Dagstuhl - SWH SPECS

https://hedgedoc.softwareheritage.org/fc4e-wp6-t6-2-specs-template?edit

Component	*Dagstuhl software source code and metadata deposit*
Category	***RSAC***
Contact person	*Michael Wagner, Ramy-Badr Ahmed*
Email address	*michael.wagner@dagstuhl.de , ramy.ahmed@dagstuhl.de*
Contributors	*Ramy-Badr Ahmed*
Version	*1.0*
Data	

Overview

Develop APIs and connectors between Dagstuhl and Software Heritage to support:

- *Archival: automate the archival in Software Heritage of the source code of artefacts associated with research articles.*
- *Reference: expose the corresponding SWHID on the journal's publication record.*
- *Description: enable the deposit and retrieval in Software Heritage of curated metadata for software associated to publications.*
- *Citation: deposit and retrieve the preferred citation information for software associated with publications; export citation information in one or more of the common open citation formats (BibLaTeX, CSL, codemeta.json).*

Objectives

#	*Short description* *<Define 1 or more objectives of the component/service>*
1	*Archival:* • *Ensure and promote publicity of source code of artefacts associated with research articles.* • *Offer instructions on research code submission to either a Forge, a Scholarly Repository or to Dagstuhl directly as a software bundle.* • ***Automate the archival process in Software Heritage of the source code of artefacts associated with research articles.*** • *Build a database of SWHIDs of the associated source code for Dasgstuhl's own record.*
2	*Reference:* • *Ensure that Researchers can retrieve an identifier for the submitted source code from Dagstuhl.* • *Add a reference to the archived software artefact on research document.*

Objectives	
	• ***Expose the corresponding SWHID on the journal's publication record.***
3	*Description:* • *Perform a metadata check for the submitted source code.* • *Request modification from authors, if necessary, with the aid of codemeta.json-form in order to enhance metadata integrity and completeness.* • *Ensure that modifications in the article's metadata will be deposited to Software Heritage.* • ***Enable the deposit and retrieval in Software Heritage of curated metadata for software associated with publications.***
4	*Citation:* • *Provide guidelines and best practices regarding citation of software so that software is eventually cited with correct attributions by Researchers.* • ***Deposit and retrieve the preferred citation information for software associated with publications.*** • ***Export citation information in one or more of the common open citation formats (BibLaTeX, CSL, codemeta.json).***
5	*Integrate and showcase the Overview and the above Objectives into an operational infrastructure used by Dagstuhl.*

Out of Scope	
#	*Short description*
1	*When source code is not yet publicly available:* *Researchers decide on which path to follow to publicise their source code (URL, DOI or software bundle) then inform Dagstuhl.* *This step will be done by the user and not by the infrastructure*

Requirements

The requirements focus on researchers interacting with publishers, scholarly repositories, and how the publisher interacts with the universal source code archive considering the four SIRS pillars: Archive-Reference-Describe-Citation.

General Terms

- **Researcher**: An individual researcher or a full research team requesting publication.

- **Forge**: Code hosting platform (e.g. Github).

- **Scholarly Repository**: Repository run by a research institution (e.g. HAL).

- **Publisher**: An organisation that prepares submitted research to produce a publication (e.g. Dagstuhl Publishing).

User stories

#	Description of the user story	Reference
		https://hedgedoc.softwareheritage.org/fc4e-wp6-t6-2-specs-template#2-Requirements
1	***As a*** *Researcher* ***I can*** *submit a non-public source code to Dagstuhl* ***so that*** *it will be publicised as Software Origin for the universal source code archive, SWH, and added to Dagstuhl's own record.* ***Summary:*** *Researchers submit articles with source code that is only known to authors.*	*Archive:* Use Case A1
2	***As a*** *Researcher* ***I can*** *submit a publicly available source code to Dagstuhl* ***so that*** *it will be recorded/archived by Dagstuhl in the universal source code archive, SWH, and added to Dagstuhl's own record.* ***Summary:*** *Dagstuhl receives a public software origin from authors where the submitted source is.*	*Archive:* Use Case A2
3	***As a*** *Researcher* ***I can*** *submit a source code that is already SWH-archived to Dagstuhl* ***so that*** *it will be added to the Publisher's own record.* ***Summary:*** *Dagstuhl directly receives an article with a SWHID for the associated source code.*	*Archive:* Use Case A3
4	***As a*** *Researcher* ***I can*** *retrieve an identifier from Dagstuhl* ***so that*** *I can reference the record and source code artefacts on the article's pdf.* ***Summary:*** *Dagstuhl generates a self-contained string consisting of some curated metadata. Researchers retrieve an identifier from Dagstuhl to reference the record and source code artefacts.*	*Reference:* Use Case R1
5	***As a*** *Publisher* ***I can*** *add a reference to the archived software*	*Reference:* Use Case R2

User stories		
	*artefact on document **so that** the related software is clearly linked to the text.* ***Summary:** Dagstuhl adds reference on research articles to clearly map the software artefact to the submitted text.*	
6	***As a** Researcher I can describe my software using software specific metadata **so that** the software is findable and reusable.* ***Summary:** Dagstuhl provides codemeta.json-form in order to enhance metadata integrity and completeness.*	*Describe: Use Case D1*
7	***As a** Researcher **I can** retrieve a metadata export for a software, **so that** the metadata can be used in other workflows.* ***Summary:** Researchers can retrieve a metadata export for a software deposit on a scholarly repository from Dagstuhl.*	*Describe: Use Case D2*
8	***As a** Researcher **I can** update the metadata on my software, **so that** the metadata can be as accurate and complete as possible.* ***Summary:** Researchers can update the metadata on their software. Dagstuhl requests corresponding metadata updates in SWH.*	*Describe: Use Case D3*
9	***As a** Researcher **I can** retrieve a citation or BibTeX export for a software, **so that** the metadata can be as accurate and complete as possible.* ***Summary:** Researchers can retrieve a citation export for a software deposit from Dagstuhl.*	*Cite: Use Case C1*
10	***As a** Publisher **I can** provide guidelines and best practises regarding citation of software, **so***	*Cite: Use Case C2*

User stories		
	***that** Researchers are **to correctly cite their software..* ***Summary:** Dagstuhl provides best practices and guidelines addressing citation of software. Dagstuhl scans for potentially missing references and/or citations to ensure correct citations by Researchers.*	

User requirements (user = Researchers, Authors)				
#	Short description	Priority H / M / L	Feasibility 1-5	Reference <JIRA issue number> or <URL to external reference>
1	Any Researcher can access software/article metadata on Dagstuhl.			
2	Authors of accepted Papers can take advantage of the prospective services from Dagstuhl.			

- Priority: H=High, M=Medium, L=Low
- Feasibility: marking between 1 - 5, 5 is easy to implement and 1 is very difficult

Functional requirements			
#	Short description	Priority	Reference
1	***Title:** Get information about a software origin stored in SWH.* *Explanation: API utilising SWH "Archive" endpoints.*	H	SWH Endpoints
2	***Title:** Get information about the metadata of some software origin stored in SWH.* *Explanation: API utilising SWH "MetaData" endpoints.*	H	SWH Endpoints
3	***Title:** Trigger Archival to SWH* *Explanation: API utilising SWH "Request archival" endpoints.*	H	SWH Endpoints

- Priority: H=High, M=Medium, L=Low

Non-functional requirements

#	Short description	Priority	Reference
1	Offer instructions on research source code submission and releasing to the public.	M	
2	Provide best practices and guidelines addressing citation of software.	H	

- Priority: H=High, M=Medium, L=Low

Specifications

Architectural design

Design diagram is provided in the "Requirement and diagrams workspace" HedgeDoc WP6 T6.2 link: https://hedgedoc.softwareheritage.org/fc4e-wp6-t6-2-specs-template?both#V-Publication-and-Propagation

See Annexe A: Sequence diagram publisher workflow

This diagram is inspired by the SIRS report, namely the variants described in subsections;

- Publisher Implements Review on Source Code Submitted as a Bundle,
- Source Code Fully Handled on the Author Side,
- Publisher Implements Review on Publicly Available Source Code Hosted on Public Forge

of section 4.3.2.

Functional specifications

#	Short description	Priority	Reference
	Intro to Specs Given the above workflow of the Architectural Design, we aim to provide several back- and front-end services that handle the four SIRS report pillars each. Bundling these services altogether into an operational infrastructure used by Dagstuhl; it should be possible to promote software archiving to SWH and enriching metadata stored in SWH.		<JIRA issue number> or <URL to external reference>
1	Create a new front-end UI automating interaction with SWH through the back-end.	H	Dagstuhl: dblp
2	Validate existence in SWH of submitted source code, namely, repository URL or SWHID.	H	SWH Endpoint: /api/1/origin

Functional specifications

3	Validate latest version in SWH is compatible with the submitted artefact.	H	SWH Endpoints: /api/1/snapshot /api/1/release /api/1/revision
4	Validate the stored metadata in SWH is compatible with that of the submitted artefact, namely, archived repository URL or SWHID.	H	SWH Endpoints: /api/1/raw-extrinsic-metadata /api/1/origin/intrinsic-metadata
5	Trigger Archival of the submitted source code.	H	SWH Endpoints: /api/1/origin/save /api/1/origin/save/webhook/

- Priority: H=High, M=Medium, L=Low

Service specifications

#	Short description	Priority	Reference
1	HTTP Connectors	H	HTTP Protocol Standard
2	User Interface	H	dblp

- Priority: H=High, M=Medium, L=Low

Operational specifications

#	Short description	Priority	Reference
1	MVC Framework (e.g.: Laravel)	H	Laravel
2	Full demo projects to interact with SWH.	M	
3	Bypassing SWH rate-limit.	L	

- Priority: H=High, M=Medium, L=Low

Integration with EOSC Core components

#	Short description	Priority	Reference
1	<Provide a short description of operational specifications>		<JIRA issue number> or <URL to external reference>

Integration with EOSC Core components			
...			

- Priority: H=High, M=Medium, L=Low

External references

External references		
#	Short description	Reference
1	Links will be provided for testing after the system reaches Beta state.	<url>
…		

External references - SWH		
#	Short description	Reference
1	<Provide a short description of external reference>	<url>
…		

Annexe

Annexe A: Sequence diagram publisher workflow