Software Source Code Identification Working Group
SCID WG

Morane Gruenpeter
Inria, Software Heritage

16th September 2020 - RDA France annual Meeting
Software Source Code Identification Working Group

The SCID WG Goal: **capture and analyze** the software identification state-of-the-art in the scholarly ecosystem

**Co-chairs**
- Roberto Di Cosmo
- Martin Fenner
- Daniel S. Katz

**Secretariat Liaison**
- Stefanie Kethers

**RDA page**
https://www.rd-alliance.org/groups/software-source-code-identification-wg

**Repository**
https://github.com/force11/force11-rda-scidwg

**Chronology...**
03/2018 Spawned at RDA P11 in Berlin from the
- RDA Software Source Code IG &
- FORCE11 Software Citation Implementation WG
10/2018 - TAB endorsement

4/2019 - RDA P13, Philadelphia
- WG kick-off

10/2019 - **FORCE2019**, Edinburgh  Full day hackathon on research software

03/2020 - RDA VP15 session online

07/2020 - Output in community review  **DOI:10.15497/RDA00053**
Output: Use cases and identifier schemes for persistent software source code identification

Authors of the SCID WG output (alphabetical order by name)

- Alice Allen, Astronomy Source Code Library & U. Maryland, USA
- Anita Bandrowski - University of California San Diego, USA
- Peter Chan - Stanford University Libraries, California, USA
- Roberto Di Cosmo - Software Heritage, Inria and University of Paris, France
- Martin Fenner - DataCite, Germany
- Leyla Garcia - ZB MED Information Centre for Life Sciences
- Morane Gruenpeter - Inria, Software Heritage, France
- Catherine M Jones - UKRI STFC, UK
- Daniel S. Katz - University of Illinois at Urbana-Champaign, USA
- John Kunze - California Digital Library, University of California, USA
- Moritz Schubotz - swMATH, FIZ Karlsruhe, Germany
- Ilian T. Todorov - UKRI STFC Daresbury Laboratory, UK
- And the participants of the SCID WG (listed in Appendix B)

Editor: Morane Gruenpeter - Inria, Software Heritage, France
Output structure

- Introduction
  - The SCID WG
- Definitions
  - Actors in the scholarly ecosystem
  - What do we want to identify or the granularity of software?
  - What is at stake
- Use cases
  - Classified into one of the following actions: archiving, referencing, describing, citing
- Identifiers schemas
  - Intrinsic identifiers
  - Extrinsic identifiers
- Summary of findings
- Conclusion
Definition: Actors

Software stakeholders in the scholarly ecosystem
Identification target - what do we want to identify?

Software concept / project / collection
Description in registry, a homepage or any other form of metadata record
- Project versions (for example Python2 and Python3)
- Modules
- Sub-modules

Software artifact
- Executable (download link)
- Software source code
  - Dynamic artifact - current development code (on collaborative development platform)
  - Archived copy
    - Snapshot (all branches, all dev history)
    - Release / Package
    - Commit- a specific point in development history
    - Directory
    - File
    - Algorithm

Software context
- Complementary artifacts - Software artifacts that are external to the source code
  - the software environment, tutorial (Jupyter notebook), Data (input/output data), etc.
- Articles
- Documentation
What is at stake

[Archive]

ensure (research) software artifacts *are not lost*

[Reference]

ensure (research) software artifacts *can be precisely identified*

[Describe]

make it easy to *discover / find* (research) software artifacts

[Credit]

ensure *proper credit* is given to authors
## The use cases collection (a small excerpt)

<table>
<thead>
<tr>
<th>Actor</th>
<th>Use case description</th>
<th>Action</th>
<th>Identification target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>Identify all the software artifacts I hold</td>
<td>Archiving, referencing</td>
<td>Release and smaller artifacts</td>
</tr>
<tr>
<td>Citation manager</td>
<td>Curate the software citation entries</td>
<td>Credit</td>
<td>Project, release</td>
</tr>
<tr>
<td>Curator / librarian / digital archivist</td>
<td>Catalog and browse the development history of legacy software source code for preservation purposes (The Apollo mission source code is a good scenario on how making code available on GitHub isn’t enough for persistence purposes)</td>
<td>Archiving</td>
<td>Project, release and smaller artifacts depending on the reference</td>
</tr>
<tr>
<td>Publisher</td>
<td>Create/retrieve identifiers quickly for use in the paper for all software including commercial packages.</td>
<td>Referencing, describing</td>
<td>Any item (all granularity levels)</td>
</tr>
<tr>
<td>Registry</td>
<td>Identify and curate the software entries I hold</td>
<td>Archiving, referencing, describing, credit</td>
<td>Project</td>
</tr>
<tr>
<td>Researcher as a software user (RSU)</td>
<td>Access and use SSC no longer available on a collaborative platform</td>
<td>Archiving</td>
<td>Snapshot, release, revision, directory</td>
</tr>
</tbody>
</table>
Intrinsic identifier: the Software Heritage ID (SWHID)

- **Intrinsic**: compute a unique digital fingerprint
- **Decentralised**: do not need a registry, only agreement on a standard
- **Cryptographically strong** identifiers
Extrinsic identifier: the HAL ID

[Describe][Cite]

Software Heritage

[Archive][Reference]
<table>
<thead>
<tr>
<th>Granularity level (GL)</th>
<th>ID target</th>
<th>Extrinsic identifiers</th>
<th>Intrinsic identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ASCL</td>
<td>ARK</td>
</tr>
<tr>
<td>GL1</td>
<td>project</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GL2</td>
<td>project version</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GL3</td>
<td>module</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GL4</td>
<td>repository</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GL5</td>
<td>repository snapshot</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GL6</td>
<td>release</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GL7</td>
<td>commit</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GL8</td>
<td>directory</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GL9</td>
<td>file</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GL10</td>
<td>Code fragment</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Next steps

- **Version 2** of the SCID WG output will be published integrating comments
- The working group has now **completed** its work
- Maintenance of the SCID output transfers to the **SSC IG**
- Related groups on Software:
  - RDA, ReSA and FORCE11 [FAIR for Research Software Working Group](https://fair4rs.org) (FAIR4RS WG)
    - Launched in July
    - Welcome to join the work defining FAIR principles for research software
  - EOSC software infrastructures task force (SIRS TF)
    - Publish recommendations in December
  - RDA [Software Source Code Interest Group (SSC IG)](https://www.rd-alliance.org/node/8147)
    - Ongoing IG since 2017
  - FORCE11 [Software Implementation Working Group](https://force11.net/sciwg) (SCIWG)
    - Ongoing WG about software citation