Science ouverte
Université Grenoble Alpes

Référencer les logiciels de la recherche grâce à Software Heritage et HAL

Open Science Days @ UGA 2022

Lucie Albaret, Alexis Arnaud, Florent Chuffart, Violaine Louvet, Cellule Data Grenoble Alpes

14 décembre 2022

Pourquoi référencer les codes de recherche ?

Pourquoi archiver ?

- Le code source est fragile :
  - Obsolescence des formats, problème matériel, dépendances à des outils (forge par exemple) qui disparaissent ...
  - La perte des codes ayant été utilisés pour de la production scientifique arrive malheureusement régulièrement
- Les logiciels sont un des piliers des processus de recherche, au côté des publications et des données et il est essentiel de les préserver

Pourquoi signaler ?

- Assurer la description
- Faciliter la recherche / Rechercher facilement (par domaine scientifique en particulier)
- Permettre la citation
- Valoriser les logiciels

HAL, Hyper Articles en Ligne

- **Archive ouverte pluridisciplinaire**
- Initiée en 2000 par le CNRS et exploitée par le CCSD – Centre pour la Communication Scientifique Directe
- Fournit des outils pour l'archivage et la diffusion ouverte des résultats scientifiques.
- Où les chercheurs peuvent déposer leurs résultats académiques dans le respect de leurs droits d'auteur
- Supporte différents types de dépôt :
  - Publications,
  - Documents (par exemple préprints et rapport),
  - Thèses ...
- Pour rendre la **recherche aussi accessible et ouverte que possible**

Software Heritage

- Initiative dont l'objectif est de construire une **archive universelle des codes sources**
- En les collectant, les préservant et les partageant sur le long term
- Lancée en 2016 par INRIA et soutenue par l'UNESCO
- Collecte de l'**intégralité des logiciels disponibles publiquement** sous forme de code source.
- Depuis des plateformes d'hébergement de code, comme GitHub, GitLab.com ou Bitbucket, et des archives de paquets, comme Npm ou Pypi ...

HAL + Software Heritage

Un nouveau type de dépôt sur HAL

- Collaboration initiée en 2018 entre HAL et SWH
- Phase de test avec INRIA
- En particulier, le dépôt SWHID en beta-test depuis début d'année sur HAL-INRIA et attendu en production en début d'année

Complémentarité des deux plateformes

- Grande visibilité des logiciels dans une démarche de science ouverte via HAL
- Archivage pérenne via Software Heritage
- Modération des métadonnées
- Différents formats d'export pour faciliter la citation

Exemple d’un code de recherche référencé

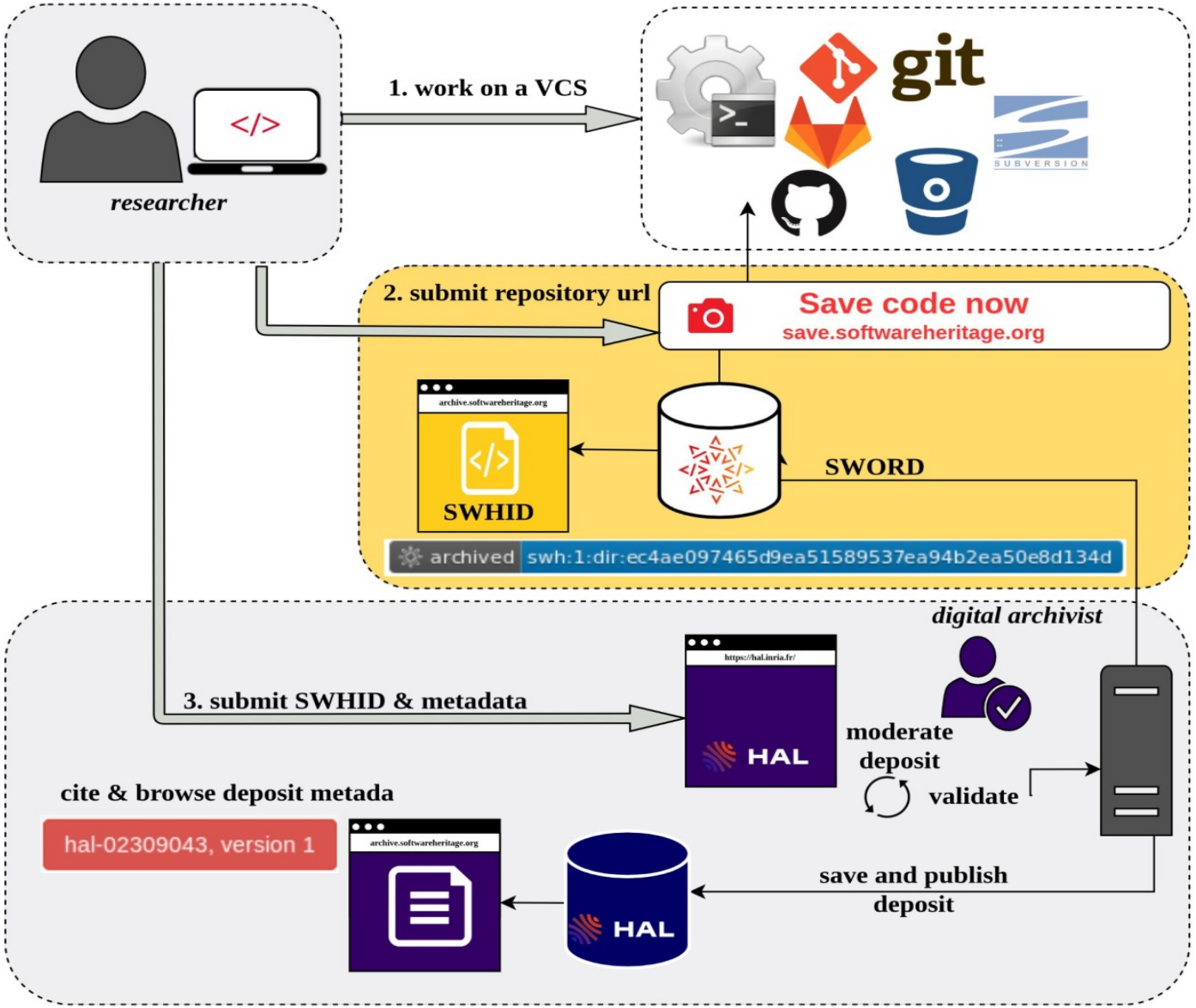
Science ouverte Grenoble Alpes

DEMO

Préalables

- **On s'intéresse ici au dépôt via SWHID**
  - Cela suppose l'utilisation d'une forge (qui soit moissonnée par SWH) : c'est **LA bonne pratique pour le développement des codes**
  - **Processus SWH puis HAL** : Les **informations sur la forge sont toujours à jour** donc l'archive sur SWH l'est aussi (possibilité de forcer la mise à jour) et il reste à signaler sur HAL
  - A noter qu'il est aussi possible de déposer via une archive zip sur HAL. Dans ce cas les fichiers des codes sont automatiquement archivés sur SWH.
- **Que déposer ?**
  - Le code source
  - D'un logiciel développé dans un milieu académique

Etapes du processus



Checklist : préparer le logiciel

- **Les fichiers indispensables**
  - AUTHORS
  - LICENSE
  - README
- Optionnel mais **fortement conseillé**
  - Codemeta.json

DEMO

Archivage sur SWH

Science ouverte Grenoble Alpes

- **Code déjà présent**
  - Cas où le code est hébergé sur une forge moissonnée par SWH : par exemple gricad-gitlab
  - Il existe un plugin de navigateur très pratique pour savoir si le code est déjà archivé ou pas

DEMO

- **Save code now**
  - **DEMO**
- L'archivage conduit à l'existence d'un identifiant unique : le **SWHID**
  - **DEMO**

Métadonnées

- **Automatiser au maximum**
  - Utiliser le fichier **codemeta.json** : métadonnées ajoutées ensuite automatiquement sur HAL
  - Vocabulaire CodeMeta qui étend le vocabulaire schema.org
  - Existence d'un outil web pour faciliter la création du fichier : **CodeMeta generator**

- **Métadonnées obligatoires**
  - Nom du logiciel
  - Licences
  - Auteurs

- **DEMO**

Signalement sur HAL

- **Nouveau dépôt**
  - Ne pas déposer de fichier mais indiquer le SWHID
    - Préférer le SWHID avec contexte (pour garder le lien vers l'historique de développement)
- **Vérifier les métadonnées insérées** automatiquement via le fichier codemeta et **compléter les métadonnées** qui ne sont pas encore remplies codemeta
- Vérifier les **auteurs et les affiliations**
- **Valider** le dépôt
- **DEMO**

Quelques éléments sur le processus de modération

- **Etapes de modération**
  - Vérification de la **validité du SWHID**
  - Vérification sur SWH des **éléments sur le logiciel** (parcourir le dépôt)
  - Vérifier les **métadonnées obligatoires** : nom, domaine, licence, auteur
  - Vérifier les **fichiers** README, LICENSE, AUTHORS, codemeta.json sur le repository
  - Vérifier les **url** (lien vers la forge, lien vers les DOI si indiqués …)
- A noter : finalisation de la **phase d'expérimentation** via le portail INRIA et ouverture progressive à tous les portails en 2023.
  - La modération nécessite des moyens humains

Liens codes, données et publications

- **Référencement**
  - Des publications sur HAL / autres éditeurs
  - Des données sur Recherche Data Gouv / autres entrepôts
  - Des codes sur Software Heritage et HAL
- Nécessité de **faire le lien** entre toutes ces productions

Références

- Create software deposit in HAL: User guide and best practices , https://hal.archives-ouvertes.fr/hal-01872189
- Modérer un dépôt logiciel dans HAL : dépôt source et dépôt SWHID, https://hal.inria.fr/hal-01876705v2
- HOWTO archive and reference your code, https://www.softwareheritage.org/howto-archive-and-reference-your-code/
- Depositing scientific software into Software Heritage, https://www.softwareheritage.org/2018/09/28/depositing-scientific-software-into-software-heritage/
- Tutoriaux vidéos : https://youtube.com/playlist?list=PLD2VqrZz2-u3bOWtoCoBIh5Flt6iYXsq3
- Save code now : https://archive.softwareheritage.org/save/

- Merci à Morane Gruenpeter et à Sabrina Granger de Software Heritage !