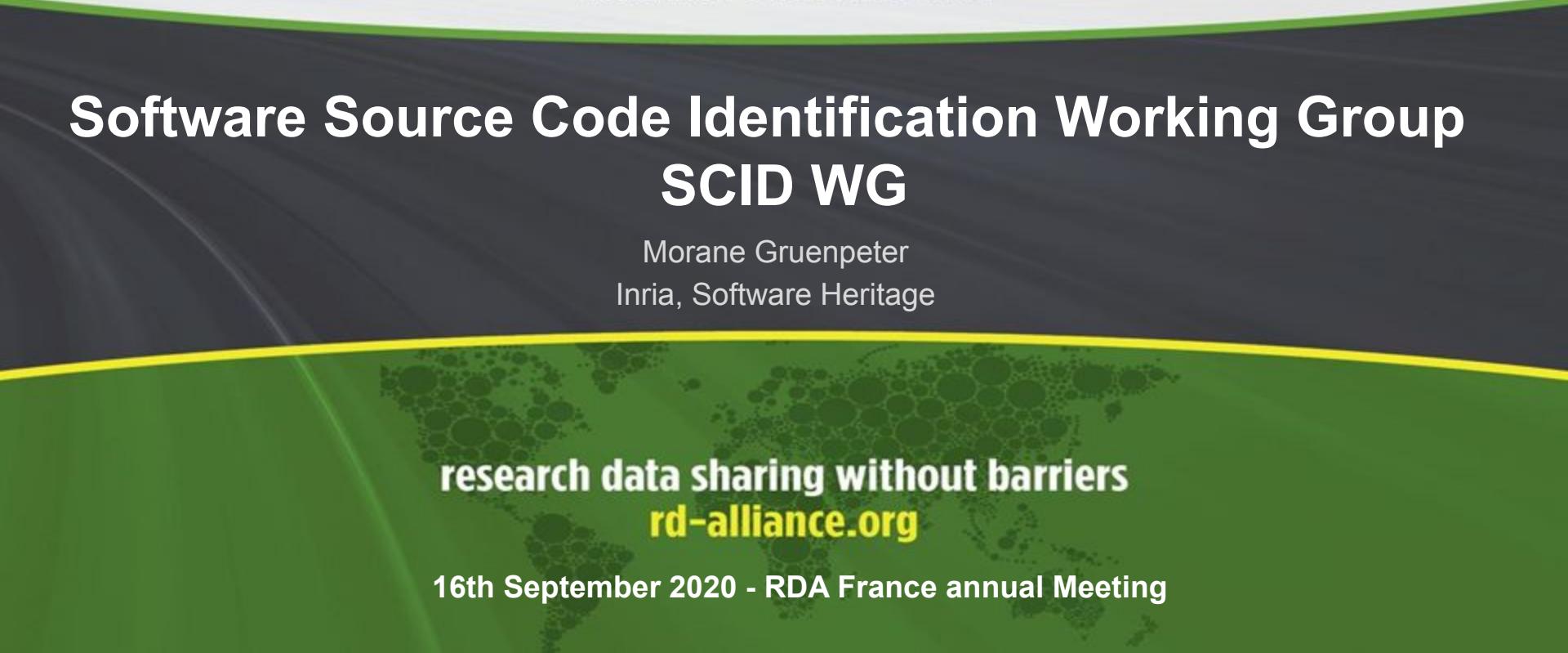




Software Source Code Identification Working Group SCID WG

Morane Gruenpeter
Inria, Software Heritage



research data sharing without barriers
rd-alliance.org

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)

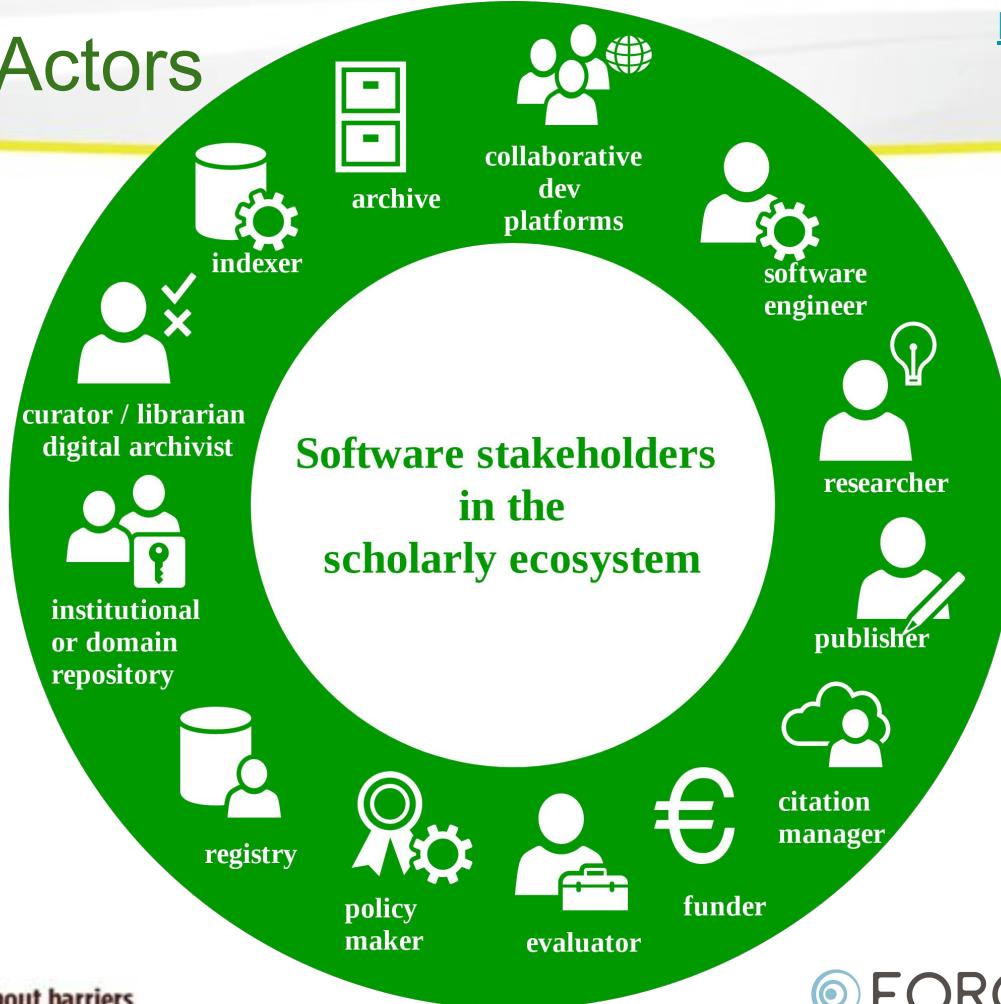
- Alice Allen, Astronomy Source Code Library & U. Maryland, USA
- Anita Bandrowski - University of California San Diego, USA
- Peter Chan - Stanford University Libraries, California, USA
- Roberto Di Cosmo - Software Heritage, Inria and University of Paris, France
- Martin Fenner - DataCite, Germany
- 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



Identification target - what do we want to identify?

Software concept / project / collection

Description in registry, a homepage or any other form of metadata record

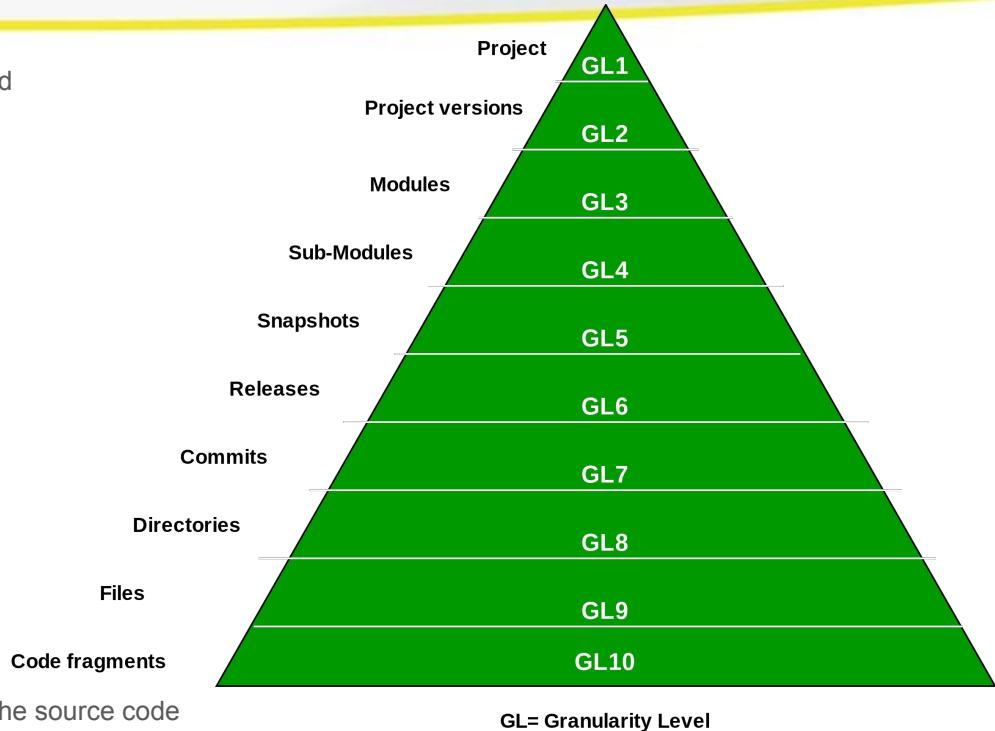
- Project versions (for example Python2 and Python3)
- Modules
- Sub-modules

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



GL= Granularity Level

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)

Actor	Use case description	Action	Identification target
Archive	Identify all the software artifacts I hold	Archiving, referencing	Release and smaller artifacts
Citation manager	Curate the software citation entries	Credit	Project, release
Curator / librarian / digital archivist	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)	Archiving	Project, release and smaller artifacts depending on the reference
Publisher	Create/retrieve identifiers quickly for use in the paper for all software including commercial packages.	Referencing, describing	Any item (all granularity levels)
Registry	Identify and curate the software entries I hold	Archiving, referencing, describing, credit	Project
Researcher as a software user (RSU)	Access and use SSC no longer available on a collaborative platform	Archiving	Snapshot, release, revision, directory

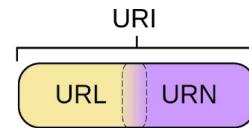
Identifiers schemas



HAL-ID



Digital Object Identifier



Wiki Item identifier (Qxxx)

ASCL.net

Astrophysics Source Code Library

ARK

Archival Resource Key

Software Identification

Handle

Handle System identifiers



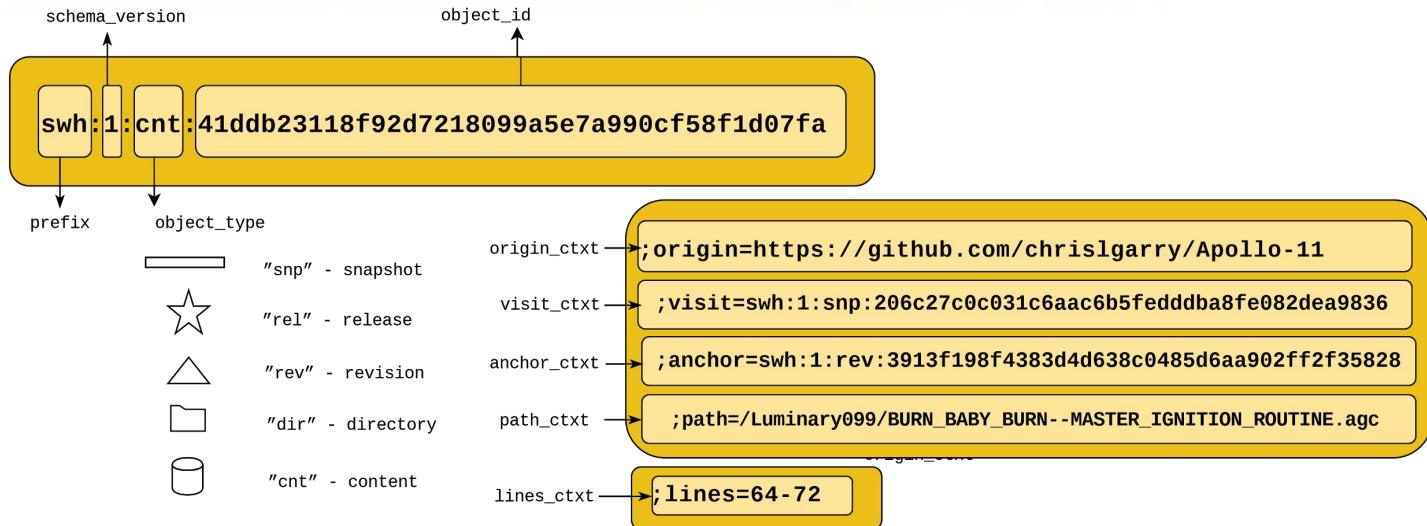
research data sharing without barriers
rd-alliance.org

 FORCE11
The Future of Research Communications and e-Scholarship

 RDA
RESEARCH DATA ALLIANCE

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

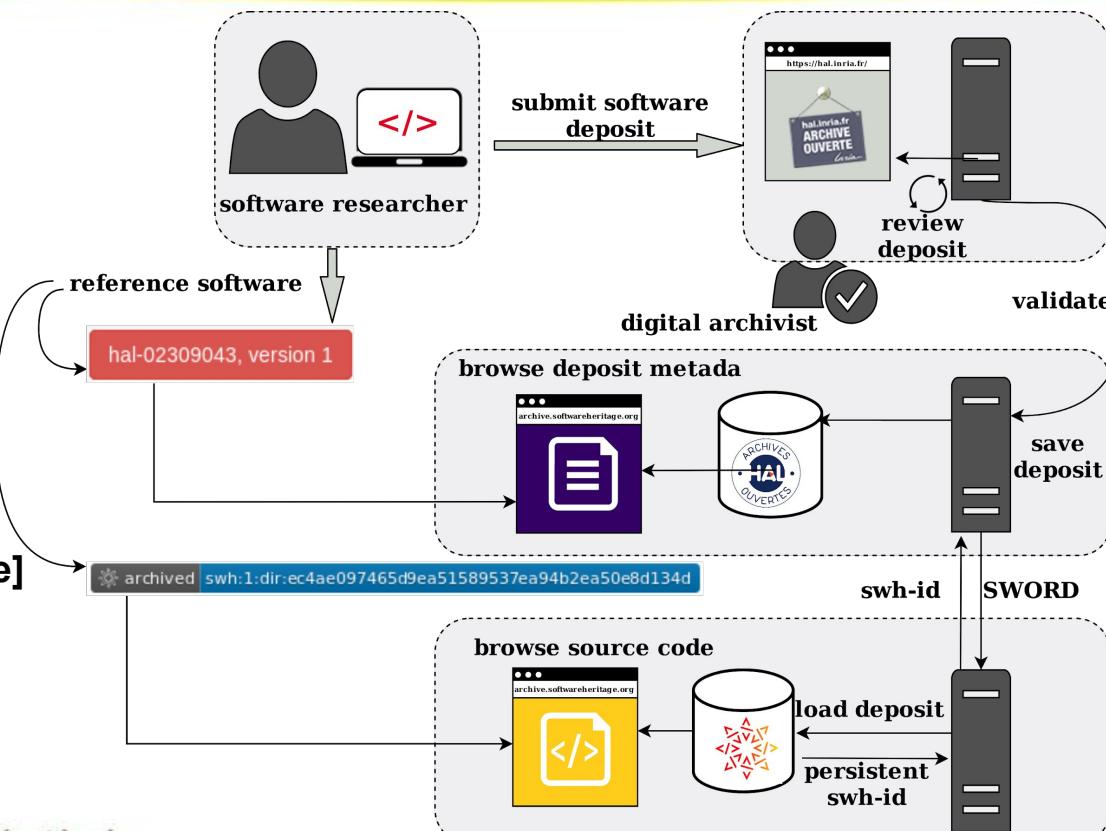


[Describe][Cite]



Software Heritage

[Archive][Reference]



[Deposit guide](#)

Summary

Granularity level (GL)	ID target	Extrinsic identifiers								Intrinsic identifiers		
		ASCL	ARK	DOI	HAL	URL	RRID	SwMath	Wikidata		Hash	SWHID
		entity	property									
GL1	project	X	X	X	X	X	X	X	X			
GL2	project version		X							X		
GL3	module		X							X		
GL4	repository		X			X				X		
GL5	repository snapshot		X							X		X
GL6	release		X	X						X	X	X
GL7	commit		X							X	X	X
GL8	directory		X	X	X*					X	X	X
GL9	file		X	X							X	X
GL10	Code fragment		X									X

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](#) (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 \(SSC IG\)](#)
 - Ongoing IG since 2017
 - FORCE11 [Software Implementation Working Group](#) (SCIWG)
 - Ongoing WG about software citation