releases (5), and Visits. A 'Branch: HEAD' dropdown shows the commit hash `6086680`. At the bottom, a 'Tip revision' is shown: `b138e412f40a4add298d3dab32c106fffd86d62a` authored by Franck Michel on 02 June 2023, 14:04:58 UTC. The file `Fix codemeta.json` is listed. In the top right corner of the main area, there are buttons for 'History', 'Download', and 'Save again'. The 'Save again' button is highlighted with a red rectangular box.

Source : Software Heritage. Franck Michel. SPARQL Micro-Services.
https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://github.com/frmichel/sparql-micro-service, consulté
 le 25/05/2023

En amont du dépôt

- Sélectionner et récupérer l'identifiant SWHID du code source à déposer dans HAL
 - Cliquer sur Permalinks
 - Choisir le type d'objet « Directory »
 - Ajouter les informations contextuelles (Cela permet de garder un lien avec l'historique de développement)
 - Copier l'identifiant



Software Heritage

Browse the archive

Enter a SWHID to resolve or keyword(s) to search for in origin URLs

https://github.com/frmichel/sparql-micro-service

02 June 2023, 14:05:43 UTC

<> Code Branches Releases Visits

Revision: b138e412f40a4add9

Tip revision: b138e412f40a4add9

Fix codemeta.json

select below a type of object currently browsed in order to display its associated SWHID and permalink.

directory revision snapshot

archived repository archived swh:1:dir:60866808c3fe54782fa93a8ebf7c849584b5e8ea

Iframe embedding

```
swh:1:dir:60866808c3fe54782fa93a8ebf7c849584b5e8ea;
origin=https://github.com/frmichel/sparql-micro-service;
visit=swh:1:snp:a942f5053178ab76149aa4cfc3ff95b1d33a24f1;
anchor=swh:1:rev:b138e412f40a4add298d3dab32c106fffd86d62a
```

☒ Add contextual information

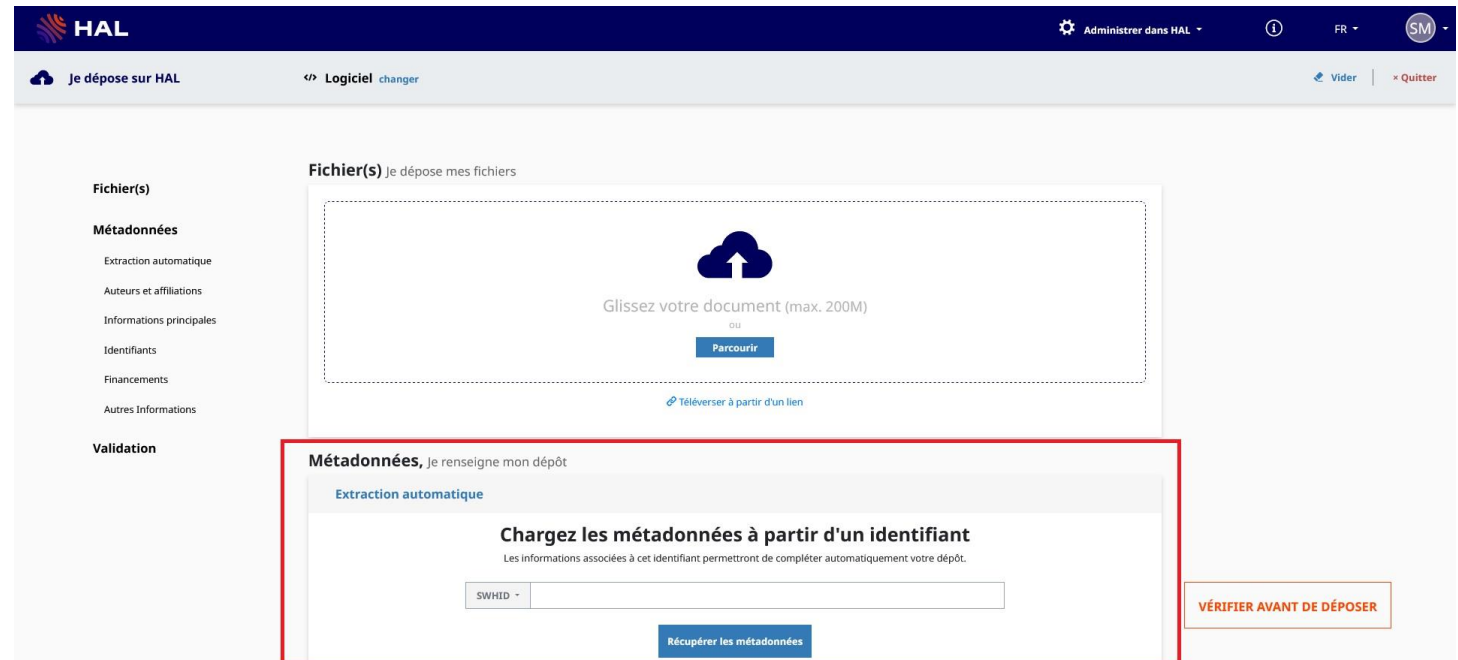
Copy identifier Copy permalink

demo deployment doc services src .gitattributes .gitignore

Source : Software Heritage.
 Franck Michel. SPARQL
 Micro-Services.
https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://github.com/frmichel/sparql-micro-service, consulté le 25/05/2023

Le dépôt à partir du SWHID

- Se connecter à son compte utilisateur HAL et cliquer sur le bouton « Déposer »
- Sélectionner le type de document « Logiciel »
- Renseigner le SWHID dans le formulaire
- Contrôler et Compléter les métadonnées :
 - Si un fichier codemeta.json est présent dans le code source HAL peut récupérer automatiquement les métadonnées
- Valider le dépôt



HAL

Administrer dans HAL

FR

SM

Je dépose sur HAL

Logiciel changer

Vider Quitter

Fichier(s) Je dépose mes fichiers

Glissez votre document (max. 200M)
ou
Parcourir

Téléverser à partir d'un lien

Métadonnées, je renseigne mon dépôt

Extraction automatique

Chargez les métadonnées à partir d'un identifiant
Les informations associées à cet identifiant permettront de compléter automatiquement votre dépôt.

SWHID

Récupérer les métadonnées

VÉRIFIER AVANT DE DÉPOSER

Source : Formulaire de dépôt HAL, consulté le 25/05/2023

A VOUS !

Vous allez tester un dépôt SWHID dans HAL



15 min

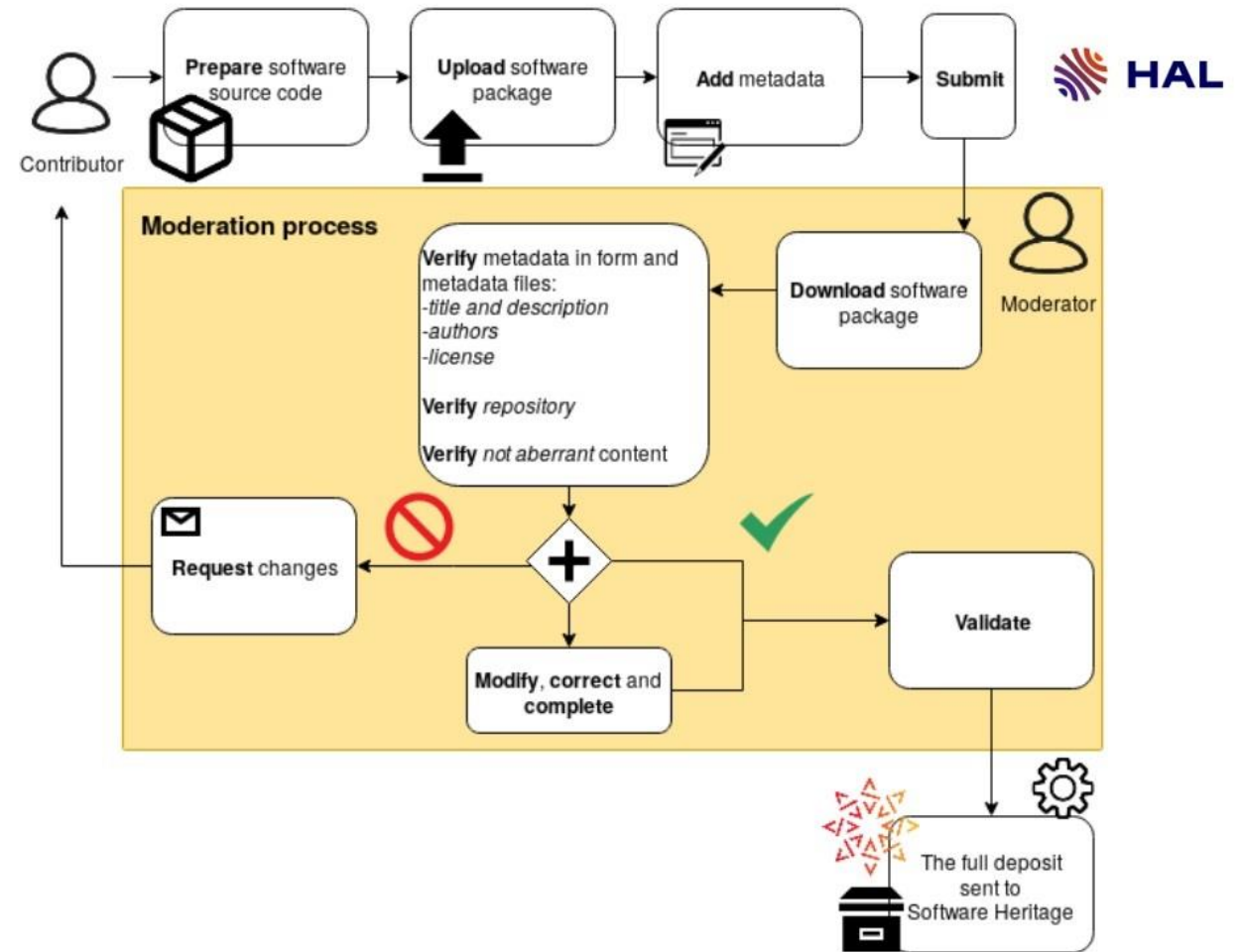
- Rendez vous sur Software Heritage (<https://archive.softwareheritage.org/>)
- Recherchez le projet : ***SPARQL Micro-Services*** de l'auteur ***Franck Michel***
- Dans SWH, vérifiez la présence de tous les fichiers nécessaires
- Récupérez le SWHID avec les informations contextuelles
- Déposez dans HAL à partir du SWHID (attention, dans l'environnement de test !)
- Complétez les métadonnées manquantes (affiliation, domaine, etc.)
- Vérifiez les métadonnées déposées
- Validez le dépôt

04

Les grands principes de la
modération des codes
sources dans HAL

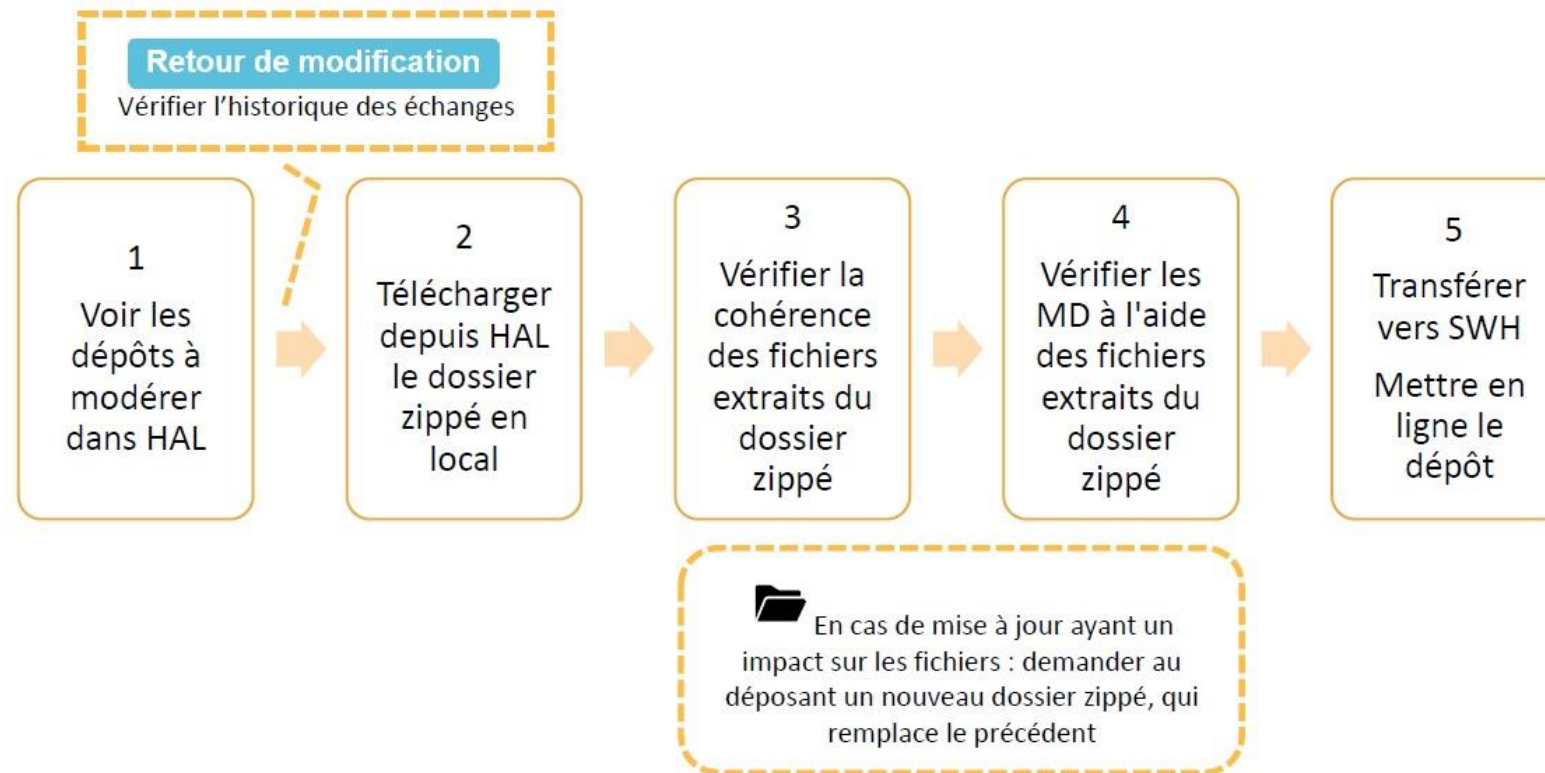
Modèle général

- Vérification du dépôt avant sa mise en ligne
 - Contrôle de la cohérence et de l'exactitude des métadonnées déposées
- Dialogue entre le modérateur et le déposant (demandes de modifications, de corrections, de compléments)
- Validation, mise en ligne sur HAL et transfert vers SWH
- Il ne s'agit pas du contrôle de la lisibilité ou de la qualité du logiciel



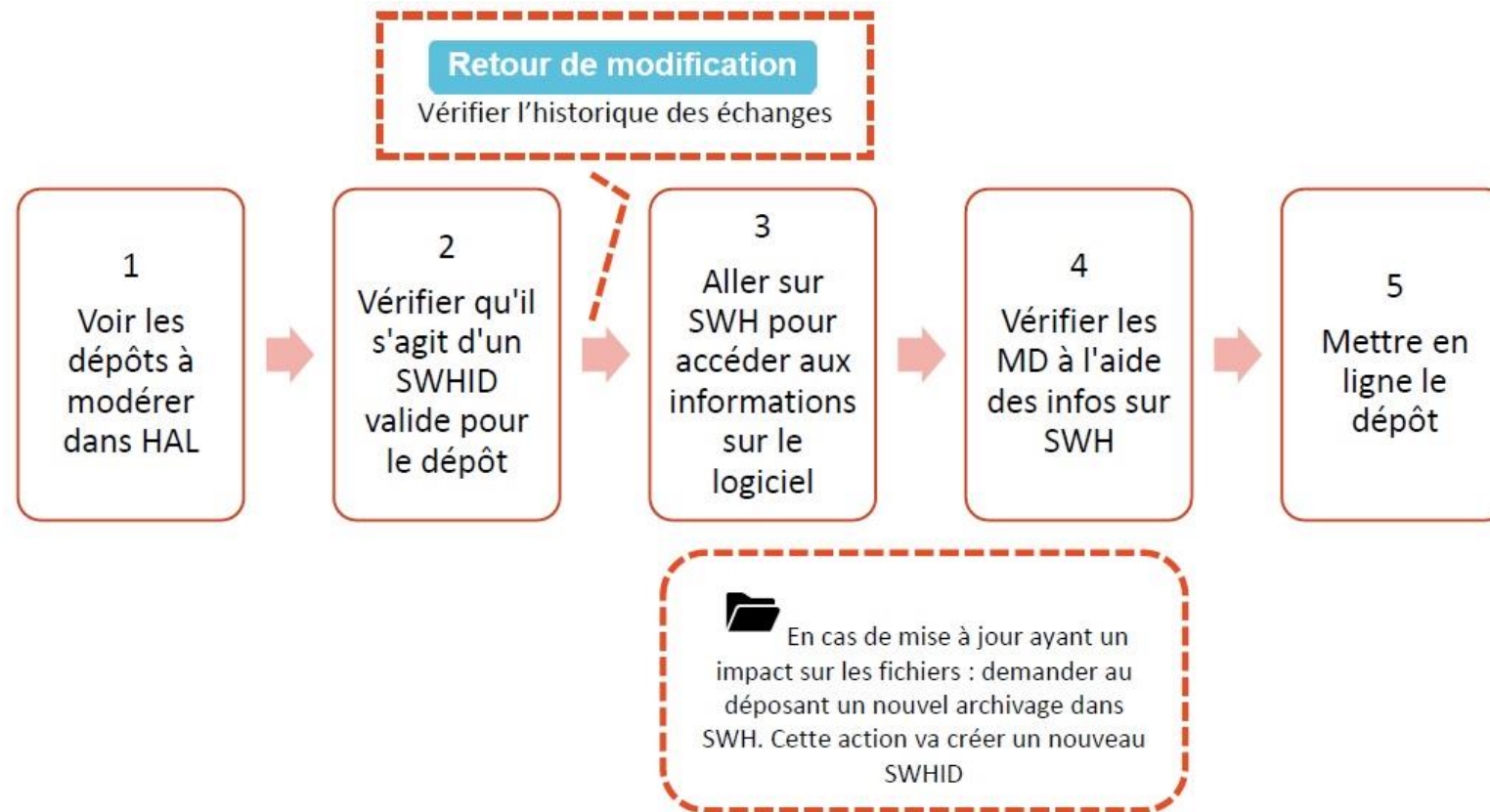
Source : Roberto DI COSMO, Morane GRUENPETER, Bruno MARMOL, et al. « Curated Archiving of Research Software Artifacts : Lessons Learned from the French Open Archive (HAL) », IJDC. 2020, vol. 15, no 1. En ligne : <https://doi.org/10.2218/ijdc.v15i1.698>, consulté le 25/05/2023

Modérer un dépôt source



Source : Sabrina Granger, Morane Gruenpeter, Alain Monteil, Estelle Nivault, Jozefina Sadowska. Modérer un dépôt logiciel dans HAL : dépôt source et dépôt SWHID. [Rapport Technique] Inria; CCSD; Software Heritage. 2022. <https://hal.science/hal-01876705>, consulté le 25/05/2023

Modérer un dépôt SWHID



Source : Sabrina Granger, Morane Gruenpeter, Alain Monteil, Estelle Nivault, Jozefina Sadowska. Modérer un dépôt logiciel dans HAL : dépôt source et dépôt SWHID. [Rapport Technique] Inria; CCSD; Software Heritage. 2022. <https://hal.science/hal-01876705>, consulté le 25/05/2023

05

Des cas d'usage

05.1

Articuler Publications et Logiciels dans HAL

Lier Publications, Données et Logiciels dans HAL

➤ Dans le formulaire de dépôt, des champs permettent de lier le logiciel avec :

- Des ressources identifiées dans d'autres entrepôts
- Des données
- Des publications dans HAL

Identifiants

Ajoutez les identifiants du dépôt dans d'autres contextes: DOI, arXiv, PubMed, ADS, Software Heritage-hash, et autres.

DOI ▾

10.5281/zenodo.7756248

Supprimer

SWHID ▾

swlh:1:dir:60866808c3fe54782fa93a8ebf7c849584b5e8ea;origin=https://github.com/frmichel/sparql-micro-s

Supprimer

ArXiv ▾

+ Ajouter

Données associées

Ajoutez l'identifiant DOI fourni par l'entrepôt où vos données sont archivées.

+ Ajouter

Publications associées

Ajoutez les identifiants des publications associées à votre logiciel

HAL ▾

hal-01722792

Supprimer

DOI ▾

+ Ajouter

Source : Formulaire de dépôt HAL. Franck Michel. SPARQL Micro-Services. 2018, <https://hal.science/hal-04115815>, consulté le 25/05/2023

05.2

Un logiciel avec plusieurs versions dans HAL

Quand déposer une nouvelle version ?

Ce que nous pouvons conseiller :

- Une nouvelle version dans HAL = une nouvelle édition (fonctionnalités, auteurs, etc.) ≠ une mise à jour sans impact notable sur le projet initial

Création VS Amélioration

- Options possibles pour un même logiciel :

Dépôt V1 = dépôt SWHID , Dépôt V2 = dépôt Source

Ou vice et versa

(Mais aussi : méthode dépôt V1 = méthode dépôt V2)

- Autre principe : toute version précédente mise en ligne a un SWHID :

- soit via un transfert vers SWH si le dépôt initial est un dépôt source
- soit SWHID "natif" si le dépôt initial est un dépôt SWHID

Exception : pas de SWHID pour une version antérieure mise en ligne si, dans le cas d'un dépôt source, le déposant a refusé le transfert vers SWH

- Pour le modérateur, pas de changement dans la procédure de mise en ligne

Références

- Pierre Alliez, Roberto Di Cosmo, Benjamin Guedj, Alain Girault, Mohand-Said Hacid, et al.. Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria. *Computing in Science and Engineering*, 2019, pp.1-14. [10.1109/MCSE.2019.2949413](#). [hal-02135891v2](#)
- Yannick Barborini, Roberto Di Cosmo, Antoine R. Dumont, Morane Gruenpeter, Bruno P Marmol, et al.. La création du nouveau type de dépôt scientifique - Le logiciel. *JSO 2018 - 7es journées Science Ouverte Couperin : 100 % open access : initiatives pour une transition réussie*, Jan 2018, Paris, France. , 2018. [hal-01688726](#)
- CCSD. (2022, 12 juillet). *Parlons Science ouverte #4 : les logiciels dans HAL*. [Vidéo]. Canal-U. <https://www.canal-u.tv/131520>
- Mélanie Clément-Fontaine, Roberto Di Cosmo, Bastien Guerry, Patrick Moreau, François Pellegrini. Note d'opportunité sur la valorisation des logiciels issus de la recherche. [Rapport de recherche] Comité pour la science ouverte. 2019, 6 p. [hal-03606374](#)
- Roberto Di Cosmo, Morane Gruenpeter, Bruno P Marmol, Alain Monteil, Laurent Romary, et al.. Curated Archiving of Research Software Artifacts : lessons learned from the French open archive (HAL). IDCC 2020 - International Digital Curation Conference, Feb 2020, Dublin, Ireland. [10.2218/ijdc.v15i1.698](#). [hal-02475835](#)
- Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli. Referencing Source Code Artifacts: a Separate Concern in Software Citation. *Computing in Science and Engineering*, In press, pp.1-9. [10.1109/MCSE.2019.2963148](#). [hal-02446202](#)
- Sabrina Granger, Morane Gruenpeter, Alain Monteil, Estelle Nivault, Jozefina Sadowska. Modérer un dépôt logiciel dans HAL : dépôt source et dépôt SWHID. [Rapport Technique] Inria; CCSD; Software Heritage. 2022. [hal-01876705v2](#)
- Morane Gruenpeter, Jozefina Sadowska, Estelle Nivault, Alain Monteil. Create software deposit in HAL: User guide and best practices. [Technical Report] Inria; CCSD; Software Heritage. 2022. [hal-01872189v2](#)
- Ministère de l'Enseignement supérieur et de la Recherche. Science ouverte - Codes et logiciels. Passeport pour la science ouverte, 2022. https://www.ouvrirlascience.fr/wp-content/uploads/2022/10/Passeport_Codes-et-logiciels_WEB.pdf
- Alain Monteil, Morane Gruenpeter, Jozefina Sadowska, Estelle Nivault. "Garantir la cohérence des données constitue le cœur de notre activité" : entretien autour des enjeux descriptifs du code source. Bulletin des bibliothèques de France, 2021, Dossier BBF 2021-1 • Code source : libérer le patrimoine !. [hal-03239502](#)
- Jozefina Sadowska, Estelle Nivault, Morane Gruenpeter. MSH SUD. (2021, 9 juin). *Valorisation du logiciel open source via HAL et Software Heritage*. [Vidéo]. Canal-U. <https://www.canal-u.tv/135836>



Merci !

Pour en savoir plus, rendez-vous sur : <https://doc.archives-ouvertes.fr/deposer/deposer-le-code-source/>