Project Number: 101057264 Start Date of Project: 01/06/2022 Duration: 36 months

Deliverable 1.2

FAIRCORE4EOSC Technical Specifications

Dissemination Level	PU
Due Date of Deliverable	31/01/2023, Month 8
Actual Submission Date	31/01/2023
Work Package	WP 1, Project coordination and strategic alignment
Task	T1.2
Type	R – Document, report
Version	V1.0
Number of Pages	p.1 – p.162

Deliverable Abstract

The FAIRCORE4EOSC project started in June 2022 and will deliver nine new EOSC-Core components in support of a FAIR research life cycle, bridging the gaps identified in the EOSC SRIA. More specifically, the components will enable an EOSC PID infrastructure, an EOSC research software infrastructure, support for sharing and access to metadata schemas and crosswalks and offer advanced research-intent driven discovery services over all EOSC repositories.

This deliverable summarises the overall technical specifications for the FAIRCORE4EOSC components. The deliverable is a snapshot of the input received from WP2, WP3, WP4, WP5, WP6 and WP7 via the milestones at M7. This deliverable will be further updated during the project duration on the project wiki to take into account new requirements and feedback received. The Wiki page that this deliverable is derived from automatically mirrors updates to the content provided by the work packages.

Funded by the European Union

FAIRCORE4EOSC has received funding from the EU's Horizon Euro programme under Grant Agreement no. 101057264.

DELIVERY SLIP

Contribution	Name	Partner/Activity	ORCID
Lead author(s)	**Tommi Suominen**	**CSC**	**0000-0002-4382-7425**
Contributor(s)	**Claudio Atzori**	**CNR**	**0000-0001-9613-6639**
	Miriam Baglioni	**CNR**	**0000-0002-2273-9004**
	Christos Tsapelas	**Athene RC**	**0000-0003-2732-9140**
	George Katsogiannis	**Athene RC**	**0000-0001-8285-5827**
	Serafeim Chatzopoulos	**OpenAIRE**	**0000-0003-1714-5225**
	Thanasis Vergoulis	**OpenAIRE**	**0000-0003-0555-4128**
	Stavroula Eleftherakis	**Athena RC**	**0000-0001-9527-6920**
	Paolo Manghi	**OpenAIRE**	**0000-0001-7291-3210**
	Katerina Iatropoulou	**Athena RC**	**0000-0001-9991-9719**
	Konstantina Galouni	**Athena RC**	**0000-0002-9306-6652**
	Sarala Wimalaratne	**DataCite**	**0000-0002-5355-2576**
	Joonas Kesäniemi	**CSC**	**0000-0002-3770-0006**
	Daan Broeder	**CLARIN**	**0000-0002-8446-3410**
	Sven Bingert	**GWDG**	**0000-0001-9547-1582**
	Chris Ariyo	**CSC**	**0000-0002-8790-4082**
	Hans Lienhop	**GWDG**	**0000-0002-0066-4953**
	Shorn Tolley	**ARDC**	
	Rob Leney	**ARDC**	
	Matthias Liffers	**ARDC**	
	Siobhan MCCafferly	**ARDC**	
	Melanie Barlow	**ARDC**	
	Natasha Simons	**ARDC**	
	Themis Zamani	**GRNET**	**0000-0003-1148-1738**
	Lars Holm Nielsen	**CERN**	**0000-0001-8135-3489**
	Wim Hugo	**DANS**	**0000-0002-0255-5101**
	Wilko Steinhoff	**DANS**	**0000-0003-1106-8441**
	Michael Wagner		
	Willem Elbers	**CLARIN**	**0000-0002-0191-9102**
	Ramy-Badr Ahmed		
	Alain Monteil	**INRIA**	**0000-0003-3150-4837**
	Maud Medves	**INRIA**	**0000-0002-0624-1564**
	Maxence Azzouz	**FIZ**	**0000-0002-8710-9548**
	Moritz Schubotz	**FIZ**	**0000-0001-7141-4997**
	Kyle Snyder	**SURF**	**0000-0001-8416-2442**
	Mark van de Sanden	**SURF**	**0000-0002-2718-8918**
	Clifford Tatum	**SURF**	**0000-0002-2212-3197**
	Heinrich Widmann	**DKRZ**	**0000-0001-9871-2687**
	Anu Märkälä	**CSC**	

Funded by the European Union

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

	Pihla Kauranen	**CSC**	
Reviewer(s)	**Heinrich Widmann**	**DKRZ**	**0000-0001-9871-2687**
	Juha Hakala	**UH**	**0000-0003-1067-5020**

DOCUMENT LOG

Issue	Date	Comment	Author/Editor/Reviewer
v0.1	31/12/2022	Formation of the requirements per component as input to FAIRCORE4EOSC milestones submitted in M7 (December 2022)	Most of those listed as contributors under "Delivery Slip".
v0.2	25/01/2023	Compilation of the requirements from Work Packages into deliverable 1.2 during January 2023	All deliverable contributors listed in the delivery slip of this document
v0.3	27/01/2023	Internal review by two reviewers	Juha Hakala, Heinrich Widmann
v0.4	30/01/2023	Version addressing feedback received from the reviewers	All deliverable contributors listed in the delivery slip of this document
v.1.0	31/01/2023	Final version	Tommi Suominen, Anu Märkälä, Pihla Kauranen

TERMINOLOGY

Terminology/Acronym	Definition
AAI	authentication and authorization infrastructure
AGH/AGH-UST	Akademia Gorniczo-Hutnicza im. Stanislawa Staszica w Krakowie
API	Application Programming Interface
ARDC	Australian Research Data Commons
ATHINA RC	Athina-Erevnitiko Kentro Kainotomias Stis Technologies Tis Pliroforias, Ton Epikoinonion Kai Tis Gnosis
CAT	EOSC Compliance Assessment Toolkit
CERN	Organisation Europeenne Pour La Recherche Nucleair
CFF	Citation File Format
CLARIN	CLARIN ERIC
CNR	Consiglio Nazionale delle Ricerche
CODRA	full name unused (Content Object Repository Discovery and Resolution (or Registration) Architecture)
COMMPLA	COMMPLA SRL
CRIS	Current Research Information System
CSC	CSC-Tieteen Tietotekniikan Keskus OY
CSL	Citation Style Language
DATACITE	Datacite-International Data Citation Initiative EV
DKRZ	Deutsches Klimarechenzentrum GMBH
DOI	Digital Object Identifier
DOIP	Digital object interface protocol
DO-IRP	Digital object identifier resolution protocol
DTR	EOSC Data Type Registry
ENES	The European Network for Earth System Modelling
EOSC	The European Open Science Cloud
FAIR	Findable, Accessible, Interoperable, reusable
FC4E	FAIRCORE4EOSC
FDO	FAIR Digital Objects

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

FIZ	FIZ Karlsruhe - Leibniz-Institut für Informationsinfrastruktur GMBH
GDPR	General Data Protection Regulation
GRNET	National Infrastructures for Research and Technology
GUI	Graphical User interface
GWDG	Gesellschaft Für Wissenschaftliche Datenverarbeitung MBH Göttingen
HTML	HyperText Markup Language
INRIA	Institut National de Recherche en Informatique et Automatique
ISNI	International Standard Name Identifier
JSON	JavaScript Object Notation
KNAW-DANS	Koninklijke Nederlandse Akademie van Wetenschappen
LZI	Schloss Dagstuhl - Leibniz Zentrum für Informatik GMBH
MSCR	EOSC Metadata Schema and Crosswalk Registry
MSCR GUI	(EOSC Metadata Schema and Crosswalk Registry) Graphical User Interface
NDA	Non Disclosure Agreement
NL	natural language
OPENAIRE	OPENAIRE AMKE
PID	Persistent Identifier
PID Graph	EOSC Persistent Identifier Graph
PIDMR	EOSC Persistent Identifier Meta Resolver
RAiD	Research Activity Identifier Service
RDA	Research Data Alliance
RDF	Resource Description Framework
RDGraph	EOSC Research Discovery Graph
REST API	Representational state transfer application programming interface
ROR	Research Organization Registry
RPO	Research Performing Organisation
RSAC	EOSC Research Software APIs and Connectors
SEMAF	Semantic Mapping Framework
SIRS	Scholarly Infrastructures of Research Software
SPA	Single-Page-Application
SRIA	Strategic Research and Innovation Agenda
SURF	SURF BV
SWH	Software Heritage
SWHID	Software Heritage intrinsic identifiers
SWHM	EOSC Software Heritage Mirror
TBD	to be discussed/decided
TF	Task Force
TRL	technology readiness levels
TRUST (principle)	Transparency, Responsibility, User focus, Sustainability and Technology
TRUST-IT	TRUST-IT SRL
TSG	Technical Steering Group
TU Wien	Technishe Universität Wien
UH	University of Helsinki
UI	User interface
UNIWARSAW	Uniwersytet Warszawski
URI (CURIE)	Uniform Resource Identifier (Compact URI)
URL FAQ	Uniform Resource Locator Frequently Asked Questions
URN: NBN	Uniform Resource Name: National Bibliography Number
UX	User Experience
WP	Work Package

Funded by the European Union

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

XML	Extensible Markup Language
zbMATH	formerly known as: Zentralblatt MATH

Funded by the European Union

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Table of Contents

1 Introduction 1
2 FAIRCORE4EOSC components 1
2.1 Integration with the EOSC-Core 4
2.2 Demonstrators Versus Case Studies 4
3 Implementation of the new FAIRCORE4EOSC Components 5
3.1 Requirement collection process and developing technical specifications 6
3.2 Technical Specifications 6
3.3 Requirements Gathering Instructions 7
3.4 Jira Agile Board 7
3.5 Software Management 8
4 User Requirement Analysis Specification Reporting 8
5 WP3 – EOSC Federated Search Service and PID Graph Service 8
5.1 RDGraph - Graph data service 9
5.1.1 Requirements 10
5.1.2 Specifications 11
5.1.3 External references 12
5.2 RDGraph - Natural Language Search 12
5.2.1 Requirements 13
5.2.2 Specifications 14
5.2.3 External references 16
5.3 RDGraph - Impact-based search 16
5.3.1 Requirements 17
5.3.2 Specifications 18
5.3.3 External references 19
5.4 RDGraph - Community recommendation profiles 20
5.4.1 Requirements 21
5.4.2 Specifications 22
5.4.3 External references 24
5.5 RDGraph - RAiD inference service 24
5.5.1 Requirements 25
5.5.2 Specifications 26
5.5.3 External references 27
5.6 RDGraph - RDGraph portals 28
5.6.1 Requirements 29

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

5.6.2 Specifications ... 31
5.6.3 External references ... 33
5.7 PID Graph access via APIs and data dumps ... 33
5.7.1 Requirements ... 34
5.7.2 Specifications ... 35
5.8 PID Graph - Data Usage Statistics ... 37
5.8.1 Requirements ... 38
5.8.2 Specifications ... 39
5.9 PIDGraph - Link Claims ... 41
5.9.1 Requirements ... 42
5.9.2 Specifications ... 44
6 WP4 – EOSC registries for research data and metadata interoperability and research activity tracking 45
6.1 T4.2 MSCR - Specifications ... 46
6.1.1 Requirements ... 47
6.1.2 Specifications ... 50
6.1.3 External references ... 57
6.2 T4.3 DTR - Specifications ... 57
6.2.1 Requirements ... 58
6.2.2 Specifications ... 61
6.2.3 External references ... 65
6.3 T4.4 RAiD - Specifications ... 65
6.3.1 Requirements ... 67
6.3.2 Specifications ... 76
6.3.3 External references ... 80
6.4 T4.5 Demonstrator - Specifications ... 80
6.4.1 MSCR seeding demonstrator T4.5.1 ... 80
6.4.2 MSCR Managing B2FIND Schema and crosswalks T4.5.2 ... 82
6.4.3 MSCR Managing B2SHARE schema T4.5.6 ... 84
6.4.4 Using the DTR for typing metadata schema element and attribute values T4.5.3 ... 85
6.4.5 The DTR as registration authority for extended mime-types T4.5.4 ... 87
6.4.6 B2Inst DTR demonstrator T4.5.5 ... 88
7 WP5 – EOSC PID Meta Resolver and PID Kernel Information Profiles ... 89
7.1 T5.2 PID Meta Resolver - Requirement Analysis ... 89
7.1.1 Requirements ... 90
7.1.2 Specifications ... 93
7.1.3 External references ... 101
7.2 T5.3 PID Kernel Information Profiles - Technical Specifications ... 102

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

7.2.1 Requirements 103
7.2.2 Specifications 103
7.2.3 External references 103
8 WP6 –Services and tools to archive, reference, describe and cite research software 103
8.1 Introduction 103
8.2 RSAC component for scholarly repositories 105
8.2.1 InvenioRDM - SWH SPECS 105
8.2.2 DataVerse - SWH SPECS 123
8.3 RSAC component for Publishers 129
8.3.1 Dagstuhl - SWH SPECS 129
8.3.2 Episciences - SWH SPECS 136
8.4 RSAC component for aggregators 142
8.4.1 swMATH - SWH SPECS 142
8.4.2 OpenAIRE - SWH SPECS 147
9 WP7 – User Requirements from the Case Studies 151
9.1 Climate Case Study 151
9.1.1 Specifications 153
9.1.2 External references 153
9.2 Mathematics Case Study 154
9.2.1 Requirements 154
9.2.2 Specifications 156
9.2.3 External references 156
9.3 Service Providers and Research Data Management Communities Case Study 157
9.3.1 Specifications 158
9.3.2 External references 159
9.4 Social Sciences and Humanities Case Study 159
9.4.1 Requirements 161

List of Figures

Figure 1 – Architecture overview of the new FAIRCORE4EOSC components in relation to the EOSC-Core .. 2
Figure 2 – FAIRCORE4EOSC Implementation Methodology 5
Figure 3 11
Figure 4 15
Figure 5 18
Figure 6 23

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Figure 7 26
Figure 8 31
Figure 9 36
Figure 10 36
Figure 11 40
Figure 12 40
Figure 13 44
Figure 14 44
Figure 15 51
Figure 16 52
Figure 17 53
Figure 18 53
Figure 19 62
Figure 20 62
Figure 21 63
Figure 22 94
Figure 23 95
Figure 24 – Expanded diagram of a architecture of interconnected scholarly infrastructures supporting archival, reference, description and citation of (research) software source code from SIRS report[1] 104
Figure 25 110
Figure 26 111
Figure 27 112
Figure 28 113
Figure 29 – Sequence diagram publisher workflow 134
Figure 30 141
Figure 31 – Sequence diagram - swMath to SWH 145
Figure 32 151
Figure 33 157
Figure 34 158
Figure 35 159

List of Tables

Table 1 – FAIRCORE4EOSC EOSC-Core components 2

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Executive Summary

The European Open Science Cloud (EOSC), one of the pillars for the realisation of the European Strategy for Data, is an ecosystem of research data and related services that will enable and enhance seamless access to and reliable re- use of FAIR research outputs (including data, publications, software, etc.). One of the priorities highlighted in the EOSC Strategic Research and Innovation Agenda (SRIA) is the establishment of the Web of FAIR data and a Minimum Viable EOSC by 2027, featuring the core components and functions to enable EOSC to operate. The development of the EOSC-Core was initiated in the Horizon 2020 calls which have delivered a rich palette of use cases, demonstrations, data, services and tools. However, there are still challenges that require to be addressed.

The FAIRCORE4EOSC project started in June 2022 and will deliver nine new EOSC-Core components in support of a FAIR research life cycle, bridging the gaps identified in the EOSC SRIA. More specifically, the components will enable an EOSC PID infrastructure, an EOSC research software infrastructure, support for sharing and access to metadata schemas and crosswalks and offer advanced research-intent driven discovery services over all EOSC repositories.

This deliverable summarises the overall technical specifications for the FAIRCORE4EOSC components. The deliverable is a snapshot of the input received from WP2, WP3, WP4, WP5, WP6 and WP7 via the milestones at M7. This deliverable will be further updated during the project duration on the project wiki to take into account new requirements and feedback received. The Wiki page that this deliverable is derived from automatically mirrors updates to the content provided by the work packages.

Funded by the European Union

FAIRCORE4EOSC has received funding from the EU's Horizon Euro programme under Grant Agreement no. 101057264.

1 Introduction

The European Open Science Cloud (EOSC)[1], one of the pillars for the realisation of the European Strategy for Data[2], is an ecosystem of research data and related services that will enable and enhance seamless access to and reliable re- use of FAIR[3] research outputs (including data, publications, software, etc.). One of the priorities highlighted in the EOSC Strategic Research and Innovation Agenda[4] (SRIA) is the establishment of the **Web of FAIR data** and a **Minimum Viable EOSC** by 2027, featuring the core components and functions to enable EOSC to operate. The development of the EOSC-Core was initiated in the Horizon 2020 calls which have delivered a rich palette of use cases, demonstrations, data, services and tools. However, there are still challenges that require to be addressed.

The FAIRCORE4EOSC project started in June 2022 and will deliver nine new EOSC-Core components in support of a FAIR research life cycle, bridging the gaps identified in the EOSC SRIA. More specifically, the components will enable an EOSC PID infrastructure, an EOSC research software infrastructure, support for sharing and access to metadata schemas and crosswalks and offer advanced research-intent driven discovery services over all EOSC repositories.

This report summarises the overall technical specifications for the FAIRCORE4EOSC components. The report is based on the input received by WP2, WP3, WP4, WP5, WP6 and WP7 via the milestones at M7. This deliverable will be further updated during the project duration on the project wiki to keep into account new requirements and feedback received.

2 FAIRCORE4EOSC components

The FAIRCORE4EOSC project focuses on the development and realisation of EOSC-Core components supporting a FAIR EOSC, addressing gaps identified in the SRIA. These components will extend the EOSC-Core platform with capabilities to allow researchers, research communities, and other stakeholders of EOSC to increase FAIRness within their practices. Leveraging existing technologies and services, the project will develop nine new EOSC-Core components aimed to improve the discoverability and interoperability of an increased amount of research outputs.

Figure 1 shows an architecture overview of the components to be developed in FAIRCORE4EOSC (green boxes) in relation to the existing EOSC-Core (yellow boxes).

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

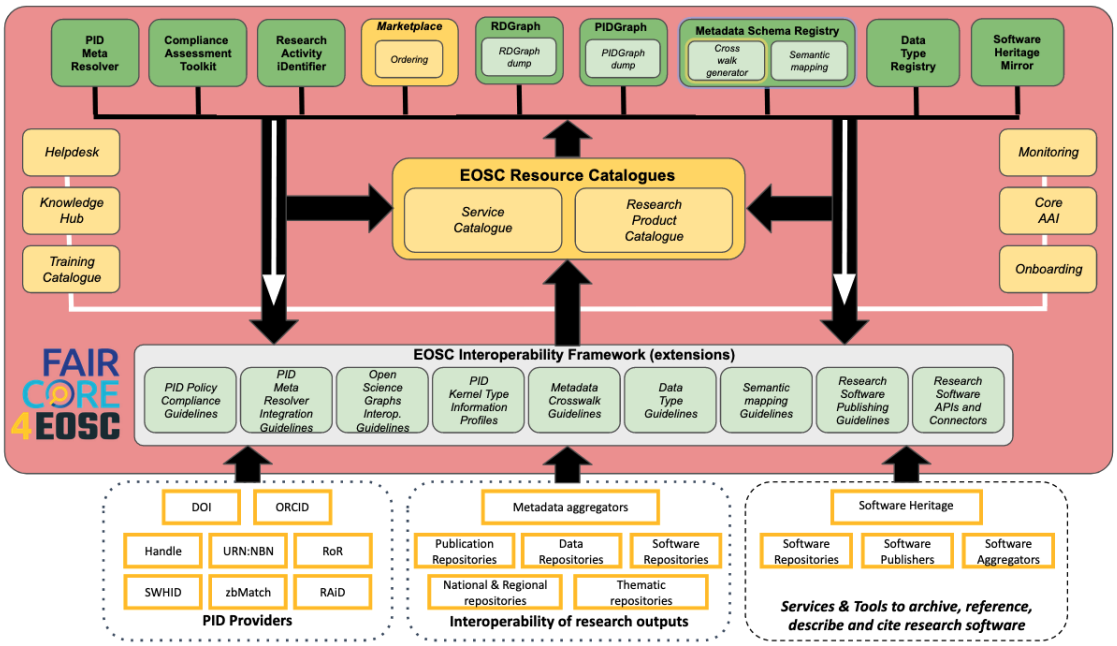

Figure 1 – Architecture overview of the new FAIRCORE4EOSC components in relation to the EOSC-Core

Table 1 provides a short description of the components being developed.

Table 1 – FAIRCORE4EOSC EOSC-Core components

Core Component	Description
EOSC Research Discovery Graph	EOSC Research Discovery Graph (RDGraph) is a flexible and federated EOSC search service across EOSC repositories that extends EOSC Research Catalogue making it compatible with the specifications provided by the RDA's "Open Scientific Graph for FAIR Data" working group and incorporating additional entities like the Research Activity Identifiers (RAiDs). In its core, it is based on OpenAIRE Research Graph and it will become exposed through its APIs and data dumps. In addition, RDGraph offers a variety of advanced functionalities that leverage intelligent community-oriented discovery tools developed by the partners of the FAIRCORE4EOSC project. The respective toolset includes a tool to support querying the RDGraph contents using natural language, a tool to facilitate the discovery of EOSC products and services exploiting their scientific impact, a tool to create user community profiles capturing the preferences of the respective groups of users, and a tool to automatically discover the existence of potential RAiDs in the RDGraph content. Finally, RDGraph makes available the previous services through a set of carefully designed discovery and management portals.
EOSC PID Graph	Services for providing access to the PID Graph, which is made up of links and records gathered from persistent identifier (PID) authority data sources. PID metadata access APIs, software components supporting Open Science graph interoperability (sharing of graph data), and extension of the authoritative sources enabling links between PID entities are some of the services that will be provided.
EOSC Metadata Schema and	Support publishing, discovery and access of metadata schemas and provide functions to operationalise metadata conversions by combining crosswalks.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Core Component	Description
Crosswalk Registry	
EOSC Data Type Registry	Provide user friendly and machine actionable Interfaces for the registration and usage of Data Types and Kernel Information Profiles. Data Types promote the standardisation of meta data attributes and thus accelerate the reuse of referenced objects. Kernel Information Profiles set a standard for lists of Data Types to be expected along with a PID.
EOSC PID Meta Resolver	Provides users with a common interface to resolve different types of PIDs regardless of their originating system. The PIDMR either resolves to a given URI or provides Kernel Information Profiles if available.
EOSC Compliance Assessment Toolkit	The Compliance Assessment Toolkit will support the EOSC PID policy with services to encode, record, and query compliance with the policy. To do so, a wide range of compliance requirements (TRUST, FAIR, PID Policy, Reproducibility, GDPR, Licences) will be evaluated as use cases for definition of a conceptual model and supporting service specifications (the framework), followed by development and testing of operational services for PID Policy Compliance monitoring. Though primarily aimed at machine-actionable operations, the API-based services will be complemented by user interfaces to broaden its use.
EOSC Research Activity Identifier Service	The EOSC RAiD will mint PIDs for research projects, which will allow authorised EOSC users and services to manage information about project-related participants, services, and outcomes. The EOSC RAiD will thereby collect the relationships between research objects, which enriches analysis, tracking, and reporting (including EOSC service utilisation), and indirectly supports reproducibility and extends the ability to discover research entities in the EOSC RDGraph/PIDgraph.
EOSC Research Software APIs and Connectors	Ensure the long-term preservation of research software in different disciplines. APIs and connectors will be developed to interconnect research outputs infrastructures with the Software Heritage *universal source code archive*, using the CodeMeta standard, and the Software Heritage intrinsic identifiers (SWHID). The development of the sub-components is divided by type the of the infrastructure: *scholarly repositories*, *publishers* and *aggregators*.
EOSC Software Heritage Mirror	Equip EOSC with a mirror of the Software Heritage universal source code archive. In order to prevent information loss, a mirror of Software Heritage will be established by GRNET to serve the EOSC community and will be updated regularly to follow the growth of the universal source code archive.

In terms of technology readiness levels (TRLs), the ambition of FAIRCORE4EOSC as a whole is to reach at least TRL 8 (system complete and qualified). FAIRCORE4EOSC does not start developing the new EOSC-Core components from scratch. It builds upon underpinning technologies developed in previous projects, communities, infrastructures and/or within the open-source community. These technologies vary between TRL 5 (technology validated in relevant environments) and TRL 6 (technology demonstrated in relevant environments).

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

2.1 Integration with the EOSC-Core

EOSC-Core offers a set of tools that supports services from the operational perspective. The FAIRCORE4EOSC components will be seamlessly integrated with the existing EOSC-Core services so that they can offer a consistent user experience to the end user. It is expected that all the FAIRCORE4EOSC services will be integrated with EOSC Core AAI for both user and system to system authentication and authorisation. The necessary automated checks and the corresponding metrics will be developed to verify the behaviour of each new EOSC-Core service according to its specification. These metrics will be made part of the regular checks performed by EOSC-Core monitoring in order to ensure the availability and reliability of the newly developed services. GRNET will coordinate and implement these integration activities with the support of AGH/AGH-UST. GRNET and AGH/AGH-UST will work in close collaboration with the components' development teams to draft the integration specifications of the individual components and will leverage the project technical coordination meetings to identify synergies and common cross-component integration activities. The high-level strategy for the integration of the FAIRCORE4EOSC components with the EOSC-Core will be discussed in the TSG.

The FAIRCORE4EOSC project will also closely follow the evolution of the EOSC Interoperability Framework that will be extended through the EOSC Future project and the EOSC Association TFs. The resulting interoperability guidelines will be adapted and similar guidelines will also be published for the new components, as required. The EOSC interoperability framework and guidelines provided as a part of EOSC Future project results, will be followed in all identified interoperability areas. All the new components developed in scope of the project, will be supported with the appropriate representation of their respective resources/entities in the currently existing services supporting EOSC end-users as EOSC Marketplace, EOSC Portal and other components of the currently evolving EOSC Platform (a part of the EOSC Future project).

FAIRCORE4EOSC will provide new EOSC-Core services to be used in the registration of new research products (e.g., metadata schemas, crosswalks, semantic mappings, data types, research activities and compliance information). The project will ensure that these new research products are collected within the Research Product Catalogue and made available through the RD- and PIDGraphs. The EOSC RAiD service will be integrated into the EOSC Marketplace: this will facilitate the granting of resources within EOSC and, thanks to the RAiD integration into the RDGraph, researchers will be able to add discovered resources to an existing RAiD record.

2.2 Demonstrators Versus Case Studies

Five case studies will be implemented during the project to provide the development teams with user-driven requirements and to act as testbeds for the components. Each case study will use multiple components developed by the project to validate their functionalities and provide user feedback. Case studies will also be instrumental to produce long-term benefits to their users, ensuring sustainability of the integration of the new components. The case studies will not be the only validation instrument adopted by the project. A set of demonstrators will also be implemented. In contrast to the case studies, the demonstrators test single components and/or capabilities. For example, populating a new component with data from the involved communities. In addition, the demonstrators are closely connected to the development process. Therefore, they are performed from within the work package in which the development of the component is being conducted. While the case studies run for the entire duration of the project,
the demonstrators have a shorter duration (M4-M24) to allow for the explicit validation of the single component/capability to enable the production release. The combination of demonstrators and case studies provide the development teams with the feedback necessary to deliver functional user-driven components.

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

3 Implementation of the new FAIRCORE4EOSC Components

The development of the new FAIRCORE4EOSC components revolves around four main implementation phases as depicted in the following Figure.

Phase 1: Collection of requirements and technical specifications
Phase 2: BETA release
Phase 3: Cross-integration BETA release
Phase 4: Production release and final documentation

All components to be developed will share this joint development timeline. This approach will i) guarantee reactivity and agility to potential changes in the requirements/specifications; ii) detect potential incompatibilities between developed components at the earliest possible point in time; iii) ensure that the integration of each component to the EOSC-Core stays on track; iv) trigger continuous communication between the different component implementation teams; v) facilitate the sharing of intermediate results with the whole EOSC ecosystem. The development teams working on the different components will be supported by teams working on cross-component case studies driven by target user communities and a Technical Steering Group (TSG) in charge of the steering and monitoring of the overall FAIRCORE4EOSC technical roadmap.

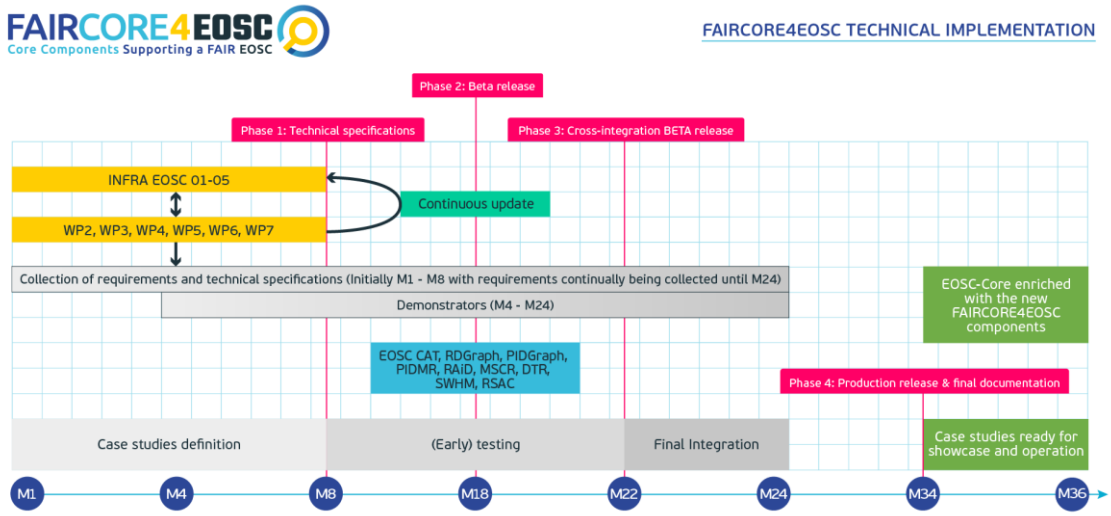

Figure 2 – FAIRCORE4EOSC Implementation Methodology

The first months of the project (M1-M8) will be dedicated to the collection and analysis of requirements and the production of the initial set of technical specifications for all the FAIRCORE4EOSC components (captured in this D1.2). An initial set of technical specifications is needed to kick-start the development of the components. After this initial release, a continuous requirement collection process will be put in place to make sure that all feedback is considered by the technical team to improve the components. Different methodologies will be utilised to collect requirements:

- Desk research: Analysis of studies providing past requirements gathering efforts (e.g., EOSC SIRS report provides a full set of requirements and recommendations for the development of tools and services for archival, reference, description and citation of research software artifacts);
- Internal meetings and milestone MS36 (due at M6) to collect requirements from the five FAIRCORE4EOSC case studies;

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

- Consultation: Dialogue with topical expert groups and supporters from across the communities. The expert groups are key in making the components successful in terms of adequate functionality and acceptance and adoption by the communities. Examples include the work done in the Research Data Alliance (RDA) for metadata schemas and the data type registries, and the work done by the BioMed communities and others in relation to semantic mappings, as investigated in the SEMAF report.

- Collaboration: Partnering with the FAIR-IMPACT project which will align FAIR practices on metadata, PIDs, metrics, etc. and promote their wide uptake. Collaboration will provide FAIRCORE4EOSC with access to research communities, repositories and other target stakeholders that could be key for the co-design of the FAIRCORE4EOSC components.

- Participation: The involvement of the technical partners in the new EOSC Association TFs (See 3.2).

The requirements will be collected into a joint project wiki and ticket system. A requirements assessment process will be established within the TSG as a cross-work package activity. The requirements are assessed and prioritised by the technical teams on the basis of feasibility, impact and effort. Based on the collected requirements, the technical teams will produce specifications for implementation at the component level. The component level specifications available on the project wiki, will also serve as technical documentation.

3.1 Requirement collection process and developing technical specifications

Since the beginning of the project, development teams and case study teams have been scoping the technical specifications and user requirements needed to start the development of the new EOSC Core Components. This design work was carried out with different methodologies and collaborative groups. Teams developed requirements with colleagues from within and across WP's as well as with other EOSC projects to leverage expertise from the EOSC community.

Outputs from these different methodologies have been formalised using a package of requirements assets. The requirements assets were designed in consultation between the Project Management Board, Technical Steering Group and WP task leaders, with the following goals: to capture requirements thoroughly and consistently, design the scope of each core component, and identify which functionalities lie outside of scope for each core component. These requirements assets are the Technical Specifications, Requirements Gathering Instructions and Jira agile board, which have been operationalized by the WP task leaders.

3.2 Technical Specifications

A Technical Specifications template was designed in the FAIRCORE4EOSC Confluence Wiki to capture the technical scope of each core component. The template structures the objective and scope of a core component, and elaborates this scope of the components on two levels:

1. *Requirements*. This level explains the user stories and requirements and (non-)functional requirements of each component.

2. *Specifications*. Here the technical design of the system is documented with architectural designs, functional, service and operational specifications as well as the technical considerations for integration of the service with EOSC Core Components and the EOSC platform core infrastructure.

In addition to the descriptions of the requirements and specifications, the Technical Specifications template also defines the priority and feasibility of requirements. These will aide in the analysis of each requirement and specification as teams refine and accept requirements for development in the coming months.

A second template, the Requirements Gathering Instructions, elaborates the context of the user requirements and ensures that the requirements contribute value to the end-users.

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

3.3 Requirements Gathering Instructions

The second asset to formalise the user requirements supplements the Technical Specifications in the form of a Requirements Gathering Instructions template. The Technical Steering Group designed this format (also within the FAIRCORE4EOSC Confluence Wiki) to capture needs, context, constraints, and beneficiaries of each core component.

During the first phase of requirements gathering, this formalisation describes how each requirement:

- Solves problems users encounter
- Fits real-world scenarios (ex: a climate scientist searching for a specific data set)
- Is potentially constrained by time, resources and funding which influence the implementation
- Benefits different users such as individual researchers, service/resource providers, research communities, project communities, private companies, and/or funders.

The requirements are also defined by the importance to the user, the channel used to gather it, who was involved in its design (affiliation) and who can verify it (contact point).

In subsequent phases of requirement gathering which start after the delivery of this D1.2, this template will then formalise dependencies and operational considerations for each requirement, such as which:

- Functional areas are affected by the implementation
- Architectural components are affected by the implementation
- Level of prioritization should be applied
- Time, resource and funding constraints influence how the implementation is carried out
- Target groups are key stakeholders (individual researchers, service/resource providers, research communities, project communities, private companies, and/or funders)

In the last phase of requirements gathering and assessment, the template formalises an estimated effort to implement the overarching design concept and the feature set of which the requirement is part, with an estimated delivery date for the overarching design concept.

3.4 Jira Agile Board

To operationalise the requirements, a Jira agile board was configured for use by all WP's in FAIRCORE4EOSC. The board was customized in collaboration between CSC and SURF to fit the goals of the requirements gathering initiative: to document requirements formally in a thorough and consistent way, and assess their readiness for development. With oversight from the Technical Steering Group, each requirement will be assessed, then refined or accepted for development.

The Jira board follows the previous formalisations mentioned above, with customised fields to capture the content defined in the Technical Specifications and Requirements Gathering Instructions. In addition, the Jira was configured to link dependencies between core components and case studies through Jira platform functionalities. The EOSC Core Components and Case Studies have been mapped to Jira components, and requirement dependencies can be linked between the requirements in the board. The project sees these dependencies as crucial to reflect in the board to identify potential prioritization blocks to the implementation of requirements, but also to identify potential chances for fruitful integration between the core components being developed in other FAIRCORE4EOSC WP's.

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

The Jira board specifically contributes to the Technical Steering Group's assessment of the requirements by enabling easy filtering and searching of Jira fields related to the requirements content such as components, user importance, and priority.

3.5 Software Management

The software packages of the services to be delivered by FAIRCORE4EOSC will be distributed under an Open Source license, following the licensing patterns of preference of the responsible development team, and maintained into public software repositories.

- **Standards and Metadata:** The software repository will contain a README file containing a brief overview of the component and other descriptive metadata following the guidelines from WP6 and community standards.
- **Software Archiving and Preservation:** All software will be made accessible through known software repositories interoperable with SoftwareHeritge to ensure archiving and preservation. Software packages will be published for citation via public repositories like Zenodo.org and/or B2SHARE.

4 User Requirement Analysis Specification Reporting

The next sections contain information on the functional requirements, which are derived from user requirements that come mainly jointly from the Demonstrators and the Case Studies. These Demonstrator and Case studies user requirements are detailed and each have their own section under "T4.5 Demonstrator - Specifications" respectively. Other user requirements, follow from the nature of the components or are a synthesis or generalization of more specific ones. These were inserted directly just before the different functional requirement sections. This requirement description format was chosen to avoid duplicating information and increase its value for future reporting.

5 WP3 – EOSC Federated Search Service and PID Graph Service

This work package is related to realising, deploying, and operating two FAIRCORE4EOSC components:

- The EOSC Research Discovery Graph (RDGraph) and
- The EOSC PID Graph (PIDGraph)

The EOSC Research Discovery Graph (RDGraph) is a flexible and federated EOSC search service across EOSC repositories that extends EOSC Research Catalogue making it compatible with the specifications provided by the RDA's "Open Scientific Graph for FAIR Data" working group and incorporating additional entities like the Research Activity Identifiers (RAiDs). In its core, it is based on OpenAIRE Research Graph and it will become exposed through its APIs and data dumps. In addition, RDGraph offers a variety of advanced functionalities that leverage intelligent community-oriented discovery tools developed by the partners of the FAIRCORE4EOSC project. The respective toolset includes a tool to support querying the RDGraph contents using natural language, a tool to facilitate the discovery of EOSC products and services exploiting their scientific impact, a tool to create user community profiles capturing the preferences of the respective groups of users, and a tool to automatically discover the existence of potential RAiDs in the RDGraph content. Finally, RDGraph makes available the previous services through a set of carefully designed discovery and management portals.

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

The EOSC PID Graph (PIDGraph) is a set of services for providing access to a PID-focused graph, which is made up of links and records gathered from persistent identifier (PID) authority data sources. PID metadata access APIs, software components supporting Open Science graph interoperability (sharing of graph data), and extension of the authoritative sources enabling links between PID entities are some of the services that will be provided.

WP3 will also provide portals and APIs for users/communities to personalize the discovery, monitoring of statistics and access to research products and services in the EOSC catalogu. Finally, it will demonstrate the added-value of such services by showcasing data exchange integration between B2FIND and the RDGraph and PIDGraph via EOSC interoperability frameworks.

5.1 RDGraph - Graph data service

Component	Graph data service
Category	RDGRAPH
Contact person	Claudio Atzori
Contributors	Claudio Atzori Miriam Baglioni
Version	
Data	

Overview
The Graph data service consists of the set of components implementing the workflows and the operations required to generate the RDGraph from the EOSC resource catalogue content, as well as offer these data via appropriate APIs and regular data dumps.

Objectives	
#	Short description
1	The RDGraph data model support the **RAiD** objects
2	**Regular** and **timely** execution of the workflows for the construction, processing, provision of the RDGraph.
3	Implement the specification from the **RDA Open Science Graph for FAIR Data**
4	The RDGraph API: extension of the **REST API** to support EOSC users, services, communities
5	Address the requirements for the activities in T3.3

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

5.1.1 Requirements

User stories		
#	Description of the user story	Reference
1	As a service provider exploiting the data in the RDGraph, I expect to find the information needed to implement my use case	TBD - for the WP3 use cases we can link the respective descriptions
2	As a user of RDGraph, I would like to find the most recent publications, datasets and research software, along with their research context (links).	TBD (project coordinators use case)
3	As a user interested in specific projects, I'd like to leverage the RAiDs available in the RDGraph to (i) identify a specific project or (ii) keep track of project activities, e.g., who is involved, who funded the project, what outputs they produce, and even what tools (e.g., software) they use.	TBD

User requirements				
#	Short description	Priority	Feasibility	Reference
1	T3.3 - Support the development of (a) novel and fair recommendation algorithm(s) for community profiles.	H	3	RDGraph - Community recommendation profiles
2	T3.3 - Support the development of ranking and filtering of research products based on impact indicators	H	2	RDGraph - Impact-based search
3	T3.3 - Support the development of natural language search services over the RDGraph	H	3	RDGraph - Natural Language Search
4	T3.3 - Support the development of (a) novel service for the inference of potential RAiD objects within the RDGraph content.	H	3	RDGraph - RAiD inference service

- 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	The RDGraph data model should be able to accomodate RAiD objects, allowing to represent the relative properties and relationships, in accordance to the specifications provided by WP4.	H	T4.4 RAiD Technical Specifications
…			

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Non-functional requirements			
#	Short description	Priority	Reference
1	To meet the scheduled updates, the RDGraph workflow should be executed on sufficient computational and storage needs.	H	TBD
2	The RDGraph data model and the methodology behind its construction and processing should be documented in a comprehensive resource to help users understand it and simplify its adoption	M	TBD
3	Define a set of procedures to guarantee the highest possible quality of data in the RDGraph	H	TBD
4	The REST API should be as self-explainable as possible and providing appropriate documentation so that to help users on how to use it.	M	TBD

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

5.1.2 Specifications

Architectural design

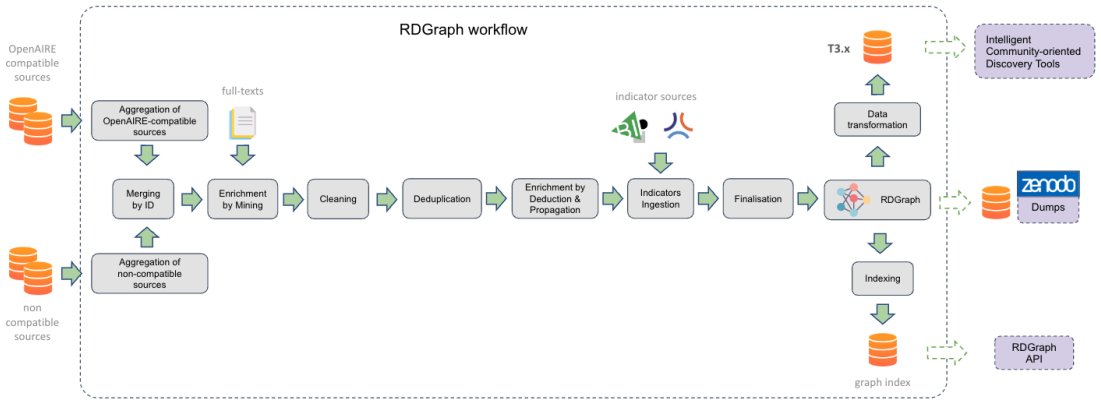

Figure 3

Functional specifications			
#	Short description	Priority	Reference(s)
1	Extension of the RDGraph workflow to produce the different data transformations required by the activities in T3.3	H	RDGraph - Community recommendation profiles RDGraph - RAiD inference service

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
			RDGraph - Natural Language Search RDGraph - Impact-based search
2	Implement the RDGraph data model extension to accomodate the RAiD objects	H	TBD
3	Implement the specification from the RDA Open Science Graph for FAIR Data	H	TBD

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

Service specifications			
#	Short description	Priority	Reference
1	The RDGraph public dump will be published on Zenodo every six months	H	TBD

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

Operational specifications			
#	Short description	Priority	Reference
1	The RDGraph workflow should run on adequate resources to ensure timely and smooth delivery	H	link to document

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

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	The RDGraph constitutes the backbone source of scholarly communication information across the EOSC Core components. Its contents will be made available through different discovery portals.	H	RDGraph - RDGraph portals

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

5.1.3 External references

5.2 RDGraph - Natural Language Search

Component	Natural Language Search

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Category	RDGRAPH
Contact person	Christos Tsapelas George Katsogiannis
Contributors	Christos Tsapelas George Katsogiannis
Version	
Data	

Overview
Natural language search services over the RDGraph. The system will translate the queries defined in natural language into the query language used by the backend engine of the graph.

Objectives	
#	Short description
1	The end users will be able to query the RDGraph using natural language.
2	Translate the user's NL Query into a Query Language executable on the graph.
3	The end users will be able to perform complex queries, like "Find datasets on climate change containing rainfall data".

Out of Scope	
#	Short description
1	The system is going to support only one query language for the RDGraph and there will be no extensions for new query languages.
2	The system is going to accept NL Queries in English only.
3	The system is not going to support schema updates of the graph.

5.2.1 Requirements

User stories		
#	Description of the user story	Reference
1	The user wants to find a certain resource in the RDGraph, but they are not familiar with its structure or ontology. The user can seamlessly pose their question using natural language and receive the desired results.	TBD

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
2	The user wants to refine their query with additional conditions (e.g., “datasets **containing rainfall data**”). They can do so by simply adding their conditions to the natural language question, without the need for additional interfaces or multiple filter toolbars.	TBD

User requirements				
#	Short description	Priority	Feasibility	Reference
1	The generated queries capture the user’s intent accurately	H	3	link
2	Generate queries in an acceptable time limit	M	3	link
3	Support queries with reasoning (hops) over the graph	H	4	link

- Feasibility: marking between 1 - 5, 5 is easy to implement and 1 is very difficult

Functional requirements			
#	Short description	Priority	Reference
1	There is an RDGraph endpoint where our system can execute queries and receive their results	H	TBD
2	There is an endpoint on user interface platform where our system receives user’s NL query	H	TBD

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

Non-functional requirements			
#	Short description	Priority	Reference
1	Our system should provide a well-documented API to simplify its use within FAIRCORE4EOSC project.	M	TBD

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

5.2.2 Specifications

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

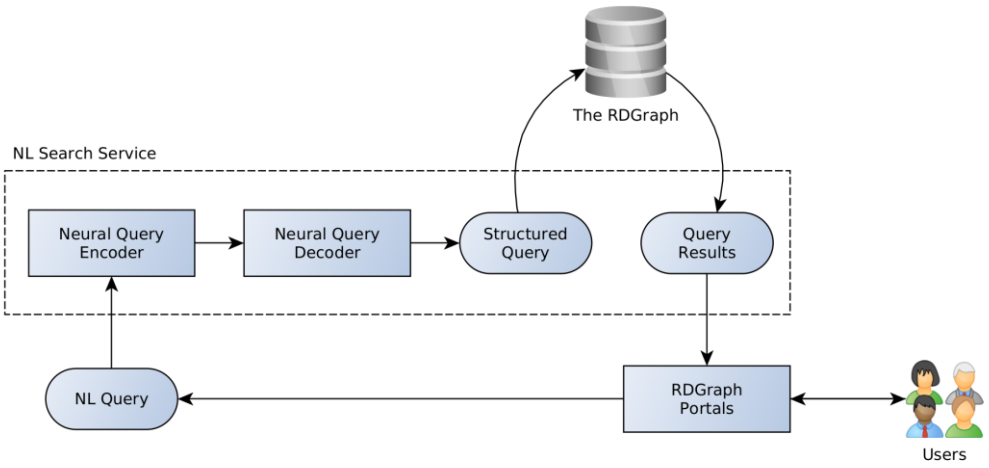

Figure 4

Functional specifications

#	Short description	Priority	Reference
1	Implement an neural NL query encoder that will accept the user's NL query and will create an internal representation for the neural network.	H	TBD
2	Implement an neural query decoder that will predict a structured query output, based on the encoded representation of the user's NL query.	H	TBD
3	Enable query execution on the RDGraph	H	TBD
4	Implement the API for communication with the RDGraph portals	H	TBD

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

Service specifications

#	Short description	Priority	Reference
1	The end users will be able to pose a multitude of queries on the RDGraph simply using natural language.	H	TBD
2	The system will provide the predicted structured query that matches the users's NL query.	M	TBD
3	The system will be able to provide additional predictions for the user's NL query, if the initial prediction is not satisfactory.	L	TBD

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Operational specifications			
#	Short description	Priority	Reference
1	The neural system will require a computational cluster, preferably with a GPU unit, in order to provide timely query predictions.	H	TBD

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

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	The NL Search service will execute queries on the RDGraph in order to provide users with the results that match their NL Query.	H	TBD
2	The NL Search service will accept NL queries that the user poses on the RDGraph portals, and will return the query results to the RDGraph portals.	H	TBD

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

5.2.3 External references

5.3 RDGraph - Impact-based search

Component	Impact-based search
Category	RDGRAPH
Contact person	Serafeim Chatzopoulos
Contributors	Serafeim Chatzopoulos Thanasis Vergoulis
Version	
Data	

Overview
The impact-based search component allows the discovery of research products (and services) incorporating citation and graph analysis techniques. In particular, this component enables ranking and filtering search results based on a variety of impact indicators; these may vary from simple, yet widely established, citation counts to more sophisticated graph analysis techniques, that can estimate the current (popularity) or overall impact (influence) of research products.

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Objectives	
#	Short description
1	Support **ranking** of research products based on impact indicators
2	Support **filtering** of research products based on impact indicators

Out of Scope	
#	Short description
1	We will not consider any indicators based on altmetrics; we will rely on impact indicators based on graph (and citation) analysis.
2	We will not consider any composite indicators or rank fusion techniques based on the provided impact indicators.

5.3.1 Requirements

User stories	
#	Description of the user story
1	As a researcher, I would like to be able to search research products using keywords, and rank/sort them based on different aspects of their impact.
2	As a researcher, I would like to be able to filter search results, keeping only those that are the more impactful based on a specific aspect.

User requirements				
#	Short description	Priority	Feasibility	Reference
1	To be able to retrieve research products based on given keywords.	H	2	link#1
2	To be able to use impact indicators to sort/rank search results.	H	2	link#2
3	To be able to use impact indicators (i.e., popularity or influence) to filter search results.	M	3	link#3
4	To be able to navigate through the results using pre-defined classes based on their impact.	M	3	link#4

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements		
#	Short description	Priority
1	The impact indicators should be calculated for each research product of the RDGraph that can get citations (e.g., publications). Citations are expected to be part of the RDGraph metadata.	H
2	The search functionality (API) should be able to incorporate impact indicators (e.g., citation counts, popularity, influence) in the ranking/sorting mechanism of the results.	H
3	The search functionality (API) should be able to allow filtering based on predefined classes of impact indicators, i.e., those being in the top-k.	M

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

Non-functional requirements		
#	Short description	Priority
1	The search API should be self-explainable with the appropriate documentation so that to help users on how to use it.	M

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

5.3.2 Specifications

Architectural design

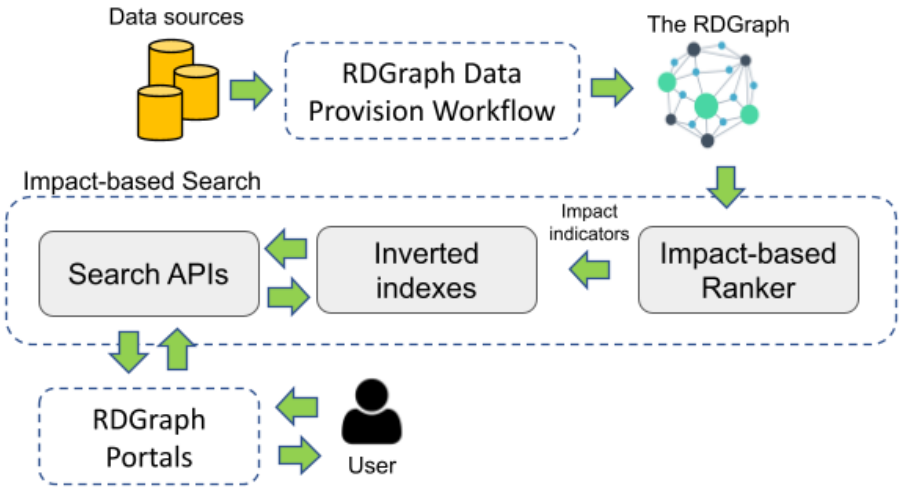

Figure 5

Functional specifications		
#	Short description	Priority

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications		
1	Implement an Impact-based Ranker; a component with scalable algorithms to calculate impact indicators using graph analysis techniques on the RDGraph.	H
2	Define "impact classes" based on each impact indicator so that to facilitate users using them in filtering search results.	M
3	Index the calculated impact indicators using appropriate Inverted Indexes in order to enable their use while performing keyword searches.	H
4	Extend the Search API by adding the functionality to incorporate impact indicators in the ranking/sorting mechanism.	H
5	Extend the Search API to support filtering by the predefined impact classes for each impact indicator.	M

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

Service specifications		
#	Short description	Priority
1	The end users (i.e., researchers) will be able to discover research products exploiting the impact-based search functionality via the RDGraph portals.	H
2	The impact scores for each indicators will also be available through the RDGrpah public dumps.	M

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

Operational specifications		
#	Short description	Priority
1	Calculating impact indicators would require a computational cluster, so that the process would not impede the RDGraph data provision workflow.	H

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

Integration with EOSC Core components		
#	Short description	Priority
1	The Impact-based Ranker will perform graph analysis techniques on the RDGraph to calculate impact indicators.	H

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

5.3.3 External references

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

External references		
#	Short description	Reference
1	T. Vergoulis, S. Chatzopoulos, I. Kanellos, P. Deligiannis, C. Tryfonopoulos, T. Dalamagas: BIP! Finder: Facilitating scientific literature search by exploiting impact-based ranking. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM), Beijing, China, November 2019	doi:10.1145/3357384.3357850
2	T. Vergoulis, I. Kanellos, C. Atzori, A. Mannocci, S. Chatzopoulos, S. La Bruzzo, N. Manola, P. Manghi: BIP! DB: A Dataset of Impact Measures for Scientific Publications. Companion Proceedings of the Web Conference 2021	doi:10.1145/3442442.3451369
3	I. Kanellos, T. Vergoulis, D. Sacharidis, T. Dalamagas, Y. Vassiliou: Impact-Based Ranking of Scientific Publications: A Survey and Experimental Evaluation. IEEE Transactions on Knowledge and Data Engineering (TKDE) 2019	doi:10.1109/TKDE.2019.2941206
4	I. Kanellos, T. Vergoulis, D. Sacharidis, T. Dalamagas, Y. Vassiliou: Ranking papers by their short-term scientific impact. International Conference on Data Engineering (ICDE) 2021	doi:10.1109/ICDE51399.2021.00190
5	S. Chatzopoulos, T. Vergoulis, I. Kanellos, T. Dalamagas, C. Tryfonopoulos: ArtSim: Improved estimation of current impact for recent articles. AimInScience @ TPDL 2020	doi:doi.org/10.1007/978-3-030-55814-7_27
6	S. Chatzopoulos, T. Vergoulis, I. Kanellos, T. Dalamagas, C. Tryfonopoulos: Further Improvements on Estimating the Popularity of Recently Published Papers. Quantitative Science Studies (QSS) 2021	doi:10.1162/qss_a_00165

5.4 RDGraph - Community recommendation profiles

Component	WP3 T.3.3, Community recommendation profiles
Category	RDGRAPH
Contact person	Stavroula Eleftheraki
Contributors	Stavroula Eleftheraki
Version	
Data	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Overview
Communities or community profiles connect users and research products (like publications, software, etc.) on a certain topic. Each community is associated with an RDGRAPH subgraph, the so-called community graph. Our goal is to create recommendation algorithms for these communities so that their users (e.g., mathematicians interested in operator theory publications) can recover content that is related to them and discover other community users with similar preferences. We distinguish three types of systems that may help with this: user-to-item recommenders, where based on the user's history and its associated community, one may recommend a list of works in which the user may be interested; user-to-user recommenders, where the user may discover potential collaborations with other members of the community; and item-to-item recommenders, where given that the current user is interested in a work A, other works (similar to A) are recommended. Moreover, we would like these recommendations to be given in a fair manner in terms of statistical parity. Publications by female authors, for instance, should be recommended as often as those by male authors. All these factors will be taken into account as we complete this task.

Objectives	
#	Short description
1	Develop (a) novel and fair recommendation algorithm(s) for community profiles.

Out of Scope	
#	Short description
1	We will not consider the "cold start" problem, in which a new research product enters the system. The cold start problem in terms of new research products is currently out of scope.
2	We will not adequately explain to users why a research product or user profile was suggested.

5.4.1 Requirements

User stories		
#	Description of the user story	Reference
1	We provide the following scenario for user-to-item recommendations: Consider a user in the field of climate change who works with climate models that characterize the interactions between energy and matter in different regions of the atmosphere. The user may want to find more works that are like the ones she or he has already interacted with.	TBD
2	We provide the following scenario for user-to-user recommendations: A mathematician working in the field of quantum theory who desires to know which other mathematicians she or he is related to. The recommender system should suggest other users with similar research interests based on the community to which this mathematician belongs.	TBD

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
3	We provide the following scenario for item-to-item recommendations: A user in the climate change community who is interested in a specific research product, such as an oceanographic data set, wishes to locate additional research products that are related to this one.	TBD

User requirements				
#	Short description	Priority	Feasibility	Reference
1	To be capable of receiving recommendations based on the user id or the research product id.	H	3	link#1
2	To obtain an unbiased list of recommendations.	M	2	link#2

- 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	Each person with a unique identifier associated with a research product of the community graph (e.g., the author of a publication with an ORCID) should be considered a community user and permitted to receive recommendations.	H	TBD
2	Each new user who joins the community should have access to recommendations, regardless of whether his or her information already exists on the community graph.	M	TBD
3	The API for the recommendation algorithm(s) should permit filtering based on the relevance of the algorithm's suggestions, so that the user sees only the top N most relevant suggestions.	M	TBD

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

Non-functional requirements			
#	Short description	Priority	Reference
1	The recommender(s) API should be easy to use, with adequate documentation to help community members understand how to use it.	M	TBD

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

5.4.2 Specifications

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

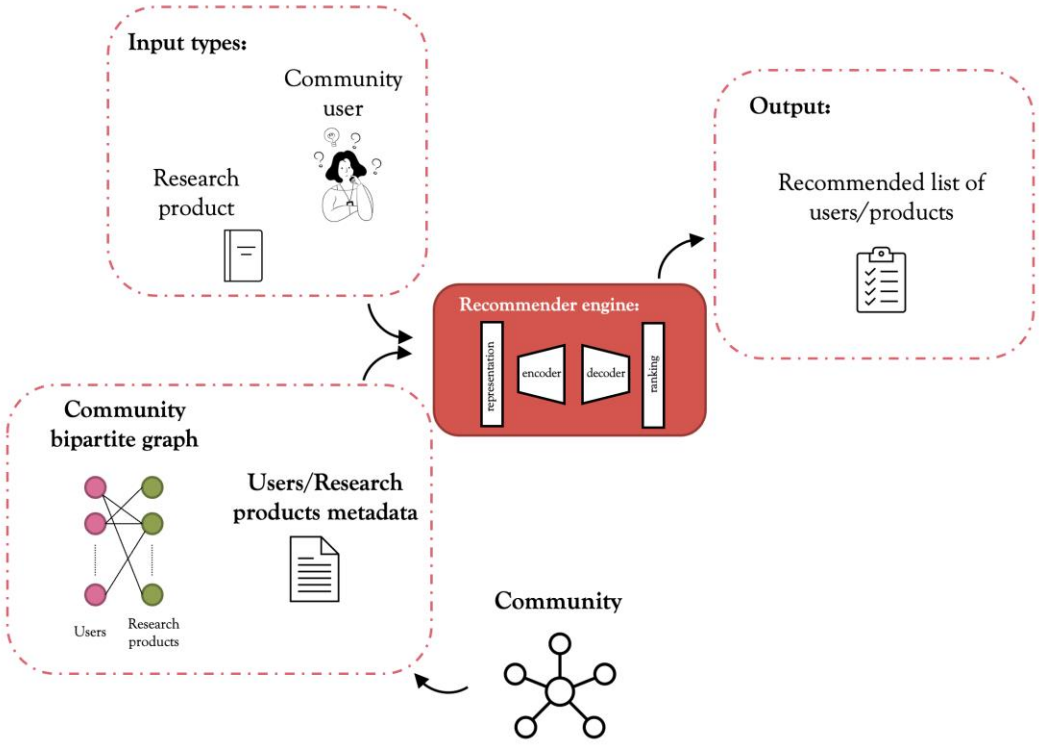

Figure 6

Functional specifications

#	Short description	Priority	Reference
1	Implement the recommender engine(s).	H	TBD
2	Extract the information needed as input to the recommender(s) from the community graph.	H	TBD
3	Extend the API of the recommender so that it can support relevance-based filtering (top-N recommendations).	M	TBD

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

Service specifications

#	Short description	Priority	Reference
1	Using the community-based recommender system(s), community users (i.e., researchers) will be able to find related research products or other community users similar to them in a fair manner.	H	TBD

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

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Operational specifications			
#	Short description	Priority	Reference
1	To generate recommendations for a user in community *K*, one must store and invert a dense matrix (a matrix of dimension $I_K \times I_K$, where I_K is the number of research products in community *K*).	H	TBD

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

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	Our service(s) will be made available through the various community portals.	L	TBD

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

5.4.3 External references

External references		
#	Short description	Reference
1	Pitoura, E., Stefanidis, K. and Koutrika, G., 2021. Fairness in rankings and recommendations: an overview. The VLDB Journal, pp.1-28.	doi:10.1007/s00778-021-00697-y

5.5 RDGraph - RAiD inference service

Component	RAiD inference service
Category	RDGRAPH
Contact person	Paolo Manghi
Contributors	Paolo Manghi Thanasis Vergoulis Serafeim Chatzopoulos
Version	
Data	

Overview
The RAiD inference service is responsible to identify possible entities in the RDGraph that could potentially constitute a RAiD. To this end, it considers structural properties of the graph as well as node connectivity and their respective metadata. Being connected to the EOSC RAiD service/registry, it facilitates EOSC users to register new RAiDs, as well as, track and manage their related activities.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Objectives	
#	Short description
1	Support automatic identification of entities in the RDGraph that can constitute a RAiD based on graph structure and node metadata.
2	Connect with the (EOSC) RAiD service/registry to suggest potential RAiDs in the RDGraph.

Out of Scope	
#	Short description
1	We will not address explainability issues; for example, why specific entities are suggested to be assigned to particular RAiD.

5.5.1 Requirements

User stories	
#	Description of the user story
1	As an EOSC user, I would like to get suggestions on possible entities (i.e., graph nodes) that could constitute a RAiD.
2	As a project manager, I would like to be able to get suggestions for possible research products that can extend existing RAiDs related to my project.

User requirements				
#	Short description	Priority	Feasibility	Reference
1	To be able to get suggestions for entities that can constitute a RAiD.	H	3	link
2	To be able to get possible extensions of existing RAiDs in the RDGraph.	H	4	link

- 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
1	This component should be able to identify sets of entities, based on their metadata or the graph's structure, that together can constitute a RAiD on the RDGraph.	H

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements		
2	The EOSC RAiD service/registry should be able to communicate with the RAiD inference service to retrieve the identified RAiDs.	H

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

Non-functional requirements		
#	Short description	Priority
1	This component should provide an API that is self-explainable and well-documented to facilitate its use from the EOSC RAiD service/registry or any other third-party user.	M

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

5.5.2 Specifications

Architectural design

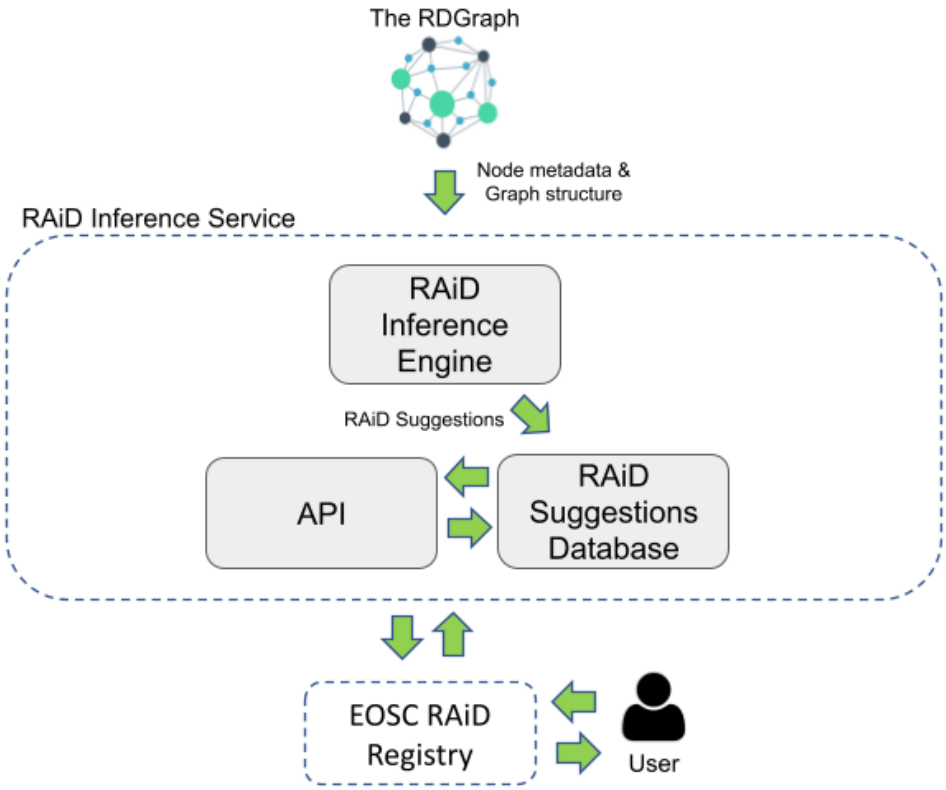

Figure 7

Functional specifications

#	Short description	Priority

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications		
1	Define a set of properties, either node attributes or structural ones, for each node/entity that can guide the identification of RAiDs in the RDGraph.	H
2	Implement appropriate inference algorithms to identify potential sets of entities that can constitute a RAiD.	H
3	Store the results (i.e., RAiD suggestions) of the RAiD inference engine in an appropriate database system so that they can be easily accessible upon request.	M
4	Create an API that would list all identified RAiDs in the graph.	H

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

Service specifications		
#	Short description	Priority
1	The end users (i.e., EOSC RAiD service users) will be able to get suggestions of sets of entities that can constitute a RAiD.	H

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

Operational specifications		
#	Short description	Priority
1	The process of inferring RAiDs in a large network, like the RDGraph, is a rather demanding and would require access to a computational cluster to be performed in a reasonable time.	H

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

Integration with EOSC Core components		
#	Short description	Priority
1	The RAiD inference service should communicate with the EOSC RAiD service/registry to provide the identified RAiDs as possible suggestions to the end user.	H

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

5.5.3 External references

External references		
#	Short description	Reference
1	S. Chatzopoulos, T. Vergoulis, P. Deligiannis, D. Skoutas, T. Dalamagas, C. Tryfonopoulos: "SciNeM: A Scalable Data Science Tool for	doi:10.5441/002/edbt.2021.76

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

External references		
	Heterogeneous Network Mining", In Proc. of the 24th International Conference on Extending Database Technology (EDBT) 2021	

5.6 RDGraph - RDGraph portals

Component	RDGraph portals
Category	CAT, RDGRAPH, PIDGRAPH, MSCR, DTR, RAID, PIDMR, RSAC, SWHM
Contact person	Katerina Iatropoulou
Contributors	Katerina Iatropoulou Konstantina Galouni
Version	
Data	

Overview
The RDGraph portal enables discovery, monitoring and access to the research products and services of the RDGraph. It will be the place where the functionalities of the Natural Based Search, Impact Based Search, Community Recommendations and RAiD management tasks will be directly available to the users via the provided user interface.

Objectives	
#	Short description
1	Make the contents of RDGraph searchable and accessible to the users via the RDGraph Portal.
2	Make Natural Based Search available to the users.
3	Make Impact Based Search available to the users.
4	Make Community Recommendations available to the users.
5	Make RAiD management functionalities available to the users.

Out of Scope	
#	Short description

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Out of Scope	
1	We will not elaborate on how the underlying Index service will integrate or implement the impact factors, the sorting functionalities and the RAiDs. We take for granted that the underlying technology exists and we just need to enhance a Search API to pass the requests properly to the underlying Index.
…	

5.6.1 Requirements

User stories		
#	Description of the user story	Reference
1	A researcher visits the RDGraph portal, wants to do a keyword search the contents of the RDGraph. The researcher wants to filter the results.	
2	A researcher visits the RDGraph Portal to query using natural language.	
3	A researcher visits the RDGraph Portal, makes a keyword search, and wants to rank/sort and filter the results based on their impact.	
4	A researcher of a specific community who is interested in a specific research product that appears in the RDGraph Portal, wishes to view and access additional research products in the RDGraph that are related to the one of interest.	
5	A user visits the RDGraph Portal and wants to view the existing RAiDs in the RDGraph.	
6	A project manager visits the RDGraph Portal and wants to be able to get suggestions for possible research products that can extend existing RAiDs related to their project.	

User requirements				
#	Short description	Priority	Feasibility	Reference
1	To be able to insert keywords and discover related content in the RDGraph.	H	5	link
2	To be able to filter the results of a keyword search by values of specific metadata fields.	H	5	link
3	To be able to make queries in natural language.	H	4	link
4	To be able to sort the results of a keyword search by their impact.	H	2	link
5	To be able to filter the results of a keyword search by their impact.	M	3	link
6	To be able to view recommended research products related to the authors of a specific research product.	H	3	link

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
7	To be able to view recommended research products related to the research products the user interacts with.	L	2	link
8	To be able to like research products and view recommended research products related to them.	M	2	link
9	To be able to view the available RAiDs of the RDGraph.	H	3	link
10	To be able to get suggestions of the possible extensions of existing RAiDs in the RDGraph.	M	3	link

- 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	The RDGraph Portal should provide search forms for keyword search.	H	
2	The RDGraph Portal should provide filter options.	H	
3	The RDGraph Portal should provide search forms for natural language.	H	
4	The RDGraph Portal should provide sorting options for the search results by their impact.	H	
5	The RDGraph Portal should provide filter options for the search results by their impact.	M	
6	The RDGraph Portal should display the recommended research products.	H	
7	The RDGraph Portal should provide a button for the "like" functionality of the research products.	M	
8	The RDGraph Portal should display the available RAiDs of the RDGraph.	H	
9	The RDGraph Portal should provide display possible extensions for the existing RAiDs.	M	

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

Non-functional requirements			
#	Short description	Priority	Reference
1	RDGraph portals should be user friendly, offering smooth user experience.	H	<JIRA issue number> or <URL to external reference>
…			

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

5.6.2 Specifications

Architectural design

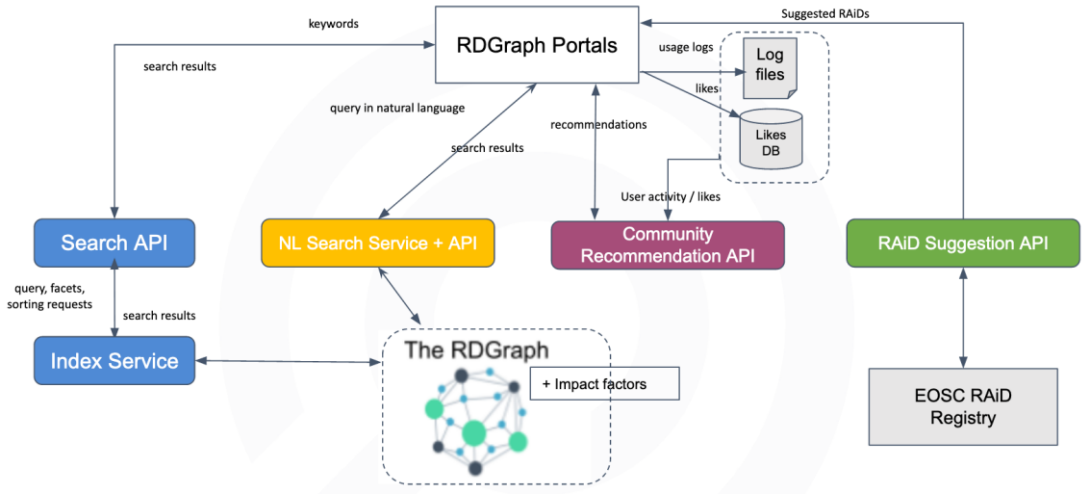

Figure 8

Functional specifications

#	Short description	Priority	Reference
1	Implement search forms for users to enter their keywords.	H	
2	Implement checkbox or radio button inputs for filter options.	H	
3	Implement search forms for users to enter queries in natural language.	H	
4	Implement select options for users to sort the search results by their impact.	H	
5	Implement checkbox or radio button input for impact-based filter options.	M	
6	Implement the parsing of the research product recommendations and implement a component that properly displays them.	H	
7	Implement a service to save and get the "liked" research products.	M	
8	Create and configure a database to save the "liked" research products.	M	
9	Implement a service to log the user interactions with the research products.	L	
11	Implement the parsing of the available RAiDs of the RDGraph and a component to properly display them.	H	
12	Implement the parsing of the possible extensions for the existing RAiDs.	M	

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Service specifications			
#	Short description	Priority	Reference
1	The RDGraph Portal should communicate with a search API that supports keyword and filter search and returns the results from the RDGraph.	H	
2	The RDGraph Portal should communicate with a natural language API that supports natural language search and returns the results from the RDGraph.	H	
3	The RDGraph Portal should communicate with a search API that supports impact-based sorting.	H	
4	The RDGraph Portal should communicate with a search API that supports impact-based filtering.	M	
5	The RDGraph Portal should communicate with a recommendation API to get the recommended research products.	H	
6	The RDGraph Portal should communicate with a recommendation API to send the user's interactions with the research products.	L	
7	The RDGraph Portal should communicate with an API to save and get the "liked" research products.	M	
8	The RDGraph Portal should communicate with an API to get the available RAiDs in RDGraph.	H	
9	The RDGraph Portal should communicate with an API to get the possible extensions for the existing RAiDs.	M	

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

Operational specifications			
#	Short description	Priority	Reference
1	The RDGraph Portal will require a machine that has direct access to the APIs of the integrated services.	H	<JIRA issue number> or <URL to external reference>
…			

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

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	The RDGraph Portals are built on top of the RDGraph. They provide interfaces where users can view, search, access and enhance the RDGraph contents.	H	<JIRA issue number> or

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Integration with EOSC Core components			
			<URL to external reference>
…			

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

5.6.3 External references

5.7 PID Graph access via APIs and data dumps

Component	PID Graph access via APIs and data dumps
Category	PIDGRAPH
Contact person	Sarala Wimalaratne

Overview
The component supports service workflows for the regular generation of data dumps related to community profiles for reuse by the EOSC community and related services. The harvesting API and data dumps will include the DOIs and metadata from the DataCite datafile for streamlined ingestion by EOSC and related services.

Objectives	
#	Short description
1	Support the harvesting of the PID Graph metadata. The API service and data dumps are made available for the community to ingest and reuse the metadata seamlessly. In addition, the component will focus on the interoperability framework for graph data exchange.

Out of scope	
#	Short description
1	The PID Graph data dumps will not include metadata of extended nodes (PIDs) and their associated metadata. The PID Graph data dumps include the primary PID and associated metadata with links to other persistent identifiers. Users of the service follow community best practice, where they can resolve metadata of relevant extended nodes (PIDs).

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

5.7.1 Requirements

User stories
As a harvester, I would like to be able to have access to a data dump with all the DOI, ORCID, and ROR metadata records in so that I can bootstrap my metadata harvesting.
As a harvester, I would like to be able to have access to a harvesting service that would allow me to get all the DOI, ORCID, and ROR metadata records in the PID Graph so that I can bootstrap my metadata harvesting.
As a Service Provider of a data search platform (B2Find administrator), I would like to be able to make harvest Datasets and publications in the PID Graph so that we can enrich our search platform (B2Find) with more resources.
As a Service Provider of a data publishing platform (B2Share administrator), I would like to be able to make harvest Datasets and publications in the PID Graph so that we can enrich our data publishing platform (B2Share) with more resources.
As a Service Provider (SURF - Netherlands) I would like to be able to harvest and query (from the PID Graph) scholarly outputs from National Institutions (Dutch Institutions), so that we can enrich our national Graph.
As a National Level Service provider (CSC - Finland) I would like to be able to harvest (from the PID Graph) scholarly outputs affiliated to National Institutions (CSC - Finland), so that we can enrich our own national systems.
As a National Level Service provider (HELIX - Greece) I would like to be able to harvest (from the PID Graph) scholarly outputs affiliated to National Institutions (HELIX - Greece), so that we can enrich our own national systems.
As a Service Provider of a data search platform (B2Find administrator) I would like to be able to harvest of Climate scholarly outputs available in the PID Graph, so to enrich the resources in my data search platform (B2Find).

User requirements

#	Short description	Priority	Feasibility	Reference
1	To be able to obtain a data dump up to last year	H	1	TBD
2	To be able to obtain a data dump up to last month	L	1	TBD
3	To be able to obtain a data dump in different formats	L	2	TBD
4	To be able to harvest PID Graph data (nodes and edges) using standard protocol	H	3	TBD
5	To be able to harvest resources affiliated to certain country	L	1	TBD
6	To be able to harvest resources of a certain field of science	L	1	TBD

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

- Feasibility: marking between 1 - 5, 5 is easy to implement and 1 very difficult

Functional requirements		
#	Short description	Priority
1	Given that user requests a DOI metadata **data dump** When the user sets the data dump request parameters Then a data dump with those parameters must be prepared And the data dump should be made ready for download	H
2	Given that user make a DOI metadata **harvesting** request When the user sets the harvesting request parameters Then the harvesting service will return records based on the set parameter	H
3	Given that user make a PID Graph Link **harvesting** request When the user sets the harvesting request parameters Then the harvesting service will return records based on the set parameter	H

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

Non-functional requirements		
#	Short description	Priority
1	The harvesting service should have response time equivalent to industry standards for harvesting (within reasonable effort)	M
2	The preparation of the data dump must be efficient and not impact the performance of the system.	H

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

5.7.2 Specifications

The specifications below address the *high-priority* user requirements that will be implemented as part of this project.

Architectural design

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

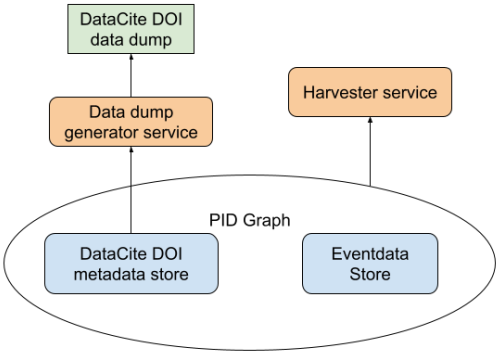

Figure 9

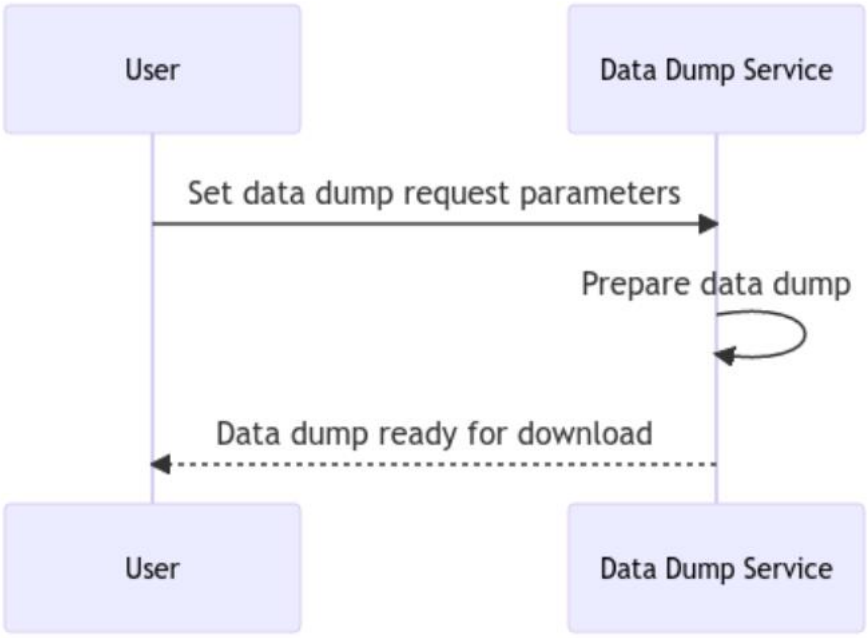

Figure 10

Functional specifications		
#	Short description	Priority
1	Annual data dump of DataCite DOI metadata	H
2	API to harvest DataCite DOI metadata	H

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

3	API to harvest PID graph links	H

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

Service specifications		
#	Short description	Priority
1	API to create annual data dump of DataCite DOI metadata in JSON format and store in an accessible location	H
2	Extending and improving existing API to harvest DataCite DOI metadata	H
3	API to harvest PID graph links	H

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

Operational specifications		
#	Short description	Priority
1	Annual data dump will be created end of every year	H
2	APIs will be available access content with some restrictions depending on the running costs	H

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

Integration with EOSC Core components		
#	Short description	Priority
1	Enable use of the DataCite data dump and APIs to feed PID graph data into RDGraph	H
…		

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

5.8 PID Graph - Data Usage Statistics

Component	Data Usage Statistics
Category	PIDGRAPH
Contact person	Sarala Wimalaratne

Overview
Integration of the data usage of DataCite DOIs with integrated statistics in PIDGraph.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Objectives	
1	In order for third parties and aggregators to ingest usage metrics into their services, the component aggregate usage data from DataCite DOIs (and integrated repositories) and exposes these via the APIs.

Out of scope	
#	Short description
1	Usage metrics are limited to the repositories that have integrated the usage track service by DataCite.
2	Usage data level metrics service tracker is scoped for scholarly outputs identified with a DOI

5.8.1 Requirements

User stories
As a provider of a data hosting platform, I would like to provide usage data level metrics to the PID graph about the resources identified with DOIs in my platform so that third parties can have access to them
As a provider of a metadata indexing service and a discovery portal, I would like to be able to obtain usage data level metrics from scholarly outputs indexed in my service so as to enrich my metadata index and provide such metrics in my portal

User requirements				
#	Short description	Priority	Feasibility	Reference
1	To be able to provide usage data level metrics to the PID Graph	L	3	TBD
2	To be able to request usage data level metrics from the PID Graph	L	3	TBD

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

Functional requirements		
#	Short description	Priority
1	Given that a data hosting platform is using DOIs to identify their data, When the hosting platform enables their platform with the DataCite usage tracker widget Then, the usage data tracked with the widget would be processed according to COUNTER And the usage data tracked would be accessible the PID Graph Interfaces	L
2	Given that usage data level metrics for a given DOI exist in the PID Graph	L

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

	When a discovery portal request a given DOI usage data level metrics any of the PID Graph Interfaces Then PID Graph Interface will return the usage level metrics of the DOI.	

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

Non-functional requirements		
#	Short description	Priority
1	Usage data level metrics will be available within ten days after the end of the calendar month.	L
2	The integration of the widget must not impact the performance of the data hosting platform.	H
3	The processed usage data must be accurate and reliable.	H
4	The accessibility of the usage data through the PID Graph Interfaces must be open and user-friendly.	M

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

5.8.2 Specifications

As part of the Make Data Count project, DataCite has been developing a usage statistics processing service. DataCite will be building upon this service to gather data usage statistics of DataCite repositories within EOSC partners.

Architectural design

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

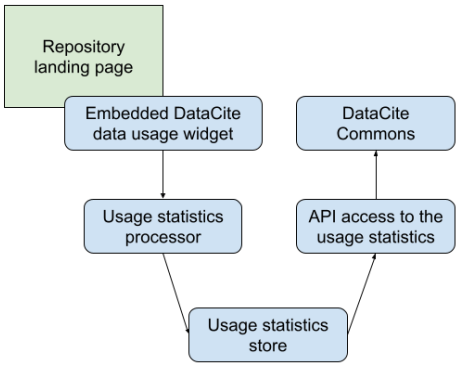

Figure 11

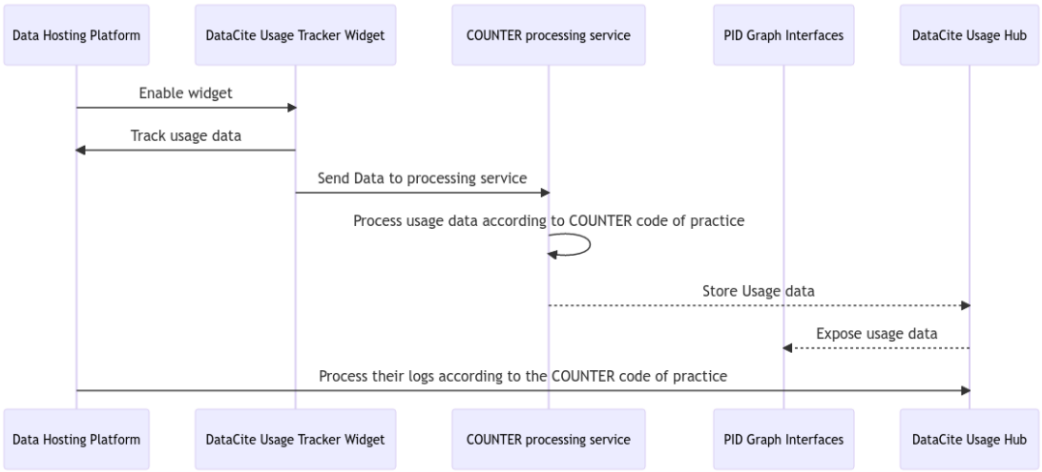

Figure 12

Functional specifications		
#	Short description	Priority
1	API access to the usage statistics	H

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

Service specifications		
#	Short description	Priority
1	Tracker widget to collect usage statistics	H
2	Usage statistics processor to collect usage statistics accounting to COUNTER code of practice	H

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

3	A service to store collected usage statistics	H
4	API access to aggregate usage statistics for a given DOI	H
…		

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

Operational specifications		
#	Short description	Priority
1	<Provide a short description of operational specifications>	[H|M|L]
…		

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

Integration with EOSC Core components		
#	Short description	Priority
1	Working with EOSC partners that have DataCite repositories to embed the tracker code in landing pages to collected usage statistics of DOIs	H
…		

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

5.9 PIDGraph - Link Claims

Component	PID link claims
Category	PIDGRAPH
Contact person	Sarala Wimalaratne

Overview
The component will ingest additional links between PID entities and expose these via PID Graph APIs. The service will support the aggregation of PID links.

Objectives	
#	Short description
1	In order for third parties and aggregators to ingest relational metadata into their, the component will ingest third-party links between PID entities and expose these via the PID Graph APIs.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Out of scope	
#	Short description
1	The third-party claims are limited to the controlled list values for relatedIdentier types (https://support.datacite.org/docs/schema-optional-properties-v43#section-12-a-related-identifier-type) and the relationTypes in the DataCite schema (https://support.datacite.org/docs/connecting-to-works#summary-of-all-relationtypes)

5.9.1 Requirements

User stories
As a Research infrastructure administrator (DKRZ), I would like to be able to make relationships of Climate scholarly outputs available in the PID Graph, so to increase the discoverability and re-use of the scholarly outputs from my Research infrastructure and the field of Climate.
As an SSH resources discovery provider (CLARIN administrator), I would like to make relationships of SSH scholarly outputs available in the PID Graph to increase the discoverability of the scholarly outputs from our discovery platform.
As a Mathematics service provider (zbMATH administrator), I would like to be able to make relationships between zbMATH author identifiers and ORCIDs available in the PID Graph so that PID Graph queries can be made in the PID graph.
As a Mathematics service provider (zbMATH administrator), I would like to be able to make relationships between zbMATH publication/software identifiers and zbMATH author identifiers available in the PID Graph so that PID Graph queries can be made in the PID graph.
As a Mathematics service provider (zbMATH administrator), I would like to be able to make relationships between zbMATH publication/software identifiers and DOIs available in the PID Graph so that PID Graph queries can be made in the PID graph.

User requirements				
#	Short description	Priority	Feasibility	Reference
1	To be able to provide same-as relationships assertions between document domain Identifiers and DOIs to the PID Graph	H	3	TBD
2	To be able to provide same-as relationships assertions between author domain Identifiers and ORCIDs to the PID Graph	M	3	TBD
3	To be able to provide part-of relationships assertions between document domain Identifiers and DOIs to the PID Graph	M	3	TBD

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

- Feasibility: marking between 1 - 5, 5 is easy to implement and 1 very difficult

Functional requirements		
#	Short description	Priority
1	Given that a research infrastructure provider has assertions between DOIs and domain Identifiers When the agent service is triggered Then the agent will delta-harvest the provider API service And the agent will store the assertions in PID Graph storage And the PID Graph interfaces will have access to the assertions	H
2	Given that a research infrastructure provider has a assertions between DOIs and domain Identifiers When the provider makes DataCite aware that they have assertions And the provider enables a standard protocol for query, pagination, and delta harvesting to DataCite Then Datacite will make preparation to add a new agent for the provider	H
3	Given that assertions between DOIs and domain Identifiers exist in the PID Graph When a user query/request/harvester the PID graph interfaces by the given domain identifier Then the PID Graph interfaces will present the relationships to DOIs, ORCIDs and RORs	M
4	Given that assertions between DOIs and domain Identifiers exist in the PID Graph When a user query/request a given domain identifier in the PID Graph frontend interfaces Then the PID Graph frontend interfaces will present domain identifier and its relationships to DOIs, ORCIDs and RORs	M

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

Non-functional requirements		
#	Short description	Priority
1	The agent service should have response time equivalent to industry standards for harvesting (within reasonable effort)	M
2	The delta-harvesting of assertions must be efficient and not cause excessive load on the provider API service.	H
3	The agent service will delta-harvest assertions from the provider API service using the provided API endpoint and standard harvesting protocols.	H
4	The storage of assertions in PID Graph storage must be secure and reliable.	H

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

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

5.9.2 Specifications

Alongside the FC4E project, DataCite will be developing additional mechanisms to ingest citation metadata from third party sources. Specifically, DataCite will be developing workflows and agents to add links to the Citation Corpus. Within FAIRCORE4EOSC, DataCite will implement an Event agent to add PID links from Scholix hubs (and similar hubs) to the PID Graph. DataCite will detail and implement the data access layer as part of these related efforts.

Architectural design

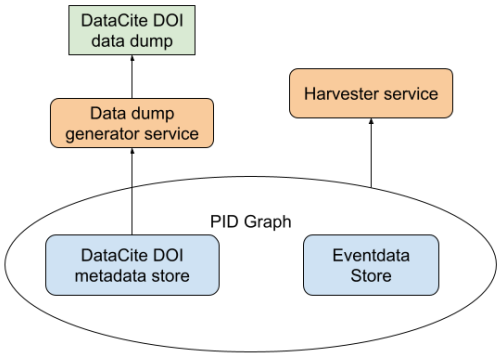

Figure 13

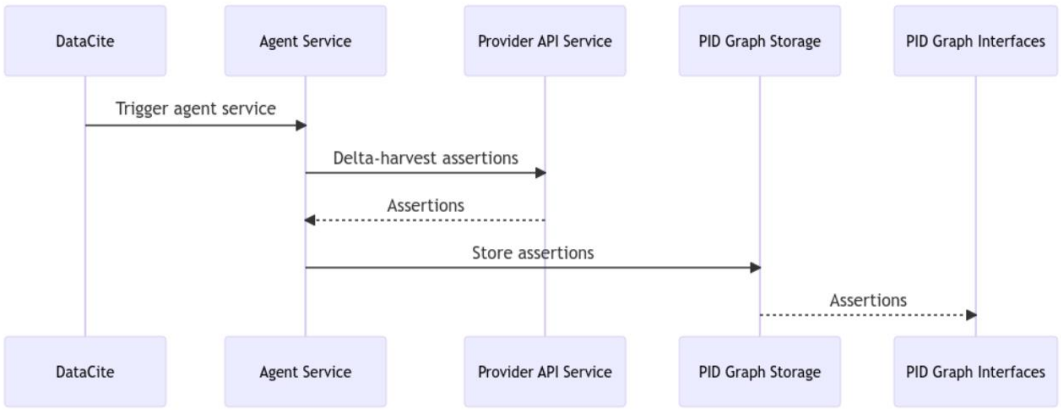

Figure 14

Functional specifications

#	Short description	Priority
1	Event agent service to query Scholix end points	H

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

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

Service specifications		
#	Short description	Priority
1	Event agent API to query EOSC partner Scholix hubs periodically to expand the Citation Corpus. This work will involve guiding the EOSC partners to expose their data in Scholix hub format so DataCite can harvest the connections.	H

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

Operational specifications		
#	Short description	Priority
1	Event agent API will query registered Scholix hubs monthly to update the Citation corpus	H
…		

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

Integration with EOSC Core components		
#	Short description	Priority
1	Working with EOSC partners to expose their links as a Scholix hub and integrating with DataCite eventdata agent to harvest links from EOSC partners	H

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

6 WP4 – EOSC registries for research data and metadata interoperability and research activity tracking

WP4 will build provide a number of components and services that are considered essential for future development of a federated interoperable EOSC and will be added to the current EOSC core ref.

EOSC core components are meant deliver the glue between the heterogeneous data and services as provided by a highly diverse landscape of thematic and national research infrastructures. The WP4 components and services will be initially used in a number of community case-studies that will be implemented in [WP7], which are targeted towards integrating the WP4 components and services in research communities workflows, and also in a number of demonstrators. Those demonstrators are primarily meant to demonstrate the functioning of the individual components and services as being fit-for-purpose. The number of demonstrators is open, meaning that next to the demonstrators described in the project proposal a additional demonstrators can (and have) been taken-up for instance to enable collaboration with external projects and initiatives, provided the project resources are sufficient or adequate resources can be provided by our collaboration partner.

The to be build WP4 components and services are:

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

- The Metadata Schema and Crosswalk Registry (MSCR), which is a new to be build service but that builds on existing landscape and requirements analysis with respect to semantic artefacts and semantic (meta) data interoperability from [FAIRSFAIR] and [SEMAF]. The MSCR will allow researchers and projects to register metadata schema and create and maintain metadata crosswalks between these, using smart tools with graphical interfaces to create crosswalks and semantic mappings. Semantic mappings and interoperability is an important aspect of a federated interoperable infrastructure, and currently there was not yet a general service provided within the EOSC. The responsible partner

- The Data Type Registry (DTR), is a service enabling researchers and projects to create, publish and share data type definitions of all sorts. For instance to standardise the structure of types of data or make them machine actionable. The DTR was developed in the past in the context of the RDA DTR Working group and it plays an important role with PID Kernel Types and the Fair Digital Objects (FDO) initiative. Standardising and sharing types is also necessary for a federated interoperable infrastructure, together with the semantic functionalities offered from the MSCR. In WP4 the DTR will be adapted for EOSC requirements and integrated in the EOSC platform, and also a number of new types will be added.

- The EOSC Research Activity Identifier service (RAiD) which will be an EOSC customised and dedicated deployment of the original (Australian ARDC operated) RAiD platform. The service will mint PIDs for research projects and allow managing of project related activities and content links. The EOSC RAiD service will be integrated in the EOSC Marketplace and RDGRaph and development work is aligned with the RAiD developments at ARDC. The EOSC RAiD service will be integrated with the ARDC maintained RAiD register as well.

The WP4 structure is:

- task 4.1 led by SURF working on consistent requirement analysis for the WP4 components and services

- task 4.2 led by CSC is responsible for implementing the MSCR

- task 4.3 led by GWDG is responsible for implementing the DTR

- task 4.4 led by SURF is responsible for implementing the RAiD service

- task 4..5 led by CLARIN ERIC is responsible for setting up a number of demonstrators for the MSCR and DTR. These include those demonstrators described in the initial proposal and the additional ones that are part of collaborations with external partners and are acquired during the project by WP4 outreach activities.

6.1 T4.2 MSCR - Specifications

Component	Metadata Schema and Crosswalk Registry
Category	MSCR
Contact person	Joonas Kesäniemi (CSC)
Email address	Joonas Kesäniemi

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Contributors	Tommi Suominen(CSC), Joonas Kesäniemi(CSC), Daan Broeder(CLARIN)
Version	1.0.0
Data	

Overview
MSCR allows registered users and communities to create, register and version schemas and crosswalks with PIDs. The published content can be searched, browsed and downloaded without restrictions. MSCR provides an API to facilitate the transformation of data from one schema to another via registered crosswalks.

Objectives	
#	Short description
1	Registration of metadata schemas/crosswalks with different levels of technical details. This can be done using the UI or through an API.
2	Minting of resolvable PIDs for registered schemas, mappings and crosswalks.
3	Facilitate conversion of data from one metadata schema to another. • Provide an UI for creating crosswalks • Downloadable crosswalks that can be used to implement data conversion
4	Provide search, browse and view functionality for the registered schemas and crosswalks and associated content through the UI and API.

Out of Scope	
#	Short description
1	Resolving minted PIDs. This should be handled by an external PID resolver service.
…	

6.1.1 Requirements

The user stories and user requirements pertaining to MSCR are collected from T4.5 & T7.1

Functional requirements			
#	Short description	Priority	Reference
1	System must support registration of both locally and externally management schemas and crosswalks.	H	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
2	System must support registration of metadata schemas with one or more of the following descriptors • Documentation (e.g. PDF) • Structural description (e.g. JSON schema) • Semantic description (e.g. OWL ontology)	H	
3	System must support user groups	H	
4	System must support the following user roles: • System admin • Group admin • User	H	
5	System must support user registration and login.	H	
6	Groups can only be created by system admins	H	
7	Group admins can only be added by system admins	H	
8	If a user is associated with multiple groups, the user must be able to select one active group at a time.	L	
9	Registered content can be owned by a group or individual user.	H	
10	Ownership of a content owned by a user can be transferred to a group.	L	
11	The system must support existing schema description formats.	H	
12	The system must support existing crosswalk description formats.	H	
13	System must support registration of hosted and referenced schemas and crosswalks.	H	
14	Only logged in users can register schemas and crosswalks.	H	
15	The system must support versioning of registered content.	H	
16	The system provides access to all the uploaded original versions of schemas and crosswalks.	M	
17	The system must support versioning of hosted schemas and crosswalks.	H	
18	The system provides provenance information related to versioning of registered schemas and crosswalks.	H	
19	The system must support flexible metadata for schemas and crosswalks	H	
20	The system generates PIDs for hosted schemas, crosswalks and mappings.	H	
21	The system facilitates the resolvability of the PIDs it has created	H	
22	The system provides a configurable metadata record for every PID.	M	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
23	Resolvable PID URLs must support content-negotiation for both humans and machines.	L	
24	The system provides a way to visualize registered schemas.	H	
25	The system provides a way to visualize registered crosswalks.	H	
26	The system must provide a way to search schemas/crosswalks based on the their metadata.	H	
27	A user must be able to retrieve all crosswalks for a given registered schema.	L	
28	A user must be able to browse version history of a registered schema/crosswalk.	M	
29	A user must be able to determine based on a PID whether or not there is a newer version of the schema/crosswalk available.	M	
30	The system provides a way for registered users to discuss about registered content.	M	
31	The system provides a way to define a status for each version of the registered schema and crosswalk.	H	
32	Logged in user can register for change notifications for registered schemas and crosswalks.	L	
33	The system provides an UI for creating schema description from scratch.	L	
34	Registered schema version can contain multiple different representations.	M	
35	The system provides a way to integrate data type information from the DTR to hosted schemas.	H	
35	Only registered users can create crosswalks.	H	
36	The system must support creation of personal schemas that are saved but not registered.	M	
37	The system provides automatically generated mapping suggestions.	L	
38	System must support contextual mappings.	L	
39	Only mappings from registered crosswalks are searchable/browsable and re-usable.	H	
40	The system must support creation of personal crosswalks that are saved but not registered.	M	
41	The system provides a way to turn a personal schema/crosswalk into a registered one.	L	
42	The system must differentiate between structural schema and vocabulary schema.	H	
43	The system provides a way to compare schema versions.	M	
44	The system provides a way to compare crosswalk versions.	M	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
45	The system provides a documentation of the crosswalk for download,	H	
46	The system provides an actionable version of the crosswalk for download	M	
47	All registered content must be downloadable without any restrictions.	H	
48	The system provides an API token read/write access to the content.	M	
49	The system provides up-to-date API documentation	H	
50	The system must provide the user the possibility to delete his or her account in a way that anonymizes any content associated to the account.	L	
51	The system must provide suitable endpoints for remote monitoring.	L	
52	The system must provide suitable endpoint for remote statistics gathering.	L	
53	Export hosted registered schemas as JSON schema.	L	B2Share
54	The system must provide API for search and CRUD operations for all major domain object types.	H	
55	The system provides UI for schema editing	M	
56	The system provides UI for creating and editing crosswalk	H	

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

Non-functional requirements			
#	Short description	Priority	Reference
1	All service related source code must be openly available.	H	
2	Licensing of the re-used open source software must be respected.	H	
3	The system must provide a unified user experience (as opposed to multiple separate UI).	H	
4	The system should be run a containerized environment.	L	
5	The system must use technologies familiar to the development team (as opposed to using resources to learn new tools).	H	
6	The system must be GDPR compliant.	H	
7	The system must use RDF based internal data model for schema and crosswalk data.	H	

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

6.1.2 Specifications

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

Interactive and evolving version of the architectural design documentation can be found in the IcePanel and can be browser from https://s.icepanel.io/gNLPEU9bqSjAq0/V33k.

Level 1 - System level

MSCR works with the EOSC core services related authentication, monitoring and statistics gathering. PID resolver integration is needed for minting new PIDs and managing the resolvable targets and possible metadata information for schemas and crosswalks PIDs. The MCSR is envisioned to implement both email and webhook based notifications. A repository refers to any external service that is used to maintain the "master" version of the schema or crosswalk information.

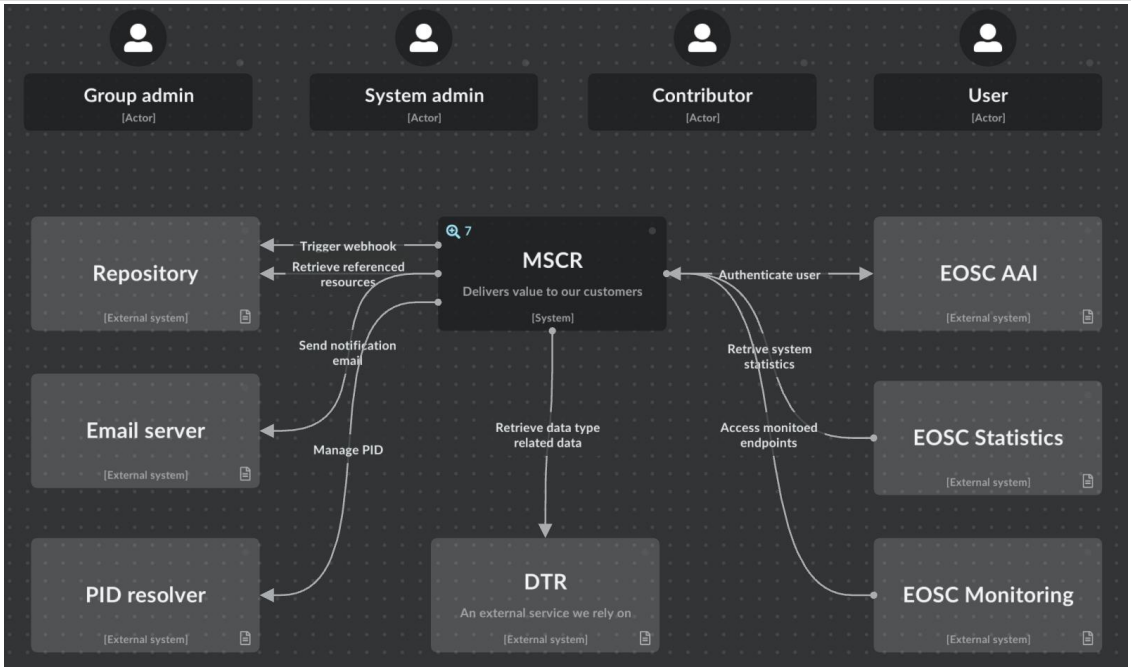

Figure 15

Level 2 - Application level

MSCR consists of separate frontend and backend/API applications. All interaction with the content happens through the API which provides a set of openly available endpoints for public information and restricted endpoints for write operations. Three different types of databases for primary data reflect the range of data that is handled by the MSCR. Triple store is the primary store for all internal schema and crosswalk related information. Relational database is used to store other important structural information such as user groups, permissions and comments. Finally, file system is used to store and serve the original versions of the hosted schemas and crosswalks.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

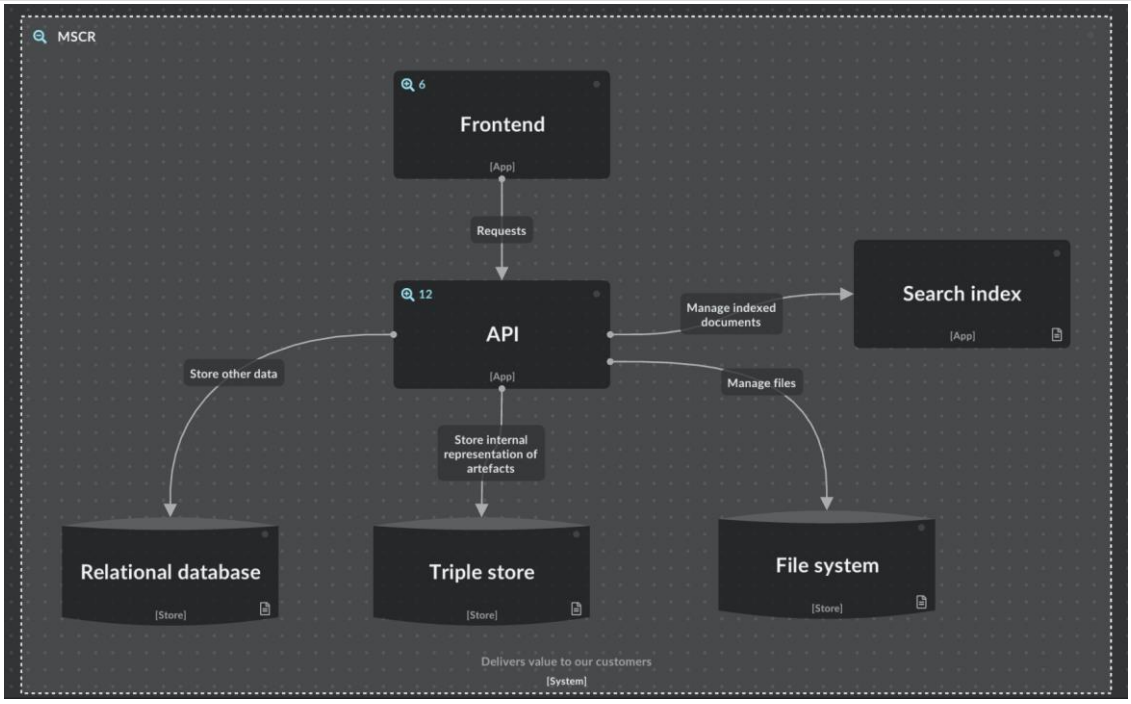

Figure 16

Level 3 - Component level

Diagrams at the component level are very preliminary at this stage of the development. As we hoping to re-use existing solutions, the components architecture will be heavily shaped by the structuring of the selected system.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

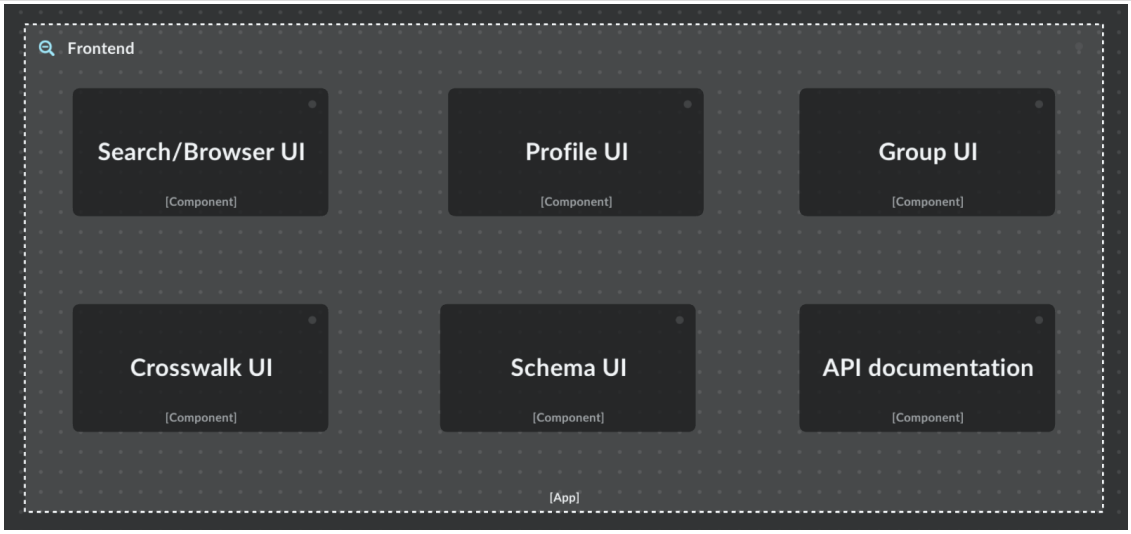

Figure 17

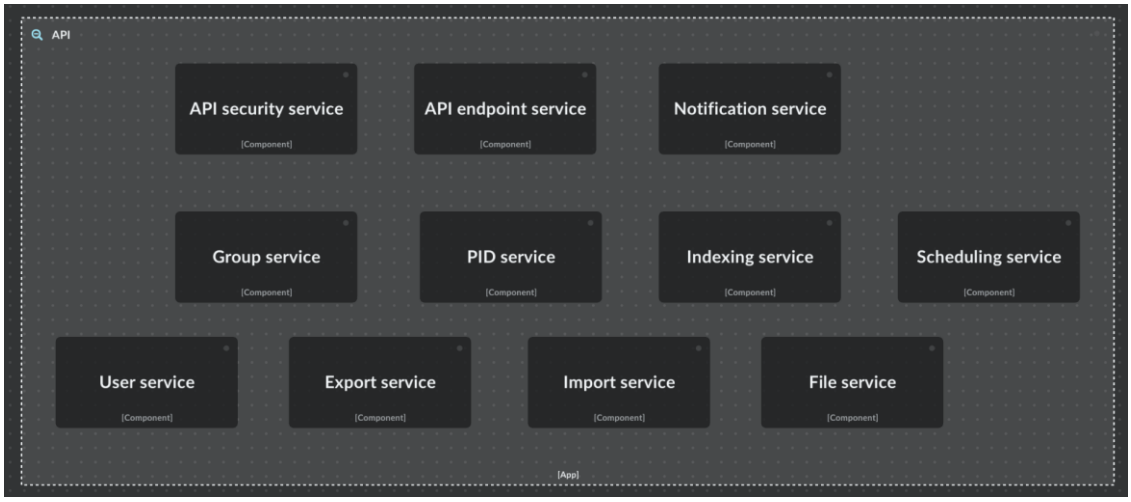

Figure 18

Functional specifications

#	Short description	Priority	Reference
1	Content registration Registered users can register schemas and crosswalks as either referenced of hosted content. In case of references registration the content is hosted and maintained in an external system where as with the hosted content the MSCR is the primary source of the content. Registered content is publicly available through the UI and the API.	H	
2	User groups	M	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	User groups are used to organize content, users and maintenance responsibility around established communities.Groups can only be created by system admins and they are used to distinguish and promote communitity driven and coordinated authorative content from individual contributions.		
3	User roles The supported user roles are system admin, group admin and registered user. System admin • Create and modify users and groups • Content related admin functions Group admin • Can add new group admins (if specified by the system admin) • Can add users to the group • Can remove users from the group • Can create API tokens for the group User • Can register schemas and crosswalks under personal account and to all the groups involved • Can create new versions of personal and groups' schemas and crosswalks. • Can modify metadata and other content of the personal and groups' schemas and crosswalks.	M	
4	Authentication and authorization Authentication is possible either against internal user registry or via EOSC AAI solution. All authorization information is handled locally and maintained by system and group admins.	M	
5	API token API tokens can be created for both users and groups. Group API tokens can be used as a sort of service accounts that can be configured for automated use of the MSCR API.	L	
6	Supported schema description formats MSCR supports a set of schema description format for schema regisration. Possible supported schema description formats include: CSV, CSV schema, DDL, XSD, JSON Schema, RDFS, OWL, SHACL. SKOS (vocabulary schemas). The implementation priorities will be determined based on requirements from demonstrators and case studies.	H	
7	Supported crosswalk description formats	H	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	The identified formats for registered crosswalk formats are: mapping tables, XSLT and SSSOM mapping sets.		
8	Versioning There is no need to keep track of all the changes related to hosted content. It is enough to record manually curated version information. Versions populate a version graph that can be utilized in the schema/crosswalk view to show previous/next versions.	M	
9	Content structure Schemas and crosswalk are described with MSCR's internal metadata schema (to be decided later). MSCR makes an internal distinctions between data model schemas and vocabulary schemas, which can be used to provide for example different UIs for different cases. Content structuring relates general description of the content (e.g. metadata for DataCite) to its versions (e.g. logical model for DataCite v4.3) and representations (XSD representing DataCite 4.3).	H	
10	Notifications Notifications are essential for keeping crosswalk maintainers and user using the best available versions. Registered users can register for notifications about new versions for any number of schemas and crosswalks. Notifications can be delivered via email or they can trigger a webhook. The latter is targeted towards automated processes.	L	
11	PID management All schemas and crosswalks (and individual mappings) in the MSCR is identified with a PID. PIDs for hosted content is managed by the MSCR and all referenced schema and crosswalk registration must provide an existing PIDs. The system facilitates the resolvability of the PIDs it has created and provides a possiblity to add configurable metadata record for PIDs.	H	
12	Commenting Basic commenting facilities are provided for registered users.	L	
13	Schema editing UI Schema UI consists of browsing and editing facilities for the registered schemas. The details of the of the schema UI are under development. In order to support crosswalks, it is necessary to work on the representation level of schemas. For interoperability perspective it would be beneficial to also include handling of implementation-agnostic logical level schemas. Schema UI provides visualization and editing for the supported schema formats.	H	
14	Crosswalk editing UI Crosswalk UI provides means for viewing and editing registered crosswalks. The details of the of the crosswalk UI are under development. The UI supports visualization and editing of crosswalks in their internal format only. There can be some supported crosswalk formats that can be registered and downloaded, but not edited/visualized.	H	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
15	API All the major capabilies related to adding, searching, editing and deleting MSCR's domain objects are available through the API. Some of the (read-only) endpoints are public. All modifications are done through restricted endpoints with authorization.	H	
16	Internal data model Internally all registered schemas and crosswalks are represented using their respective common RDF based data models. The internal data model uses SKOS, OWL and SHACL vocabularies with MSCR specific extensions.	H	
17	Operationalization of crosswalks Operationalization of crosswalks can be implemented on different levels with varied amount of support for integration. It can vary from providing instructions for humans how to implement the crosswalk (e.g. generated PDF) to some downloadable software artefact/service endpoint that provides the functionality. Details of suitable implementation or implementations are yet to be determined and are depended on e.g. capabilities and resources of the demonstrators and case studies.	L	

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

More detailed and up-to-date version of the functional specification can be found in T4.2.2 MSCR - Functional specifications.

Service specifications			
#	Short description	Priority	Reference
1	All registration and versioning related functionality can be used both through the UI and the API.	[H|M|L]	<JIRA issue number> or <URL to external reference>
…			

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

Operational specifications			
#	Short description	Priority	Reference
1	TBD	[H|M|L]	<JIRA issue number> or <URL to external reference>
…			

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	EOSC AAI can be used as an authentication provider for the MSCR.	H	
2	EOSC Monitoring is provided with endpoints that can be used to determine the status of the MSCR service in a reliable way.	M	
3	EOSC Statistics is provided with endpoints that can used to determine the state of the content of the MSCR.	L	

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

6.1.3 External references

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

6.2 T4.3 DTR - Specifications

The Data Type Registry allows the registration of types based on predefined type definitions. With a set of types, a Kernel Information Profile (see WP5 - T5.3) can be constructed. A type can be simple or complex, whereas a complex type depends on one or more simple or complex types and the depth of the hierarchy can be larger than 2.

Naming definition used in the below specifications:

- user: Any person making use of types
- admin: A person that can manage types on behalf of a steering group
- sysadmin: IT staff that is responsible of the software maintenance

Component	Data Type Registry
Category	DTR
Contact person	Sven Bingert (GWDG), Hans Lienhop (GWDG)
Email address	Sven Bingert Hans Lienhop
Contributors	GWDG, SURF, CLARIN, GRNET, AGH/AGH-UST
Version	
Data	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Overview
The Data Type Registry, or DTR, provides a centralized registry of data type definitions and instances thereof. Each entry in the DTR is provided with a persistent identifier. Entries in the DTR can be created and resolved. Created to fill the DTR with missing entries, resolved to validate entries in other metadata content.

Objectives	
#	Short description
1	Provide a registry of metadata definitions and instances
2	Provide a user interface to create new entries
3	Provide an API to resolve entries
4	Access external DTR's via the API
5	Demonstrate the added value via demonstrators

Out of Scope	
#	Short description
1	The DTR should not provide a registry for full metadata schemas. This service is provided by the MSCR.

6.2.1 Requirements

The user stories and user requirements pertaining to MSCR are collected from T4.5 & T7.1.

In addition, the following user stories and requirements are based on the DTR's natural function and/or generalisation or synthesis of those from T4.5 and T7.1.

User stories		
#	Description of the user story	Reference
1	As a <researcher> I want to <register types> because I want <to reference to it in my schema>	<URL to an external reference describing the user story in more detail>
2	As a <user> I want to <login> to <create types>	
3	As a <user> I want <registered types to be used in a PID> , so that <longterm availability can be assured>	
4	As an <admin> I want to <create an ACL>, so that <I can control access rights>	
5	As a <user> I want to <get the type description as XML>, so that <I can better include types in my software>	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
6	As a <sysadmin> I want to <be able to create a backup> so that <data loss can be prevented>	
7	As a <sysadmin> I want to <scale horizontally> so that <large amount of read requests can be served>	
8	As an <admin> I want to <assign licences to the type descriptions> so that <a reuse of the types is enabled>	
9	As a <user> I want to <reuse existing data type definitions> so that <I can reuse existing work>	
10	As a <user> I want <support for extended data types> to <further specify my intent>	
11	As a <user> I want <support for a taxonomy of the data types> so that <exploration is improved>	
12	As a <user> I want to <register enum lists for basic types> so that <I can create vocabularies>	
13	As a <user> I want to <be able to retrieve a validation schema for a type> so that <I can validate created type instances>	
14	When creating metadata crosswalks the type/value schemes of the metadata elements and attributes can be required for the actually conversion of metadata later-on. A user can specify the data-type/value schema of metadata schema's hosted in the MSCR. The MSCR uses the DTR API to provide the user with an interface to select a suitable data-type. The data-type reference is stored in the metadata schema representation and can be visualised when inspecting the metadata schema.	FC4E 4.5 demonstrator scripts part 4.5.3
15	DTR data managers seed (from IANA) and maintain an extended mime-type vocabulary (EDFR). Community experts can add to this vocabulary. Community software and experts can search and browse the vocabulary using relevant APIs for the benefit of authoritative data format typing of resources and capabilities of services.	
16	To achieve interoperability between different EOSC Core and B2 services, a centrally published maintained and versioned schema for the description of instruments should be used. Tool builders (and tool users) can make use of unique identifiers that are provided B2Inst provides for instruments. B2Inst uses internal described profiles. The DTR would provide registration and management of instrument profiles and validation. B2Inst should load the profile from the DTR and dynamically build a Form UI for users for them to enter the instrument details. Versioning of profiles should be possible.	

User requirements				
#	Short description	Priority	Feasibility	Reference

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
1	The system should provide access via a web interface	H		<JIRA issue number> or <URL to external reference>
2	The system should allow for entries to be created	H		
3	The system should allow for entries to be edited	H		
4	The system should allow for entries to be versioned	H		
5	The system should allow for authentication via password	M		
6	The system should allow for a way to set up an account	H		
7	The system should allow for searching entries via the interface and the API	H		
8	The system should allow for filtering results	H		
9	The DTR has a suitable schema (currently under discussion) for basic data-types and users can create new data-types (basic data-type schema instances)	H		
10	The DTR provides an API which allows other tools (ie MSCR) to browse / search the set of basic data-types	H		
11	The DTR should implement extended mime-type schema and the DTR should support community expert contributions and establish editorial rules	H		
12	The DTR should serve vocabulary of extended mime-types in SKOS format and support other APIs eg. SKOSMOS API (tbd) for benefit of client software	M		

- 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	The system should allow that via some provider, PID's are created and stored	H	
2	The system should allow for the ability to individually grant users access rights	H	
3	The system should allow format conversions	M	
4	The system should allow retrieval of types from external DTR's	H	
5	The system should allow creation only for authorized users	H	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
6	An entry is owned by one user	H	
7	The system provides provenance information related to versioning of registered data types	M	
8	The system should provide a way to graphically explore relations between data types	M	
9	All registered types should be downloadable	H	
10	The API should allow for authorized users to bulk upload types	M	
11	The system must allow for a user to delete his or her account	H	
12	The system must provide suitable endpoints for remote monitoring.	M	

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

Non-functional requirements			
#	Short description	Priority	Reference
1	Integrity; the data in the DTR is consistent and safe	H	
2	Reliability; the DTR minimalizes failures	H	
3	Scalability; that the DTR can answer a potentially high number of request per given time unit	H	
4	Usability; The DTR offers a multitude of endpoints for different users needs	M	

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

6.2.2 Specifications

Architectural design
From an architectural viewpoint, we propose the use of the Cordra software developed by CNRI. It already offers a UI to register digital objects, is equipped with a handle service and an authentication module and some API endpoints for outside communication. In addition to the Cordra software, we propose the implementation of a so called TypeAPI. This python based API will extend the given functionalities by offering the retrieval of validation schemas, type descriptions, a search over multiple DTR's and resolving results in human readable form. The TypeAPI will offer all those retrieval methods in multiple formats, e.g. JSON, HTML and XML. Following we provide a C4 Model document from level 1-3, with the code diagram for the TypeAPI being provided in the next stage of development.

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

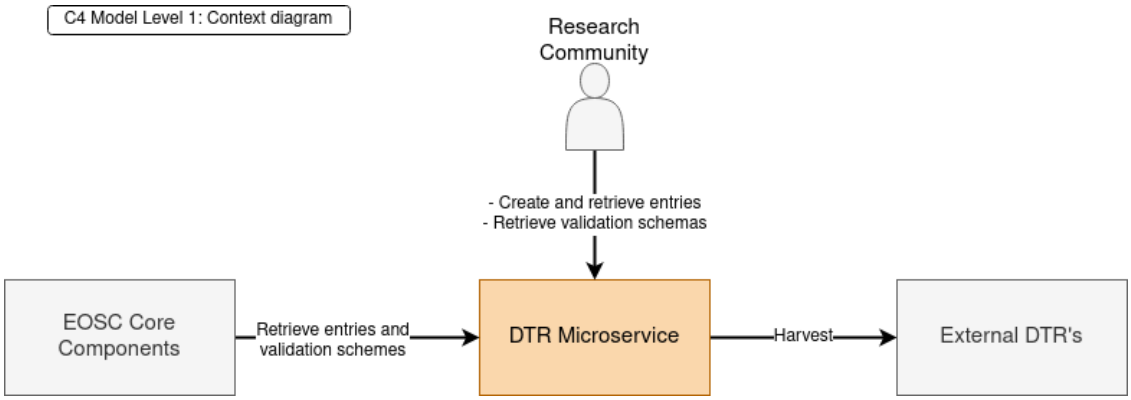

Figure 19

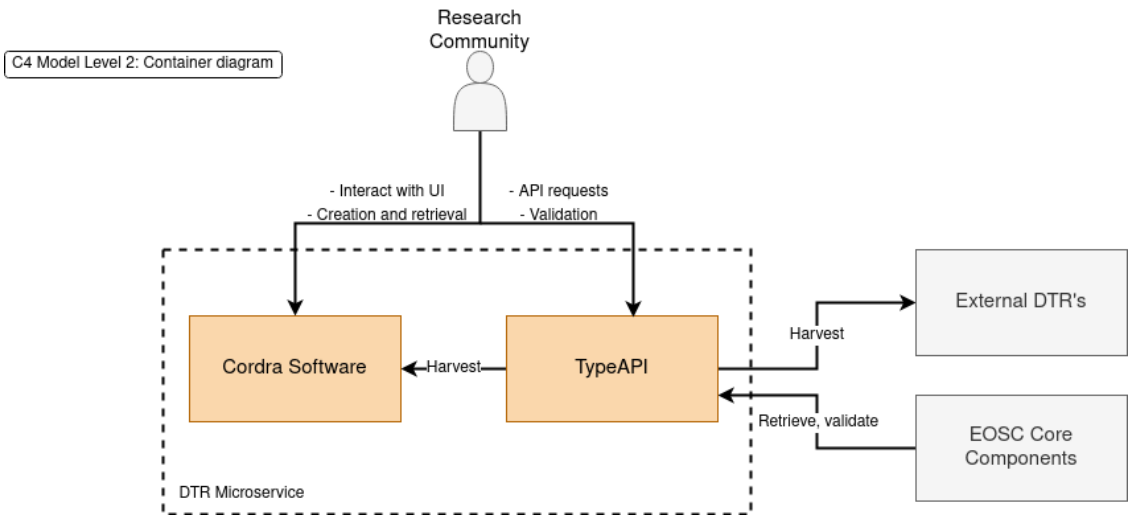

Figure 20

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

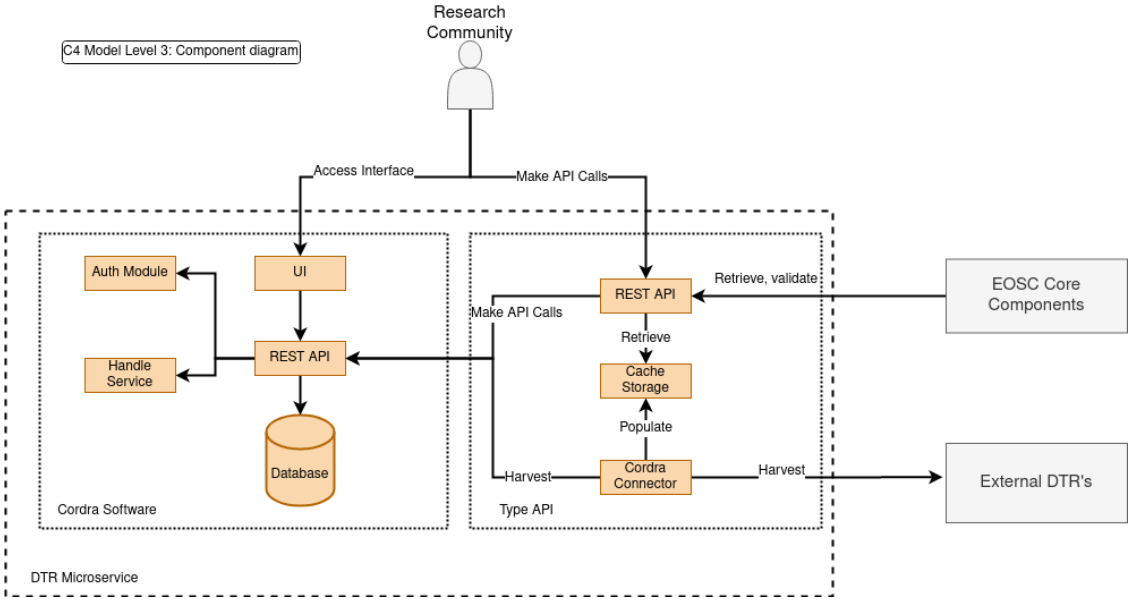

Figure 21

C4 Model 4: Code diagrams will be defined in the next stage of development.

Functional specifications

#	Short description	Priority	Reference
1	A user can, via a search bar, explore the set of registered types • User's should be given the possibility to filter results	H	
2	Creating new entries should be possible for users after authorization	H	
3	In the authorization form, users should enter their username and password	H	
4	When creating a schema, an authorized user can specify the design of the schema in JSON format	H	
5	When creating an instance of a type, authorized users should fill out a form with one input field per schema field.	H	
6	When performing a retrieval via the API, entries from multiple DTR's should be considered	H	

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

Service specifications

#	Short description	Priority	Reference

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Service specifications			
1	Retrieve the type description for a data type **GET** /api/v1/type/{pid} Parameters: • Format; to specify the format of the retrieved data • Human Readable; Specify if the results should be in human readable form	H	<JIRA issue number> or <URL to external reference>
2	Retrieve the validation schema for a data type **GET** /api/v1/schema/{pid} Parameters: • Format; to specify the format of the retrieved data • Human Readable; Specify if the results should be in human readable form	H	
3	Perform a search over all available DTRs **GET** /api/v1/search/ Parameters: • Query; The search query to be executed • Filter; Restrict the results	H	

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

Operational specifications			
#	Short description	Priority	Reference
1	The UI page should load in **3 seconds** with the total of < 200 number of simultaneous users.	[H|M|L]	<JIRA issue number> or <URL to external reference>
2	Any **GET** request (retrieving type descriptions, retrieving validation schemas, search) should return a result in **<10 seconds**.		

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

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	PID Graph can resolve type references to understand the semantics	H	<JIRA issue number>

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Integration with EOSC Core components			
2	AAI infrastructure to support the authentication, authorization of administrators for registration and updating of type descriptions and type instances	H	
3	EOSC Monitoring. The Data Type Registry via an api healthcheck checkpoint will provider its status for the EOSC Monitoring service.	M	

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

6.2.3 External references

External references		
#	Short description	Reference
1	RDA Data Type Registries working group output	https://doi.org/10.15497/A5BCD108-ECC4-41BE-91A7-20112FF77458
2	CORDRA - Highly configurable software for managing digital objects at scale.	https://www.cordra.org
3	A Persistent Identifier (PID) policy for the European Open Science Cloud (EOSC)	https://op.europa.eu/en/publication-detail/-/publication/35c5ca10-1417-11eb-b57e-01aa75ed71a1

6.3 T4.4 RAiD - Specifications

Component	EOSC Research Activity Identifier Service (RAiD)
Category	RAID
Contact person	Shawn Ross
Email address	Shawn Ross
Contributors	Shorn Tolley (ARDC), Rob Leney (ARDC), Matthias Liffers (ARDC), Siobhann McCafferty (ARDC), Melanie Barlow (ARDC), Natasha Simons (ARDC)
Version	1.0
Data	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Overview
A Research Activity Identifier (RAiD) is a globally unique, persistent identifier (PID) for research projects or activities. It consists of a unique identifier (a handle) and a metadata envelope. The metadata envelope links various components (e.g., contributors, organisations, grants, instruments, publications, datasets, etc.) to the project, via their own PIDs where available (e.g., ORCiDs, RORs, Crossref Grant IDs, DOIs, etc.). These components are only linked to the project itself; a RAiD does not assert links between components (e.g., contributors to organisations). RAiDs also contain project information not found elsewhere, such as project title, start date, description, and subject, but does not duplicate information found elsewhere. RAiDs can also be linked to one another using arbitrary qualified relationships so that, e.g., subprojects can be created. RAiD is governed by an ISO standard, which is in the final stages of approval. Under this standard, the Australian Research Data Commons (ARDC) is the global Registration Authority, which establishes policies and technical requirements governing RAiD services (with input from stakeholders). The ARDC is also a Registration Agency. Registration Agencies operate RAiD services. The standard envisions other Registration Agencies established around the world serving national, regional, or domain-specific communities. SURF is the proposed Registration Agency for FAIRCORE4EOSC WP4.4. In broad terms, a RAiD service allows creation, management, editing, search, and retrieval of RAiDs via either an Application Programming Interface (API) or graphical user interfaces (GUIs). It also must retrieve existing RAiDs and their metadata (wherever they were created globally), again via API or GUI, including the ability to search and discover RAiDs based on their metadata. The ARDC is leading the development and implementation of model RAiD services in collaboration with FAIRCORE4EOSC partners (especially SURF).

Objectives	
#	**Short description**
1	Support the creation, editing, discovery, and retrieval of RAiDs via API and GUI (web interface) by users. Creation involves minting the handle and attaching associated metadata. Editing involves updating of metadata. Discovery involves searching for a string across selected metadata elements to retrieve matching RAiDs. Retrieval involves resolving a RAiD from its handle and retrieving metadata.
2	Support bulk harvesting of RAiDs based on metadata content or RAiD association with a Registration Agency, Organisation, Service Point, or User. Harvesting by integrators (e.g., DataCite) via API will be prioritised, while harvesting based on simple parameters via a GUI will be explored.
3	Provide a console allowing Registration Agencies, Service Points, and elevated users to manage RAiD users appropriate to their roles (e.g., Registration Agency Administrators can manage users associated with their Registration Agency, Service Point Operators can manage users associated with their Service Point, and RAiD Content Managers can manage users associated with their particular RAiD).
4	Provide mechanisms allowing the Registration Authority to manage Registration Agencies and Registration Agencies to manage Service Points.
5	Develop an appropriate RAiD metadata schema, including (A) related contributors, organisations, inputs, and output, using PIDs where possible, and (B) key project information (e.g., title, description, etc.) not found elsewhere.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Objectives	
6	Support the ability to temporarily embargo a RAiD, or keep a RAiD confidential indefinitely. At minimum this objective requires making RAiDs non-discoverable (but resolvable); blocking access to embargoed / confidential RAiDs more comprehensively will be explored.
7	Produce a Registration Agency Handbook establishing RAiD policy and technical requirements.
8	Assess and revise RAiD governance to facilitate input from Registration Agencies, PID Providers, and other key stakeholders.
9	Establish a model RAiD service deployment demonstrating Objectives 1-6 above, including documentation and tools supporting its redeployment by Registration Agencies.

Out of Scope	
#	**Short description**
1	Search by API or GUI will be limited to specific metadata elements; full-text or faceted search across all elements is out of scope.
2	Blocking access to embargoed / confidential RAiDs comprehensively, with security adequate to address (e.g.) NDAs, will be explored but may be out-of-scope.
3	Bulk harvesting via GUI will be explored, but may be out-of-scope.
4	While documentation and tools for redeploying the model service will be provided, a turn-key RAiD service that can be trivially redeployed is out-of-scope.

6.3.1 Requirements

User stories		
#	**Description of the user story**	**Reference**
1	As the Registration Authority (ARDC), I need to be able to add and manage Registration Agencies so that I can allow, modify, or revoke the ability of Registration Agencies to mint RAiDs.	<URL to an external reference describing the user story in more detail>
2	As a Registration Agency (e.g., SURF), I need to be able to mint RAiDs at scale so that I can support numerous organisations.	
3	As a Registration Agency (e.g., SURF), I need to be able to resolve RAiDs at scale so that I can support numerous organisations, including harvesters / integrators (e.g., DataCite).	
4	As a Registration Agency (e.g., SURF), I need to be able to add and manage users so that I can allocate, change, or revoke access to RAiDs or RAiD service administration.	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
5	As a Registration Agency (e.g., SURF), I need to be able to add and manage Service Points so that I can allocate, change, or revoke organisational access to RAiD services we run.	
6	As a Registration Agency (e.g., SURF), I need a way to manage metadata schema versions, extensions, and vocabulary variations.	
7	As a Registration Agency (e.g., ARDC), I need to allow users to log in using a variety of identity providers (e.g., AAF, Google, ORCiD).	
8	As a Registration Agency (e.g., SURF), I would like to know the policy governing provision of RAiD services, so that I can meet my obligations under the ISO standard.	
9	As a Registration Agency (e.g., SURF) **or** a PID Provider (e.g., DataCite), I would like to influence RAiD governance, policy, and technical requirements so that the RAiD stakeholder community can work together effectively to ensure the availability of RAiD services and promote RAiD utilisation.	
10	As a Registration Agency (e.g., SURF), I would like to redeploy a model RAiD service with the minimum possible resourcing, rather than build one from scratch.	
11	As a Registration Agency (e.g., SURF), I would like to contribute to the codebase and documentation of the model RAiD service so that the service improves as rapidly as possible for everyone.	
12	As a Research Administrator (e.g., at TU Wien) participating in a national initiative involving other Research Offices (e.g., RIS Synergy), I need a single source of truth for research projects to coordinate projects across multiple universities for the purpose of grant applications, reporting, and other activities.	
13	As a Research Administrator (e.g., at TU Wien), I want an easy way for researchers to update information about their projects using a simple interface, so that maintaining project information is not too difficult or time consuming and researchers are more likely to do it.	
14	As a Research Administrator (e.g., at TU Wien), I want to be able to check project information entered by researchers, so that I know project information is accurate.	
15	As a Research Administrator (e.g., at the University of Auckland), I need to ensure that the information I have about collaborative research projects matches the information at collaborating universities.	
16	As a Research Administrator (e.g., at the University of Queensland), I want to update project information in a local Current Research Information System (CRIS) and then push those changes to a RAiD, so that I (or other users at my organisation) can produce and maintain RAiDs without interacting with an external system.	
17	As a Research Administrator, I want to be able to harvest and query project metadata from projects at my organisation so that I can produce reports connecting project inputs and outputs.	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
18	As a Research Administrator, I want to retrieve all RAiDs associated with my organisation so that I can generate reports on project activities and outputs.	
19	As a Research Administrator, I want to harvest RAiD data from my institution, so that I can holistically track, monitor, and report on projects.	
20	As a Research Administrator, I want to know how different research projects and activities are related to one another, and query across them, so that I can track impact across related projects.	
21	As a Research Administrator, I want to make project information available, so that projects can be more easily transferred from my institution to another institution or vice versa.	
22	As a Research Administrator, I want to quickly resolve a RAiD and retrieve its metadata in an easy-to-read format (e.g., a project 'landing page'), so that I can quickly get an overview of a project's activities.	
23	As a Research Administrator, I want information about all projects at my university to have a 'landing page' automatically generated from RAiD metadata, in order to improve project publicity and drive institutional web traffic.	
24	As a Research Administrator, I want to synchronise project information in our CRIS with our research finance system, so that I do not have to enter that information twice and then keep it in sync.	
25	As a Research Administrator, I want to embargo project information until a specific date (e.g., the announcement of a grant application outcome), limiting access only to specified, authenticated users.	
26	As a Research Administrator, I want to keep project information confidential indefinitely due to a Non-Disclosure Agreement, limiting access only to specified, authenticated users indefinitely.	
27	As a Research Administrator, I want to mint RAiDs, so that I can have a single identifier to track a project across its lifecycle.	
28	As a Research Administrator at a large research initiative (German Climate Computing Centre), I want to be able to connect large numbers of research activities to one another hierarchically, so that I can see and navigate the relationships between various programs, projects, and activities within my initiative. (Currently 3-4 levels with 10-50 items per level, perhaps going to 100-150 in the future per level.)	
29	As a Research Administrator at a large research initiative (German Climate Computing Centre), I want to be able to be able to programmatically interact with large numbers of research activities across a hierarchy, so that I can analyse relationships between various programs, projects, and activities within my initiative. (Currently 3-4 levels with 10-50 items per level, perhaps going to 100-150 in the future per level.)	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
30	As a Researcher, I want to pull information from a RAiD into my organisation's CRIS, so that the project record in the CRIS is up-to-date without my re-entry of that information.	
31	As a Researcher or Research Support Professional, I want to mint a RAiD once and reuse the information it contains repeatedly, avoiding double-entry of project metadata, so that I don't waste my time and make data entry errors.	
32	As a Researcher, I want to mint and maintain a RAiD for my project, so that I have all of the information about my project in a single place (including historical information).	
33	As a Researcher, I want to discover projects at other universities working in subject areas related to mine, so that I can seek out new opportunities for collaboration.	
34	As a Researcher, I want a clean web interface to enter and update RAiD project information, including the validation of connected PIDs, so that I can quickly enter accurate information about a project.	
35	As a Researcher, I want to automatically populate information about project contributors, publications, and other inputs or outputs in RAiD if I have previously associated a RAiD with an ORCiD, DOI, etc. (and potentially vice versa).	
36	As a National Research Funder (e.g., the Australian Research Council), a Government underwriting such funders, or a Private Funder, I want to minimise the cost of double-entry of data about research projects, so that grant (and other) funding can be used for more important things.	
37	As a National Research Funder (e.g., the Australian Research Council), I want to be able to report downstream outcomes from projects that I have funded in the past, so that longer-term impact of grants can be captured.	
38	As a Research Infrastructure Operator (e.g., EOSC), I want to assign a RAiD to those applying to use my infrastructure, so that I can track them and report on outcomes.	
39	As a Research Infrastructure Operator (e.g., EOSC), I want to assign a hierarchically related RAiD when a project applies for additional / multiple resources, so that I can track and report on outcomes associated with both the overall project and with each allocation of resources.	
40	As a Research Infrastructure Operator (e.g., EOSC), I can utilise common metadata schema elements, but must be able to implement customised metadata schema vocabularies that my organisation has developed (e.g., I need customised project output types aligning with the OpenAIRE schema).	
41	As a Research Infrastructure Operator (e.g., EOSC), I want to link from our services (e.g., in the EOSC Marketplace or EOSC User Portal) to a RAiD landing page so that our users can quickly view project / activity information.	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
42	As a Research Infrastructure Operator (e.g., EOSC), I want a link from the RAiD landing page to the RAiD editing page so that our users can conveniently edit RAiDs that belong to them.	
43	As a Research Infrastructure Operator (e.g., EOSC), I want users to be prompted with a reminder of any RAiDs they have already created, so that they can return to ongoing projects seamlessly and avoid duplicate RAiDs for the same projects / activities.	
44	As an eResearch Organisation (e.g., Queensland Cyber Infrastructure Foundation), I want to incorporate RAiDs into our research metadata management software (ReDBox), so that users of that software (universities) can mint RAiDs and update project metadata as part of their usual workflows using forms within the software.	
45	As the Operator of a Research Platform (e.g., Open Science Framework), I want to integrate RAiDs into my platform, so that my users can mint a PID for their own projects that accommodates changes in the project over time (rather than only designating a snapshot).	
46	As the Operator of a Research Platform (e.g., Open Science Framework) interested in open scholarship and data provenance, I want to track and expose the change history of a RAiD so that everyone can learn more about the history of the project.	
47	As an Integrator (e.g., DataCite), I want to be able to retrieve RAiDs associated with other PIDs (e.g., DOIs) and read their metadata.	
48	As an Integrator (DataCite), I want a single API service where I can retrieve RAiDs in bulk.	
49	As an Integrator (DataCite), I want a single, well-documented metadata schema with consistent elements and vocabularies.	
50	As a PID Provider (e.g., DataCite), I would like to know the technical specification of RAiD so that I can integrate RAiDs as easily as possible.	

User requirements				
#	**Short description**	**Priority**	**Feasibility**	**Reference**
1	Implement a mechanism for minting handles and applying them to a RAiD.	H	2	<JIRA issue number> or <URL to external reference>
2	Implement a metadata schema for RAiD records that includes key project information.	H	2	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
3	Support versioning of the RAiD metadata schema.	H	3	
4	Enforce common shared ‘core’ metadata elements	H	2	
5	Support ‘extended’ metadata elements (customised by Registration Agencies)	M	3	
6	Support customisation of metadata schema vocabularies by Registration Agencies	H	3	
7	Output customised metadata schema vocabularies using a single, shared vocabulary	H	3	
8	Support arbitrary, qualified / named relationships between RAiDs.	H	3	
9	Resolve RAiDs and retrieve associated metadata via GUI	H	2	
10	Resolve RAiDs and retrieve associated metadata via API	H	2	
11	Create and edit RAiDs via GUI	H	2	
12	Create and edit RAiDs via API	H	2	
13	Validate connected PIDs with originating service when creating or editing a RAiD	H	3	
14	Automatically import connected PIDs when editing a RAiD, based on the presence of the current RAiD in the metadata of the external PID	L	5	
15	Expose RAiD metadata so that users editing PIDs in other services can automatically import associated RAiDs, based on the presence of the external PID in the RAiD	L	5	
16	Search and retrieve a RAiD based on metadata (including PIDs contained in the RAiD) via GUI	H	3	
17	Search and retrieve a RAiD based on metadata (including PIDs contained in the RAiD) via API	H	3	
18	Retrieve all RAiDs associated with a Registration Agency, Organisation, Service Point, or User (via API and, resources permitting, GUI)	H	2	
19	Retrieve all RAiDs expressly related to a given RAiD (via API and/or GUI)	H	2	
20	Easily navigate between viewing a RAiD and editing a RAiD.	H	1	
21	Maintain a change history for each RAiD.	M	4	
22	Expose RAiD change history via API	M	4	
23	Expose RAiD change history via GUI	L	4	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
24	Ensure all GUIs conform to good UI/UX practices.	H	2	
25	Document API sufficient to allow independent integrations.	H	1	
26	Mint RAiDs and write metadata via API at a specified / documented rate.	H	2	
27	Resolve RAiDs and return metadata via API at a specified / documented rate.	H	2	
28	Retrieve explicitly related RAiDs at a specified / documented rate up to a specified / documented scale (e.g., number of levels in a hierarchy and number of RAiDs per level).	H	3	
30	Provide user creation and management, implementing roles and permissions common across Registration Agencies.	H	2	
31	Allow implementation of new roles and associated permissions.	M	3	
32	Provide a GUI for user, role, and permission management.	M	3	
34	Allow login via multiple providers, initially AAF, ORCiD, and Google.	H	3	
35	Provide a GUI for Service Point creation and management (for Registration Agencies).	H	2	
36	Provide a mechanism for the temporary embargo of sensitive metadata associated with a RAiD.	H	2	
37	Provide a mechanism for the permanent confidentiality of sensitive metadata associated with a RAiD.	M	4	
38	Provide a mechanism (GUI or other) facilitating the Registration Authority's ability to create, modify, or delete Registration Agencies.	L	5	
39	Availability of a Registration Agency Handbook, including policy and technical requirements.	H	2	
40	Institution of a governance model allowing community input.	H	2	
41	Availability of a model RAiD service operated by the ARDC.	H	3	
42	Availability of documentation and tools to make redeployment of the model RAiD service easier.	L	3	
43	Implementation of a collaborative development environment (e.g., GitHub) allowing contributions to code and documentation.	M	1	

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

Funded by the European Union

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
#	**Short description**	**Priority**	**Reference**
1	When a user **creates a RAiD**, a handle must be minted, metadata created, and the handle and metadata associated with one another (via API or GUI).	H	<JIRA issue number> or <URL to external reference>
2	When a user **resolves a RAiD** based on a known handle, the associated metadata must be returned (globally, via API or GUI).	H	
3	When a user **searches for RAiDs** based on a metadata element, Registration Agency, Organisation, Service Point, or User, all matching RAiDs and their metadata should be returned (globally, via API or GUI).	H	
4	When a user resolves a RAiD or selects a RAiD from search results using a GUI, a well-designed **'landing page'** presenting RAiD metadata should be displayed.	H	
4	When a user needs to **edit a RAiD**, metadata elements must be updated (via API or GUI).	H	
5	When a user creates or edits a RAiD, **validate external PIDs** with the originating service.	H	
6	When a user is viewing a RAiD landing page and wants to edit the RAiD, easy **navigation to an editing interface** should be provided (if the user has the required permissions).	H	
7	When a user requests a **data dump** of a large number of RAiD metadata records (beyond a specified threshold, e.g., from a search), that data dump must be prepared in the background and made available for later download (globally, via API or GUI).	H	
8	When a user has AAF, ORCiD, or Google credentials, allow them to **log into a RAiD service**.	H	
9	When a user with elevated permissions **creates, modifies, or deletes a user** via a GUI, the system must confirm the privileges of the elevated user and allow the change only if those privileges are sufficient.	H	
10	When a user with elevated permissions associated with a Registration Agency **creates, modifies, or deletes a Service Point** via a GUI, the system must confirm the privileges of the elevated user and allow the change only if those privileges are sufficient.	H	
11	When a user associated with a Registration Agency needs to **update the metadata schema**, a mechanism must be in place to implement the new version.	H	
12	When a user associated with a Registration Agency needs to **implement customised 'extended' metadata elements (non-'core' elements specific to**	H	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
	the Registration Agency), a mechanism must be in place to implement the customisations.		
13	When a user associated with a Registration Agency needs to **implement customised metadata schema vocabularies associated with a 'core' element**, a mechanism must be in place to implement the customisations.	H	
14	When an integrator **harvests RAiDs**, all 'core' metadata elements and associated metadata schema vocabularies must be returned consistently, without variation, across all Registration Agencies.	H	
15	When an integrator **harvests RAiDs**, all 'extended' metadata schema elements and vocabularies should be returned as key-value pairs without any standardisation.	M	
16	When a user **flags a RAiD as confidential** (temporarily or permanently), ensure that only specified, authenticated users can view sensitive RAiD metadata	M	
17	When a user wants to view a **RAiD's change history**, provide a means to do so (via either API or GUI, both if possible).	L	
18	When a user edits a RAiD, provide the ability to **automatically import external PIDs** that contain the RAiD in their metadata.	L	
19	When a user edits an external PID, provide the ability to **automatically export a RAiD** containing that PID in metadata to the external service.	L	
20	Given that Registration Agencies would like to redeploy rather than redevelop RAiD services, the ARDC must develop a **model RAiD service** that meets requirements articulated in Component 2 below. This instance will be operated by the ARDC in its role as a Registration Agency. Code will be open-source and published on GitHub.	H	

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

Non-functional requirements			
#	**Short description**	**Priority**	**Reference**
1	The minting, resolution, and bulk harvesting services should perform to tested / declared response times aligning with industry standards (within reasonable effort).	H	<JIRA issue number> or <URL to external reference>
2	Preparation of data dumps must be efficient and not impact the performance of the system.	H	
3	UI/UX of all GUIs should follow good practices making use as intuitive and efficient as possible.	H	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Non-functional requirements			
4	An approach (policy and technical) to metadata schema versioning ensuring orderly and predictable version updates.	H	
5	Documentation of metadata schema, including the distinction between 'core' and 'extended' metadata, and the customisation / standardisation of 'core' metadata schema vocabularies.	H	
6	A definition of what, if any, metadata must be displayed if a RAiD is flagged confidential, versus what metadata is sensitive and can be withheld.	M	
7	Documentation of in-built roles and permissions.	H	
8	Guidance regarding the level of security provided for embargoed and confidential RAiDs.	H	
9	API documentation sufficient to allow independent integrations.	H	
10	A Registration Agency Handbook produced by the ARDC in its role as Registration Authority, describing RAiD policy, technical requirements, service setup, and service management	H	
11	Basic user-facing documentation of RAiD service.	M	
12	A governance framework developed by the ARDC that allows input into RAiD policy and technical requirements from the RAiD stakeholder community. The current Advisory Group arrangement will be reviewed and assessed to see if it remains fit-for-purpose as the ARDC transitions into its role as Registration Authority, or if a more formal arrangement is needed. Other PID providers (e.g., DataCite, Crossref, ORCiD) will be reviewed as potential models.	H	
13	Documentation and tooling produced by the ARDC to assist other Registration Agencies in the redeployment of the model RAiD service.	L	
14	GitHub repositories for code and documentation, including use of GitHub Issues, Projects, and other collaboration tools (at https://github.com/au-research/). .	M	

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

6.3.2 Specifications

Architectural design
https://github.com/au-research/raido-v2/blob/main/doc/architecture/raido-container-c4.md
'Raido' (RAiD-Oceania) is the **model RAiD service** under development that will be used by the ARDC in its role as a Registration Agency, and made available for reuse at other Registration Agencies.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
#	**Short description**	**Priority**	**Reference**
1	Create RAiD (handle and metadata) via API	H	<JIRA issue number> or <URL to external reference>
2	Create RAiD (handle and metadata) via GUI	H	
3	Resolve RAiD from handle via API, returning metadata	H	
4	Resolve RAiD from handle via GUI, displaying metadata	H	
5	Search and retrieve RAiDs based on metadata content, Registration Agency, Organisation, Service Point, or User via API, returning handle and metadata	H	
6	Search and retrieve RAiDs based on metadata content, Registration Agency, Organisation, Service Point, or User via GUI, displaying handle and metadata	H	
7	Provide a data dump (potentially asynchronous) when metadata records from many RAiDs are requested via API, with ‘core’ metadata schema elements and vocabularies standardised	H	
8	Provide a data dump (potentially asynchronous) when metadata records from many RAiDs are requested via GUI (if possible within resource constraints), with ‘core’ metadata schema elements and vocabularies standardised	M	
9	Edit RAiD metadata via API	H	
10	Edit RAiD metadata via GUI	H	
11	Validate external PIDs when creating or editing RAiDs via either API or GUI.	H	
12	Navigate easily between viewing and editing a RAiD’s metadata	H	
13	Maintain RAiD metadata content change history	M	
14	Expose RAiD metadata content change history (via API or GUI)	L	
15	Support RAiD metadata schema versioning	H	
16	Manage transitions to new RAiD metadata schema versions	H	
17	Support extension of the RAiD metadata schema elements	M	
18	Support customisation of RAiD metadata schema vocabularies	H	
19	Log into RAiD using AAF, ORCiD, or Google credentials	H	
20	Create, modify, and delete users via GUI using predefined roles and permissions	H	
21	Create, modify, and delete roles and permissions via GUI	M	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
22	Create, modify, and delete Service Points via GUI	H	
23	Create, modify, and delete Registration Agencies (by any feasible mechanism)	L	
24	Enable temporary embargos of RAiD metadata contents	H	
25	Enable permanent confidentiality of RAiD metadata contents	M	
26	Automatically import PIDs related to a RAiD (including external PIDs and other RAiDs)	L	
27	Automatically export RAiDs to external PIDs related to the RAiD.	L	

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

Service specifications			
#	**Short description**	**Priority**	**Reference**
1	API service for resolving RAiD handles, searching RAiDs, returning RAiD metadata, creating RAiDs, and editing RAiDs	H	<JIRA issue number> or <URL to external reference>
2	GUI (e.g., web application or static web pages) for resolving a RAiD handle and displaying that RAiD's metadata content, thus providing a RAiD 'landing page'.	H	
3	GUI (e.g., web application or static web pages) for searching RAiDs by metadata content, Registration Agency, Organisation, Service Point, or User and returning matching RAiDs, any of which can then be 'clicked through' to reach that RAiD's landing page.	H	
4	GUI (e.g., web application) for editing RAiDs.	H	
5	Interoperability with other PID services for validation and automated import / export / linking of related PIDs.	M	
6	API service for requesting bulk 'data dumps' based on metadata content (including associated PIDs), Registration Agency, Organisation, Service Point, or User in JSON format.	H	
7	GUI (e.g., web application) for requesting bulk 'data dumps' based on simple parameters (resources permitting).	M	
8	Bulk data dumps standardise 'core' metadata schema elements and vocabularies for harvesting by integrators.	H	
9	Bulk data dumps report 'extended' metadata schema elements as key-value pairs as they are stored in the service.	M	
10	GUI (e.g., web application) for managing (creating, updating, deleting) users.	H	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Service specifications			
11	GUI (e.g., web application) for managing (creating, updating, deleting) user roles and permissions.	M	
12	GUI (e.g., web application) for managing (creating, updating, deleting) Service Points.	H	
13	Mechanism (TBD) for managing (creating, updating, deleting) Registration Agencies.	L	
14	Documentation of metadata schema	H	
15	Documentation of predefined user roles and permissions	H	
16	Tools and documentation to assist redeployment of model RAiD service.	L	
17	Registration Agency Handbook including policy and technical requirements.	H	
18	Revised RAiD governance framework facilitating input from Registration Agencies, PID providers, and other key stakeholders.	H	

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

Operational specifications			
#	**Short description**	**Priority**	**Reference**
1	Model RAiD service will have APIs available for content access as per service availability guidelines (potentially with some restrictions depending upon running costs)	H	<JIRA issue number> or <URL to external reference>
2	Model RAiD service will have GUIs available for content access as per service availability guidelines (potentially with some restrictions depending upon running costs)	H	
3	Model RAiD service will ensure minting, resolution, and bulk harvesting services (API or GUI) should perform to tested / declared response times	H	
4	Model RAiD service will ensure bulk harvesting services do not appreciably impact performance of the system for other users.	H	
5	Metadata schema will be reviewed and updated to ensure it remains fit-for-purpose, and versioned when changes are made.	H	

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

Integration with EOSC Core components			
#	**Short description**	**Priority**	**Reference**
1	Incorporate RAiDs into RDGraph	H	<JIRA issue number> or

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Integration with EOSC Core components			
			<URL to external reference>
2	Support RAiD harvesting by PIDGraph	H	
3	Publish metadata schema to MSCR and utilise its crosswalk support where beneficial	H	
4	Support resolution of RAiDs via PIDMR	H	
5	Provide any information requested by CAT as part of their evaluation of PIDs.	M	

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

6.3.3 External references

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

6.4 T4.5 Demonstrator - Specifications

The WP4 demonstrators only need a limited set of additional specifications beyond those that the WP4 services and components (MSCR and DTR) already provide in their respective task descriptions in this deliverable. The specifications of required changes and interactions with 'external' components and services including any human workflows will be discussed and further specified in 2023. But can in essence be found amongst the Demonstrator requirements specified in

6.4.1 MSCR seeding demonstrator T4.5.1

Component exercition	MSCR seeding demonstrator (4.5.1)
Category	MSCR
Contact person	Daan Broeder
Email address	Daan Broeder
Contributors	Chris Ariyo Heinrich Widmann Sven Bingert
Version	
Data	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
#	Description of the user story	Reference
1	This demonstrator demonstrates the ability of the MSCR to register and host metadata schema and crosswalks of different types and formats, making them FAIRer Semantic Artefacts. Users are able to use the MSCR API or the MSCR GUI to register and host semantic artefacts: metadata schema, the metadata schema related vocabularies and metadata schema crosswalks. Adding core semantic artefact metadata and provenance (tbd) is part of the registration process. among the semantic artefacts initially made available in the MSCR are: metadata schema and crosswalks from RDA metadata standard registry, B2FIND, B2SHARE (proposal text)	<URL to an external reference describing the user story in more detail> FC4E 4.5 demonstrator scripts part 4.5.1
2	The MSCR is a FAIR registry and provides users with necessary basic data management: publication (PID), access management, metadata and provenance information, searching and browsing and download the semantic artefacts in suitable formats: 1. the original deposited format 2. an upgraded normalised RDF representation, which is machine actionable (although not necessarily so for humans)	
3	Users can inspect the content of the semantic artefacts and metadata and provenance info with a suitable viewer	
4		

User requirements				
#	Short description	Priority	Feasibility	Reference
1	MSCR registration and hosting of semantic artefacts via API and GUI: metadata schema, metadata schema related vocabularies, schema crosswalks. support for registering/hosting: XSD, SKOS, RDF metadata schema support for registering/hosting: XSLT, X3M, CSV/TSV, RDF semantic crosswalks/mappings	H	[1-5]	CR 4.5.1, CR 4.5.2, CR 4.5.6 <JIRA issue number> or <URL to external reference> MSCR-FR - T4.2.R-1 - T4.2.R-2 - T4.2.R-11 - T4.2.R-12 - T4.2.R-13 - T4.2.R-42 - T4.2.R-34

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
2	MSCR issues PIDs and automatic metadata and provenance for every registered/hosted semantic artefact, and/or allows the user to provide it	H		MSCR-FR - T4.2.R-17 - T4.2.R-18 - T4.2.R-19 - T4.2.R-20 - T4.2.R-22
3	MSCR supports browsing and searching for semantic artefacts via GUI and API	H		MSCR-FR - T4.2.R-24 - T4.2.R-25 - T4.2.R-26 - T4.2.R-28
4	inspecting the semantic artefacts using suitable viewer(s)	M		MSCR-FR - T4.2.R-24 - T4.2.R-25
5	semantic artefacts initially made available in the MSCR are: metadata schema and crosswalks from RDA metadata standard registry, B2FIND, B2SHARE (proposal text)	M		

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

6.4.2 MSCR Managing B2FIND Schema and crosswalks T4.5.2

Component exercition	Managing B2FIND schema and crosswalks (4.5.2)
Category	MSCR
Contact person	Daan Broeder
Email address	Daan Broeder
Contributors	Heinrich Widmann Sven Bingert
Version	
Data	

User stories		
#	Description of the user story	Reference
1	This demonstrator shows the ability of the MSCR to support the B2FIND data managers(s) in managing the process of creating and administrating crosswalks	<URL to an external reference describing

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
	necessary for converting the B2FIND harvested community metadata to the EUDAT core metadata format. 1. B2FIND data managers use the MSCR to administrate the new community metadata schema: registration, metadata updating , etc. 2. B2FIND data managers use the MSCR (smart mapping tool) to create the (new) crosswalk and add (if needed) administrative metadata 3. B2FIND data managers view & download the crosswalk in a suitable for them understandable and usable format	the user story in more detail> FC4E 4.5 demonstrator scripts part 4.5.2
2	B2FIND manager registers B2FIND's community metadata schemas, EUDAT core md schema and mappings between them : B2FIND supports several standardised metadata schemas for the uptake of community specific metadata. For each new (meta)data provider/repository B2FIND team adapt the mapping of the specific community metadata fields to the EUDAT core metadata schema. These mappings are specified by crosswalk tables and implemented as reader and writer python methods. If these crosswalks can be uploaded encoded as python code to the MSCR is still under consideration.	
3	B2FIND manager utilise (versions of) metadata schemas and crosswalks maintained in the MSCR. Example use case : The DataCite MD schema is one of the formats supported by B2FIND and maintained in MSCR. B2FIND uses different versions, flavors and extensions of the schema hosted in MSCR. B2FIND benefits This means that B2FIND benefits from the fact that the mainteance and versioning of schemas and crosswalks is managed centrally in MSCR and does not need to be managed by B2FIND itself.	
4	When using B2FIND it's sometimes not clear how community metadata is used by B2FIND, and mapped on the EUDAT core metadata schema that is used by B2FIND. To give insight and detect errors community experts are able to check which crosswalk is used for converting their community metadata to the EUDAT core metadata used in B2FIND, and if needed send a message to the B2FIND data manager responsible	

User requirements				
#	Short description	Priority	Feasibility	Reference
1	MSCR registration and hosting of (community) metadata schema and performing basic administrative functions on these: metadating, versioning. (a metadata schema for MSCR semantic artefacts is tbd) MSCR API , GUI. Depending on the B2FIND workflow this registration/uploading takes place via the MSCR API and/or GUI	H	[1-5]	<JIRA issue number> or <URL to external reference>

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
	support for registering/hosting B2FIND used: XSD, SKOS format metadata schema support for registering/hosting B2FIND used: XSLT format semantic crosswalks/mappings			MSCR-FR - T4.2.R-11 - T4.2.R-2 - T4.2.R-3 - T4.2.R-4 - T4.2.R-6 - T4.2.R-7 - T4.2.R-11 - T4.2.R-12 - T4.2.R-54
2	(later tbd) support for specific B2FIND proprietary operationalisable crosswalk format	L		MSCR-FR - T4.2.R-46 - T4.2.R-48 - T4.2.R-54
3	MSCR supports creation of crosswalks on the basis of the hosted XSD metadata schema provided by B2FIND team	H		MSCR-FR - T4.2.R-56
4	MSCR supports inspecting the a metadata crosswalk in sufficient detail, in a suitable format eg. baseline table format by a community expert	M		MSCR-FR - T4.2.R-24
5	MSCR hosts, maintains and provides different versions of meta schema and B2FIND retrieves and utilise the appropriate version for it's specific needs in the ingestion process.	M		MSCR-FR - T4.2.R-20 - T4.2.R-21 - T4.2.R-23 - T4.2.R-17

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

6.4.3 MSCR Managing B2SHARE schema T4.5.6

Component exercition	Managing B2SHARE schema (4.5.2.1) (variant on 4.5.2 for B2SHARE)
Category	MSCR
Contact person	Daan Broeder
Email address	Daan Broeder
Contributors	Chris Ariyo
Version	
Data	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
#	Description of the user story	Reference
1	This demonstrator shows the ability of the MSCR to support the B2SHARE data managers(s) in managing the B2SHARE community metadata schema 1. B2SHARE data managers use the MSCR to administrate the new community metadata schema: registration, metadata updating , etc. (schema upload) 2. B2SHARE data managers use the MSCR to create new community metadata schema	FC4E 4.5 demonstrator scripts part 4.5.6
2	B2FIND data managers use the MSCR (smart mapping tool) to create the (new) crosswalks to convert between metadata records from different communities different B2SHARE metadata schema	
4		

User requirements				
#	Short description	Priority	Feasibility	Reference
1	MSCR registration and hosting of (community) metadata schema and performing basic administrative functions on these: metadating, versioning. (a metadata schema for MSCR semantic artefacts is tbd) MSCR API and GUI can be used for registration/uploading of B2SHARE metadata schema (JSON format)	H	[1-5]	<JIRA issue number> or <URL to external reference>
2	MSCR supports additionally creation of B2SHARE community metadata schema	L		
3				

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

6.4.4 Using the DTR for typing metadata schema element and attribute values T4.5.3

Component exercition	Using the DTR for typing metadata schema element and attribute values (4.5.3)
Category	DTR, B2SHARE
Contact person	Daan Broeder
Email address	Daan Broeder
Contributors	Chris Ariyo

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Version	
Data	

User stories		
#	Description of the user story	Reference
1	this demonstrator shows the use of the DTR in typing metadata schema elements and attributes. When creating metadata crosswalks the type/value schemes of the metadata elements and attributes can be required for the actually conversion of metadata later-on. A user can specify the data-type/value schema of metadata schema's hosted in the MSCR. The MSCR uses the DTR API to provide the user with an interface to select a suitable data-type. The data-type reference is stored in the metadata schema representation and can be visualised when inspecting the metadata schema.	FC4E 4.5 demonstrator scripts part 4.5.3
2	When creating a crosswalk using the MSCR the user is prompted for a data-type reference if this is not yet known, for every mapping that is created. The user is prompted by the MSCR to choose a suitable data-type. The user can also ignore the prompt.	

User requirements				
#	Short description	Priority	Feasibility	Reference
1	The DTR has a suitable schema (currently under discussion) for basic data-types and users can create new data-types (basic data-type schema instances)	H	[1-5]	<JIRA issue number> or <URL to external reference>
2	The DTR provides an API which allows other tools (ie MSCR) to browse / search the set of basic data-types	H		
3	The MSCR provides the user with an option to associate all metadata schema elements and attributes with a reference to a DTR registered basic data-type via a pick list	H		MSCR-FR - T4.2.R-35
4	The MSCR crosswalk creation process checks if the metadata elements and attributes involved in a mapping, have an associated data-type reference and if not allow the user to select a suitable data-type from a pick-list.	M		MSCR-FR - T4.2.R-35
4				

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

6.4.5 The DTR as registration authority for extended mime-types T4.5.4

Component exercition	The DTR as registration authority for extended mime-types (EDFR) (4.5.4)
Category	DTR
Contact person	Daan Broeder
Email address	Daan Broeder
Contributors	Sven Bingert Chris Ariyo
Version	
Data	

User stories		
#	Description of the user story	Reference
1	DTR data managers seed (from IANA) and maintain an extended mime-type vocabulary (EDFR). Community experts can add to this vocabulary. Community software and experts can search and browse the vocabulary using relevant APIs for the benefit of authoritative data format typing of resources and capabilities of services.	FC4E 4.5 demonstrator scripts part 4.5.4
2		

User requirements				
#	Short description	Priority	Feasibility	Reference
1	DTR to implement extended mime-type schema and DTR support community expert contributions and establish editorial rules	H	[1-5]	<JIRA issue number> or <URL to external reference>
2	DTR to serve vocabulary of extended mime-types in SKOS format and support other APIs eg. SKOSMOS API (tbd) for benefit of client software	H		
3	client software to be updated: CLARIN Language Resource Switchboard (task7.1.2) and SSHOC ConversionHub (optional demonstrator part)	L		

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

6.4.6 B2Inst DTR demonstrator T4.5.5

Component exercition	B2Inst DTR demonstrator (4.5.5)
Category	DTR
Contact person	Daan Broeder
Email address	Daan Broeder
Contributors	Sven Bingert
Version	
Data	

User stories		
#	Description of the user story	Reference
1	To achieve interoperability between different EOSC Core and B2 services, a centrally published maintained and versioned schema for the description of instruments should be used. Tool builders (and tool users) can make use of unique identifiers that are provided B2Inst provides for instruments. B2Inst uses internal described profiles. The DTR would provide registration and management of instrument profiles and validation. B2Inst should load the profile from the DTR and dynamically build a Form UI for users for them to enter the instrument details. Versioning of profiles should be possible.	FC4E 4.5 demonstrator scripts part 4.5.5 It is a result of an RDA Working Group PIDinst and is now a service implementation from EUDAT: https://www.rd-alliance.org/PID-instruments-May2022_webinar, https://eudat.eu/news/new-fair-registry-tool-for-scientific-instruments-under-development
2		

User requirements				
#	Short description	Priority	Feasibility	Reference
1	Registration of types required by the instrument profile		[1-5]	<JIRA issue number> or <URL to external reference>
2	Registration of the instrument profile			
3				
4				

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
4				

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

7 WP5 – EOSC PID Meta Resolver and PID Kernel Information Profiles

7.1 T5.2 PID Meta Resolver - Requirement Analysis

The increasing use of PIDs to reference all types of research results is a major step forward in meeting future requirements for the FAIRness of (research) data. Although this trend is very welcome, new challenges arise in processing these PIDs and integrating them into different research processes. The biggest difficulty is the different systems used to create and maintain PIDs. The challenge is to know which system is responsible for the resolution process, the process that provides the referenced (meta-)data for a PID. A uniform interface that allows PIDs from different systems to be resolved can simplify various processes. The PID Meta Resolver is such an interface and will be extended in the project to integrate different systems.

Component	PID Meta Resolver
Category	PIDMR
Contact person	Themis Zamani, Sven Bingert
Email address	Themis Zamani Sven Bingert
Contributors	GWDG, OpenAIRE, CSC, NLF, GRNET, DataCite, SURF, Software Heritage/Inria
Version	1.0.0
Data	

Overview
The PID Meta Resolver is a generalized resolver for mapping items into records. Actually the PID Meta Resolver will know where to route different types of identifier – ex. DOI, URN:NBN. PID Meta Resolver which should improve machine based data processing and allows to get digital object information without in-depth knowledge of the resolution mechanism of different PID systems. That enhances the collection and analysis of data collections originating not only from different sources also referenced by different PID systems. The PID Meta Resolver should return a minimal set of information. This creates the connection with the PID Kernel Information.

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Objectives	
#	Short description
1	Provide a generic scalable meta-resolver that can be used to access digital objects referenced by different PID systems.
2	Determine the structure of the information that can be expected from a resolution request
3	Provide a generic and sustainable infrastructure for information about digital objects
4	Demonstrate the added value of the newly developed Meta-Resolver a demonstrators and stakeholders.
5	Offer the newly developed Meta-Resolver as EOSC core services

Out of Scope	
#	Short description
1	Registration of PIDs of any PID system
2	Create/Update/Delete of any PID related Metadata
3	List of specific PID systems which do not provide a resolving mechanism
4	Resolving Short-URLs
5	As a funder I want to get metadata of all datasets, projects and publications an applicant is related to.
6	As a researcher I want to find data/articles, so I can access them.
7	As a life sciences researcher, I have a compact URI (CURIE) for a biomedical entity. I want to resolve the biomedical entity to a web page. I need a service that's robust to synonyms, highly responsive to community feedback, and possible to PR.
8	As a project call manager, I want to identify all projects components (pearsons, experiments, scientific objects), so I can manage means and time
9	As a researcher I want to find all products resulting from a given project
10	As a researcher I want to resolve PIDs from a secondary metadata exchange schema (e.g. FAIRtracks), so that I can locate the original record and metadata
11	As a <RESEARCHER> i want <canonical URIs> so I can <do complex queries across resources>

7.1.1 Requirements

User stories		
#	Description of the user story	Reference
	As a <USER>, I want to be able to <GOAL> so that <REASON> .	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
1	As a <USER>, I want to be able to <list the entry points of the MetaResolver> so that <I understand the possible interactions>	
2	As a <USER>, I want to be able to <see the included PID systems> so that <it can be validated if system is available>	
3	As a <USER>, I want to be able to <resolve PIDs using MetaResolver> so that <I can get resolution services available, such as the URL of the referenced object> As a <USER> i want to <add any PID string into a single location> so I <can be routed to the correct resolver and from there to the correct resource>. As a <RESEARCHER> I want to <get the service I want, such as the resource the PID> was <created for> (not 404).	
4	As a <USER>, I want to be able to <resolve PIDs using MetaResolver> so that < it return descriptive metadata>	
5	As a <USER>, I want to be able to <resolve PIDs using MetaResolver> so that < it return some data / queries from the descriptive metadata>	
6	As a <PID Service Provider>, I want to able to <provide a scalable solution> which is <able to reply to high load of requests>	
7	As a <PID Service Provider>, I want to be able to <upload my service to the MetaResolver> so that <users can resolve my own PIDs using the MetaResolver>	
8	As <USER> I want to be <able to take any ID I find> and <quickly consistently and securely know what this ID identifies>.	
9	As a <MACHINE> I want to <send a PID to the MetaResolver> and <get a url of the resource>	
10	As a <MACHINE> i want to <send a PID to the MetaResolver> and <get the metadata of the resource>	
11	As a <PID Service Provider> I want to be able to resolve (and obtain kernel information) to the bit stream of the digital object to support our machine based workflows.	
12	As a <MACHINE> I want to able to <resolve FDOs> so that <I can automatically retrieve metadata about an digital object>	
13	As a <Machine> I want to <send a PID> so that <I can get resource specific service>	

User requirements: Split above phrases in smaller atomic tasks/actions				
#	Short description	Priority	Feasibility	Reference
	<Provide a short description of the user requirement>	[H|M|L]	[1-5]	<JIRA issue number> or

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements: Split above phrases in smaller atomic tasks/actions				
				<URL to external reference>
1	The user needs a ui to understand how the service is working			
2	The user needs a page to understand which identifiers it supports			
3	The user needs a ui form and an input for the value of the PID to resolve			
4	The user needs a report from the UI with the details of the metadata of the resource.			

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

Functional requirements: **Functional requirements define what a software must do: its features and functions. An example of a functional requirement for a messenger will be something like, *"A user must be able to edit messages after they are sent to correct errors."***			
#	Short description	Priority	Reference
	<A Functional Requirements are the actions a system should implement to support the UserStories>	[H|M|L]	<JIRA issue number> or <URL to external reference>
1	The system should know how to understand the different types of Persistent Identifiers	H	
2	The system should support the schema , the structure of the information a PID resolution system supports	H	
3	The system should support a procedure on how to enable a new PID Service provider and the resolution service	H	
4	The system should support a procedure on how to enable a new PID Service provider and the support of the metadata of the resource	M	
5	The system should resolve DOIs	H	
6	The system should resolve Handles	H	
7	The system should resolve **URN:NBN**	H	
8	The system should resolve **SoftWare Heritage persistent IDentifiers**	H	
9	The system should resolve ORCIDs	L	
10	The system should resolve RORs	L	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements: Functional requirements define what a software must do: its features and functions. An example of a functional requirement for a messenger will be something like, *"A user must be able to edit messages after they are sent to correct errors."*			
11	The system should resolve FDOs	M	
12	The system should resolve RAID	M	
13	The system should resolve ISNIS	L	

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

Non-functional requirements: Non-functional requirements specify the quality attributes of the system, hence their second name — *quality attributes*. Continuing our messaging platform example, a non-functional requirement can be the speed with which a system must perform editing to satisfy user expectations, *"The message must be updated for all users in a chat within 0.1 seconds, given that all users are online and have LTE connection or better."*			
#	Short description	Priority	Reference
	<Provide a short description of non-functional requirements>	[H|M|L]	<JIRA issue number> or <URL to external reference>
1	Correctness; that the MetaResolver API respects the specification	H	
2	Performance; that the MetaResolver replies in specified times	H	
3	Scalability; that the MetaResolver can answer high number of request per given time unit	H	
4	Usability; that the MetaResolver provides necessary endpoints for different user needs	M	
5	Robustness; that the MetaResolver is able to handle errors and failures	M	

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

7.1.2 Specifications

Architectural design
The design follows the methodology of the C4Model (https://c4model.com)

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

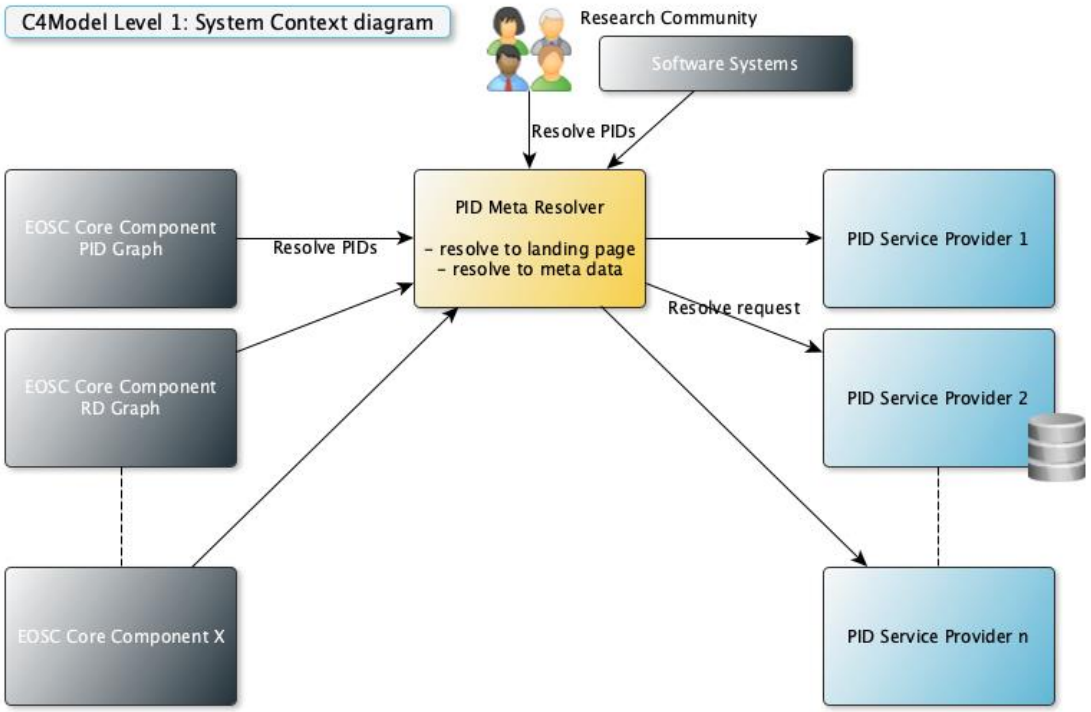

Figure 22

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

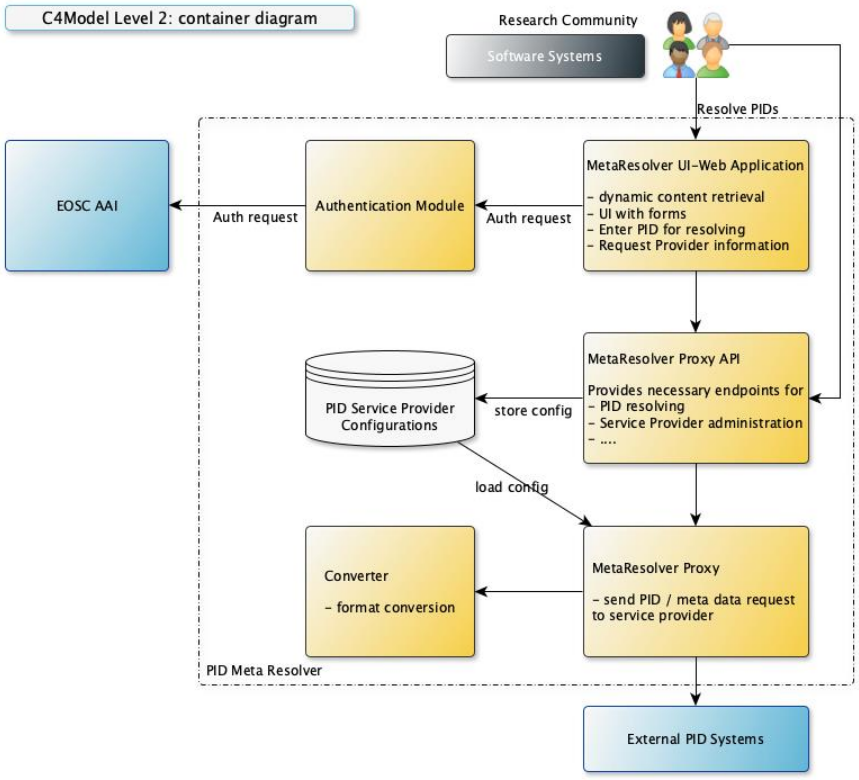

Figure 23

In Level3 of the C4Model each container of Level2 will be described by an 'Component Diagram'. Level 3 and Level 4 will be derived in the next steps

Functional specifications: **Functional specifications describe the functionality your users should experience. This doesn't include design elements or technical information, such as what software will be used, just the basic functional requirements. It needs to be looked at from the user's perspective, with an understanding of potential features, screens, menus and dialogue.**

- **What fields need to be included on the form?**
- **What happens when a user enters the wrong information?**
- **What is displayed if an error occurs?**

#	Short description	Priority	Reference
	<Provide a short description of functional technical specification> <Comment by Sven: It specifies the functions that a system or component must perform> <Comment by Sven: more detailed user requirement>	[H|M|L]	<JIRA issue number> or <URL to external reference>

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications: Functional specifications describe the functionality your users should experience. This doesn't include design elements or technical information, such as what software will be used, just the basic functional requirements. It needs to be looked at from the user's perspective, with an understanding of potential features, screens, menus and dialogue. • **What fields need to be included on the form?** • **What happens when a user enters the wrong information?** • **What is displayed if an error occurs?**			
1	A User via a form will request to resolve a PID. The form should have the following fields. • an input for the PID • a checkbox if the user wants to resolve the PID or to see the metadata of the PID	H	
2	A User via a form will request to resolve a PID. If the user types a PID that is not supported the user will get an error message, e.g., "This PID type is not supported"	H	
3	A Service Provider will request to be part of the Metaresolver. A form will ask him about the following: • NAME of the provider • REGEX for PID name structure • RESOLUTION_ENDPOINT where the PIDs can be resolved • METADATA_ENDPOINT where the PID metadata can be queried	H	
4	Machine can use the MetaResolver via the API. The developer can start using the service via the main url. This redirects directly to the source.	H	
5	Machine can use the Metaresolver. The developer can start using the service via the main url with the metadata . This redirects directly to the source metadata.	H	
6	Documentation of the API to be used on command line	M	
7	A Service Provider will update its information via a form. Therefore an authorisation request in the UI has to be performed	M	
8	A Service Provider can add • ADDITIONAL_QUERY_ENDPOINTS • SCHEMA the metadata followed	L	

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

Service specifications			
#	Short description	Priority	Reference

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Service specifications			
	API Endpoints names provided in the table below are examples and will be refined during the implementation phase		
1	An Single-Page-Application (SPA) User Interface (UI) with a search box and 2 tick boxes defining either the redirection to the resource, or the display of the metadata.	H	
2	The UI should have a form to support to new Service Providers.	M	
3	The UI should inform the users about the PID Service Providers it supports.	L	
4	The MetaResolver offers the main url pattern SERVICE_URL/REQUESTED_PID and redirects directly to the source.	H	
5	The MetaResolver offers the main url pattern for metadata request SERVICE_URL/metadata/REQUESTED_PID and redirects directly to the source metadata.	H	
6	The MetaResolver should understand the different types of PID based on the syntax (the first characters of the string) or the resolver address in the HTTP URI representation of the PID • 10. for DOIs (as specified in ISO 26324:2022), resolver address (https://doi.org or http://dx.doi.org) for DOIs as specified in ISO 26324:2022, which can begin with any character • 21. for EPIC Handles, resolver address for other Handles, which can begin with any character • urn:nid: for URN (e.g. URN:NBN: or URN:DOI:). Some URN namespaces have sub-divisions (e.g. URN:NBN:de, URN:NBN:fi) which have an impact on resolution • orcid: 16-digit number that is compatible with the ISO Standard (ISO 27729), also known as the International Standard Name Identifier (ISNI). ORCIDs are usually expressed as HTTP URIs (e.g. https://orcid.org/0000-0003-1067-5020) • ISNI: ISNIs are usually expressed as HTTP URIs (e.g. https://isni.org/isni/0000000072366205). ISNI standard specifies also human-readable format (e.g. ISNI 1422 4586 3573 0476) which is not actionable and rarely used in practice. • "swh": for Software heritage software (TODO: SVEN) • Research Activity Identifier (RAiD, ISO 23527) has not been implemented yet. The ISO standard is under development (https://www.iso.org/standard/75931.html), but it is expected to be published in the near future. According to the draft standard, RAiDs may be Handles, DOIs or other identifiers assigned by the RAiD Registration Authority. Human readable format is similar to ISNI (e.g. RAID 123456/654321. HTTP URI representation of Handle and DOI RAiDs will use a dedicated RAiD proxy	H	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Service specifications			

	address (e.g. https://raid _resolver.org/123456/654321). In this HTTP URI format it will be possible to separate RAiDs from other Handles and DOIs.
	• FDO (TODO: Sven)
	• for unknown types:
	  ○ API REQUEST to figure out which PID system is responsible for an PID with unknown naming system
	  ○ else error		

7	The MetaResolver has an api . The description of the API should follow the OPENApi 3.0 specification.	H	
8	The MetaResolver has a list of PID providers that it supports.	H	
9	The MetaResolver supports the Handle Service Provider with the PID resolution service and the metadata and query metadata endpoints	H	
10	The MetaResolver supports the DOIs Service Provider with the PID resolution service and the metadata and query metadata endpoints	H	
11	The MetaResolver supports the URNs Service Provider with the PID resolution service and the metadata and query metadata endpoints	M	
12	The MetaResolver supports the ORCID Service Provider with the PID resolution service and the metadata and query metadata endpoints	M	
13	The MetaResolver supports the Software heritage software Service Provider with the PID resolution service and the metadata and query metadata endpoints	H	
14	**PROVIDER** The MetaResolver support the integration of a new provider via **POST** /integrations/register/ • name: the name of the provider • regex for PID name structure • schema: a schema the metadata follow • resolution_endpoint: the endpoint where the PID can be resolved • metadata_endpoint: the endpoint where the PID metadata exist • query_metadata_endpoint: multiple endpoints for query An email will be send to the Administrators of the service to inform them for the new request. Description: At a registration step a new provider can register a new PID. The require . Dependency: It requires authentication and authorization mechasism.	H	
15	**ADMIN:** The MetaResolver supports the addition of a new provider. This requires code changes. The service is modular and via configuration files and code extensions	H	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Service specifications			
	we can support new Service Providers. This is a process that is triggered by the register action above and is validated by the admin user. The data kept for each PID Service Provider in the configuration files are the following: • name: the name of the provider • regex for PID name structure • schema: a schema the metadata follow • resolution_endpoint: the endpoint where the PID can be resolved • metadata_endpoint: the endpoint where the PID metadata exist • query_metadata_endpoint: multiple endpoints for query		
16	The MetaResolver get the list of enabled providers **GET** /api/v1/providers parameters: status: • all : enabled, registered Description: This endpoint get the list of enabled providers and of course of the registered ones.	H	
17	The MetaResolver get the details of a provider **GET** /api/v1/providers/[PROVIDER_NAME]	H	
18	The MetaResolver via a predefined list of Service Providers would understand the type of provider based on the PID send by the user. **POST** /api/v1/pid/{pid-requested}/provider PID: the users PID request result • type:ex. DOI • schema: https://www.test.com/datatype • resolution_endpoint: the endpoint where the PID can be resolved • metadata_endpoint: the endpoint where the PID metadata exist	M	
19	The MetaResolver will reply with the metadata when the user requests the metadata of a PID **GET** /api/v1/pid/{pid-requested}/metadata	M	
20	The MetaResolver get will redirect to the resource **GET** /api/v1/pid/{pid-requested} The service via this endpoint will respond with a redirect url to the correct resource URL.	H	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Service specifications			
21	The API should return an api endpoint **GET** /api/v1/version Description: Return the version of the service.	L	
22	The API should return an api endpoint **GET** /api/v1/healthcheck Description: Check the api health for monitoring reasons.	L	
23	The API should support pagination in all api endpoints (if required)	L	

24	The API should support a common way of error reporting
	• error
	  ○ type: 4xx, 5xxx (authentication, server)
	  ○ name: a descriptive title of the error
	  ○ description: a description for the user
		H	

25	The API should accept JSON	H	
26	The API should respond using JSON	H	
27	The API should additionally accept XML	L	
28	The API should additionally respond using XML	L	

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

Operational specifications			
#	Short description	Priority	Reference
1	<Provide a short description of operational specifications>	[H|M|L]	<JIRA issue number> or <URL to external reference>
1	The UI page should load in **3 seconds** with the total of < 200 number of simultaneous users.	H	
2	The API should respond in the same time as the resolution service. It should add as little as possible latency to the existing latency introduced by the resolution services (ex. DOI, handle). This will be needs a more detailed profiling. A profiling based on the different services.	H	

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

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	PID Graph resolves Identifiers using MetaResolver not yet known by the PID Graph service	H	<JIRA issue number> or <URL to external reference>
2	AAI infrastructure to support the authentication , authorization of PID Providers for registration and updating of provider information	M	
3	EOSC Monitoring. The Metaresolver via an api healthcheck checkpoint will provider its status for the EOSC Monitoring service.	M	

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

7.1.3 External references

External references		
#	Short description	Reference
1	Handle System	http://www.handle.net
2	Requirement Analysis	https://docs.google.com/document/d/1oD75aCkKp7LB-9qD7WZ_Ib83lVdOwNYTH2ClDdVfBOY/edit#heading=h.zhu9yjagh1na
3	DOIP	Digital object interface protocol https://www.dona.net/specsandsoftware
4	DO-IRP (Revised Handle specification published 2022. Replaces RFCs 3650-3652, and may become an ITU standard in the future)	Digital object identifier resolution protocol https://www.dona.net/specsandsoftware
5	DOI Specification	ISO 26324:2022 Digital object identifier system
6	URN syntax	https://www c-editor.org/rfc/rfc8141.html
7	URN:NBN general description	https://www.rfc-editor.org/rfc/rfc8458.html
8	URN:NBN namespace registration	https://www.iana.org/assignments/urn-formal/nbn
9	All URN namespaces (n =80)	https://www.iana.org/assignments/urn-namespaces
10	FDO Specifications	Version 1.2 ; FDO Forum Proposed Recommendation 19. November 2022

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

External references		
		https://docs.google.com/document/d/1GAj-1owAAPDF7hVis2dYPPCOHiPHrIXIo4j-S5AdEFI/edit
11	Meta Resolver from the ARK community	http://n2t-dev.n2t.net/

7.2 T5.3 PID Kernel Information Profiles - Technical Specifications

This Task focus on the definition and implementation of Kernel Information Profiles and is not a EOSC Core Component. Thus the technical specifications below differ from the software development done in the other tasks. The requirements derived and the communities involved are described in: KIP Specification (Sharepoint). The tables will be filled if suitable.

The Kernel Information Profiles need to registered in a suitable software solution. Thus this tasks is linked with WP4.

Component	Not a Core Component
Category	DTR
Contact person	Sven Bingert (GWDG)
Email address	Sven Bingert
Contributors	
Version	1.0
Data	

Overview
The PID Kernel Information Profile describes the set of kernel attributes that are being selected by a repository to describe the digital object. The profile is dependent on the type of object and the needs of the community. Only openly declared attributes with widely agreed semantic types will provide machine actionability.

Objectives	
#	Short description
1	Derive Governance Model for Kernel Information Profiles
2	Implement a registry for Kernel Information Profiles

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Out of Scope	
#	Short description
1	Registration of Types/Vocabularies is not part of this taks.

7.2.1 Requirements

User stories		
#	Description of the user story	Reference
1	As a <Community> I want to be able to <register a kernel information profile> so that <the kernel information is standardized>	
2	As a <user> I want to be able to <search for existing registered types> so that <a kernel information profile can be compiled>	
3	As a <user> I want to be able to <validate a metadata set against a kernel information profile> so that <I can verify the completeness and accuracy of the given metadata>	
4	As a <user> I want to be able to <list PID service providers and the provided kernel information> so that <I learn how to query for additional information>	

7.2.2 Specifications

Architectural design
<Provide a high-level architectural design diagram of the component/service>

7.2.3 External references

8 WP6 –Services and tools to archive, reference, describe and cite research software

8.1 Introduction

In 2021, the Scholarly Infrastructures of Research Software (SIRS) report[1], was published including a set of recommendations to allow EOSC to include software, in the scholarly ecosystem, next to publications and

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

data. The SIRS report was built upon a survey and documentation of a representative panel of notable operational infrastructures across Europe, comparing their scopes and approaches. A subset of these Research Software infrastructures joined forces to turn the SIRS recommendations into reality through the FAIRCORE4EOSC project.

One of the WP6 objectives is to develop tools and services for archival, reference, description and citation of research software artifacts, by implementing the key recommendations of the EOSC SIRS report to interconnect scholarly repositories, publishers and aggregators with the Software Heritage universal source code archive, using the CodeMeta standard, and the Software Heritage intrinsic identifiers (SWHID).

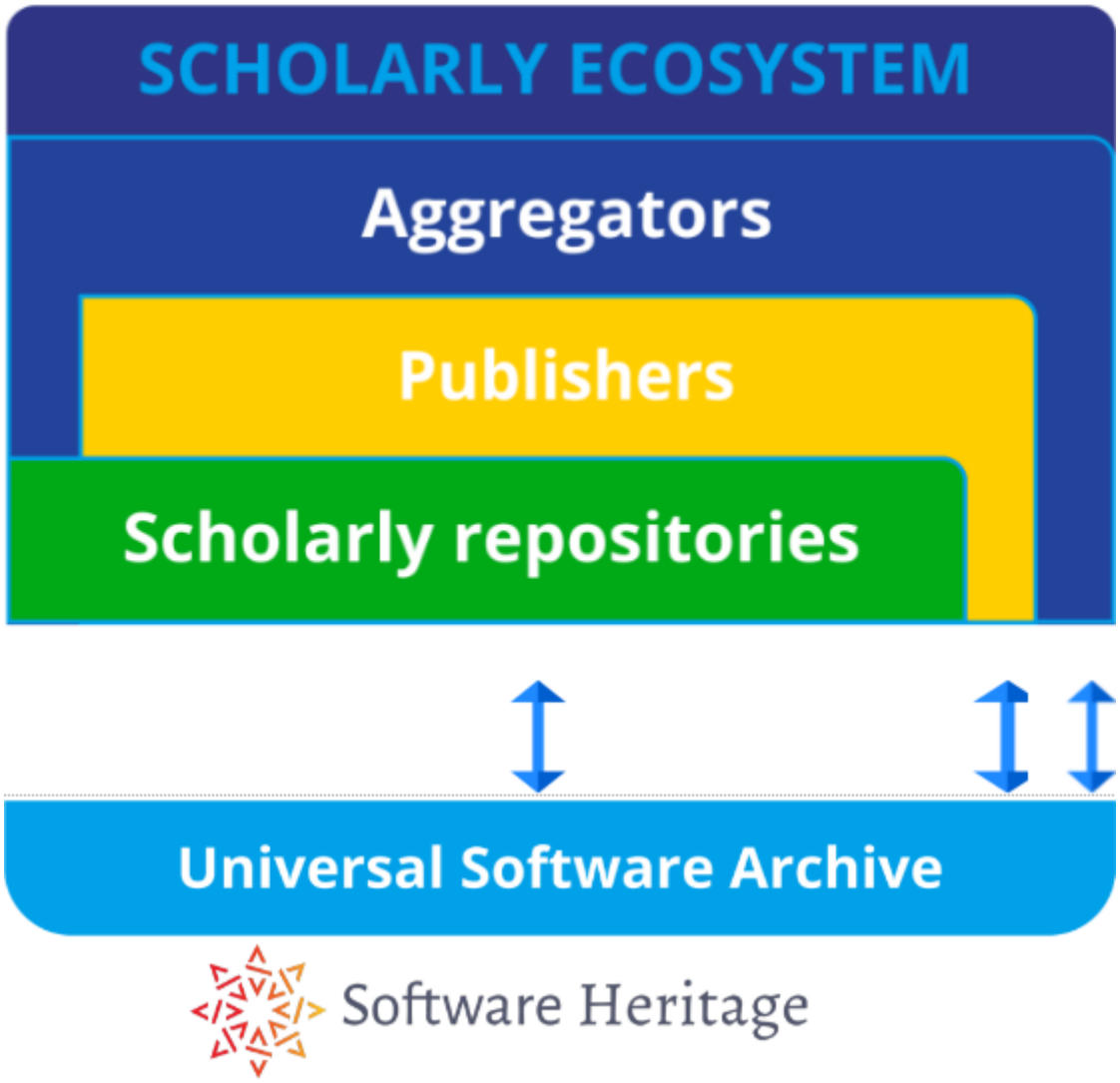

Figure 24 – Expanded diagram of a architecture of interconnected scholarly infrastructures supporting archival, reference, description and citation of (research) software source code from SIRS report[1]

In the following subsections you will find a snapshot of the specifications documents created by the FAIRCORE4EOSC partners for the research software infrastructures that were identified in the project proposal. Three teams were assigned three tasks, each team developed for each infrastructure a specifications document to follow, the tasks are divided by infrastructure type:

- T6.1 API and Connectors Between Scholarly Repositories and Software Heritage [M1-M36] Lead: CERN, Participants: KNAW-DANS, GWDG, INRIA, FONDATION INRIA, DATACITE

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

- T6.2 API and Connectors Between Open Access Publishers and Software Heritage [M1-M36] Lead: LZI Participants: INRIA, FONDATION INRIA
- T6.3 API and Connectors Between Aggregators and Software Heritage [M1-M36] Lead: FIZ Participants: INRIA, FONDATION INRIA, DATACITE, OPENAIRE, LZI

The WP6 lead has divided the work into subcomponents and each subcomponent was represented by a task lead or a task participant.

The RSAC subcomponents specification documents by infrastructure type:

Scholarly repositories:

- InvenioRDM - SWH SPECS
- DataVerse - SWH SPECS

Publishers:

- Dagstuhl - SWH SPECS
- Episciences - SWH SPECS

Aggregators:

- SwMath - SWH SPECS
- OpenAire - SWH SPECS

The specifications documents are a blueprint for the development of the APIs and connectors from various scholarly infrastructures to support research software.

Building APIs and connectors in different research software infrastructures for a more connected scholarly ecosystem is a significant step toward the realization of the SIRS report and its recommendations. In the SIRS report, long term preservation, describing research software with machine actionable metadata standards and the usage of the SWHID as persistent software artifact identification are part of the specifications submitted by the partners in the FAIRCORE4EOSC WP6.

8.2 RSAC component for scholarly repositories

8.2.1 InvenioRDM - SWH SPECS

Component	InvenioRDM software source code & metadata deposit
Category	RSAC
Contact person	Lars Holm Nielsen
Contributors	
Version	2.0
Data	2023-01-20

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Overview
Integration of InvenioRDM (Scholarly Repository) with Software Heritage Overall, the purpose is to develop tools and services for archival, reference, description and citation of research software artifacts, by implementing the key recommendations of the EOSC SIRS report to interconnect scholarly repositories with the Software Heritage universal source code archive, using the CodeMeta standard, and the Software Heritage intrinsic identifiers (SWHID)

Objectives	
#	Short description
1	Deposit in Software Heritage research software artifacts uploaded into InvenioRDM repositories, obtaining the corresponding SWHID
2	Expose SWHID in the artifact record maintained by InvenioRDM
3	Enable InvenioRDM to deposit and/or retrieve (curated) metadata of software artifacts from Software Heritage
4	Export citation information in one or more of the common open citation formats (BibLaTeX, CSL, codemeta.json)
5	Integrate and showcase the above into Zenodo

Out of Scope	
#	Short description
1	Removal of content Software source code deposited in the repository may be subject to a take-down request or other requirements requiring the removal of the deposited material (e.g. legal reasons). The removal of content is assumed to be done independently for each service according to their policies. Thus there will be no automatic notification between the scholarly repository and the universal source code archive about removed content. The moment at which the content was transferred between the two services, the content was assumed to be public. This is consistent with how removal of content that might have been harvested by other repositories are dealt with.
2	Mixed deposit workflows The deposit workflows outlined in UC1, UC2 and UC3 can be mixed up. For instance, - V1: Deposit as a bundle (UC1) - V2: Deposit by enabling automatic deposit for a new release (UC2) - V3: Deposit by registering an already archived bundle (UC3) Each of the three versions leads to a new record in the scholarly repository which is linked to the other versions. The requirements specifically do not deal with how to transition a user between the different workflows.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Out of Scope
Similarly, a user may use the UC1 workflow to deposit multiple versions, and later figure out that an old release also has to be published. Another example, is a user depositing software source code and later decides to change it to restricted (with or without moderation). We specifically exclude dealing with these cases, and leave it up to the scholarly repository how it wants to deal with it according to its existing policies.

8.2.1.1 Requirements

Overall, the requirements focus on researchers interacting with scholarly repositories, and how the repository interacts with the universal source code archive.
The user stories refer to the following roles:

- Researcher: An individual researcher or full research team

User Stories				
#	Short description	Priority	Feasibility	Reference
A1	As a researcher, I want to deposit bundle of source code into repository, so that it will be archived for the long term in the universal source code archive, SWH	H	4	
A2	As a researcher, I want to register already SWH-archived source code in a repository], so that I obtain an extrinsic PID (e.g. DOI) for my software [and more easily integrate cite it in with journal authoring system	L	3	
A3	As a researcher, I want to deposit a new release of source code into a repository, so that it will be archived for the long term in the universal source code archive, SWH	M	3	
R1	As a researcher, I want to retrieve a persistent extrinsic and intrinsic identifier from the repository, so that it will reference the record and source code artifacts	H	5	
D1	As a researcher, I want to describe software using software specific metadata, so that the software is findable and reusable	H	4	
D2	As a researcher, I want to retrieve metadata export for deposit in repository, so that the metadata can be used in other workflows	M	4	
D3	As a researcher, I want to update metadata of software in repository, so that the metadata can be as accurate and complete as possible	L	4	
C1	As a researcher, I want to retrieve a citation export format from the repository, so that the software will be cited with correct attribution	H	4	

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

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

- Feasibility: marking between 1 - 5, 5 is easy to implement and 1 is very difficult

Functional requirements			
#	Short description	Priority	Reference
UC1	Deposit Bundle Through an Scholarly Repository • A researcher explicitly deposits a software bundle (.tar.gz, .zip, etc.) and associated metadata into a scholarly repository. • Optional: The scholarly repository may implement a moderation mechanism to ensure a certain level of quality of the deposit (deduplication, affiliations of the team members, coherence of the metadata, etc.). • Once accepted, the bundle and metadata are archived in SWH, either immediately, or at the end of the optional embargo period. *Notes* • User stories A1 and A3 are considered as almost identical as both will produce a record in the scholarly repository (i.e. A3 does not modify an existing record, but creates a new record instead so that versions can be differentiated). • InvenioRDM can be configured with or without moderation support. The trigger to deposit the bundle in SWH is thus the public availability of a published record (i.e. either immediately or after an embargo period). See annexe A: Sequence diagram	H	User stories: A1, A3, R1, D1
UC2	Automated deposit of New Releases into Scholarly Repository and SWH (Manual and Automated) Automated/manual deposit of new releases from a forge into a scholarly repository, as e.g. implemented in the Zenodo-GitHub integration, is an extension to depositing a bundle into scholarly repository. • The scholarly repository is notified about a new software release either automatically by the forge, or manually by a researcher. • The scholarly repository automatically extracts the deposit bundle and associated metadata from the forge. • Continues as "UC1 Deposit Bundle Through an Scholarly Repository"	M	User stories: A3
UC3	Registering Already Archived Software in an Scholarly Repository A researcher may need to register an artifact that has been already archived in SWH in a scholarly repository. To avoid duplication of work, machine readable metadata contained in designated files in the source code should be used to prefill	L	User stories: A2

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
	the metadata deposit form.		
UC4	Updating Existing Metadata in an Scholarly Repository A research team may also need to update the metadata registered in a scholarly repository about an already archived software artifact. Previous versions of the metadata record should be recorded together with information on who changed what, when, and why. The changes should be made public to the extent possible by the repository's policies. *Notes* • Previous versions of the metadata record should be recorded and available, together with information on who changed what, when, and why.	L	User stories: D3

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

Non-functional requirements			
#	Short description	Priority	Reference
1	Personal data compliance The scholarly repository and universal source code archive are owned by different legal entities, and thus any exchange of personal data must be clearly reflected in the scholarly repository's privacy policy. Overall, exchange of any personal data should be avoided, and if done must comply with legal requirements.	H	
2	Metadata formats must support interoperability and software properties exchange The scholarly repository and universal source code archive exchange metadata in both directions. The repository sends metadata to SWH, and the repository fetches metadata from SWH. The metadata formats used for the exchange must enable the software specific metadata to be exchanged as well as enable the metadata formats to be used for citation generation. Likewise the metadata may be registered/harvested by other discovery systems.	H	User stories: D1, D2, C1
3	Reliable exchange of deposits between repository and SWH The scholarly repository is responsible for ensuring software records are sent to the universal source code archive and must be able to determine if a software record has been published to SWH.	M	

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

8.2.1.2 Specifications

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

Sequence diagram UC1

Deposit Bundle Through an InvenioRDM repository

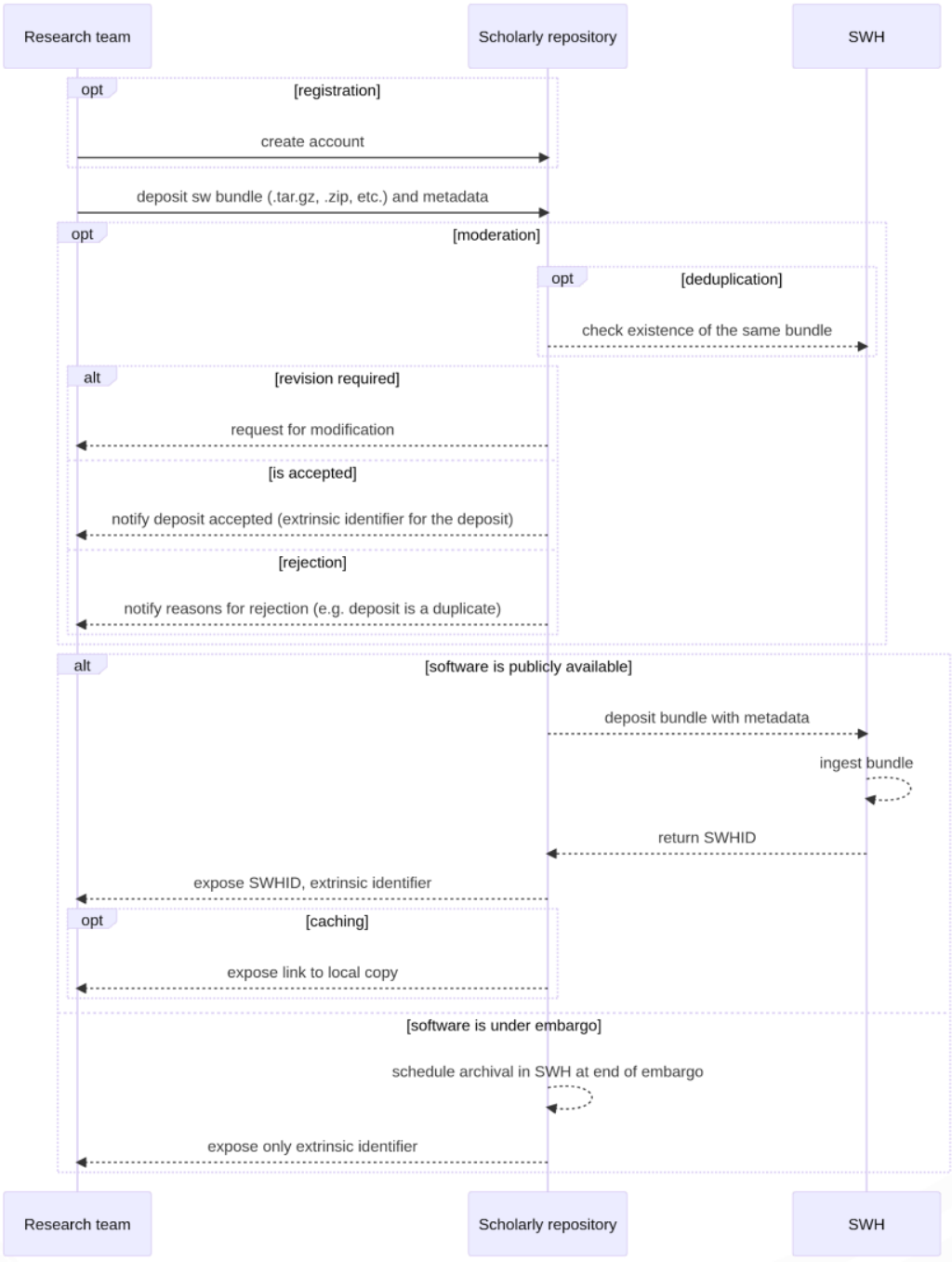

Figure 25

Sequence diagrams UC2

Automated deposit of New Releases into InvenioRDM and SWH (Manual and Automated)

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

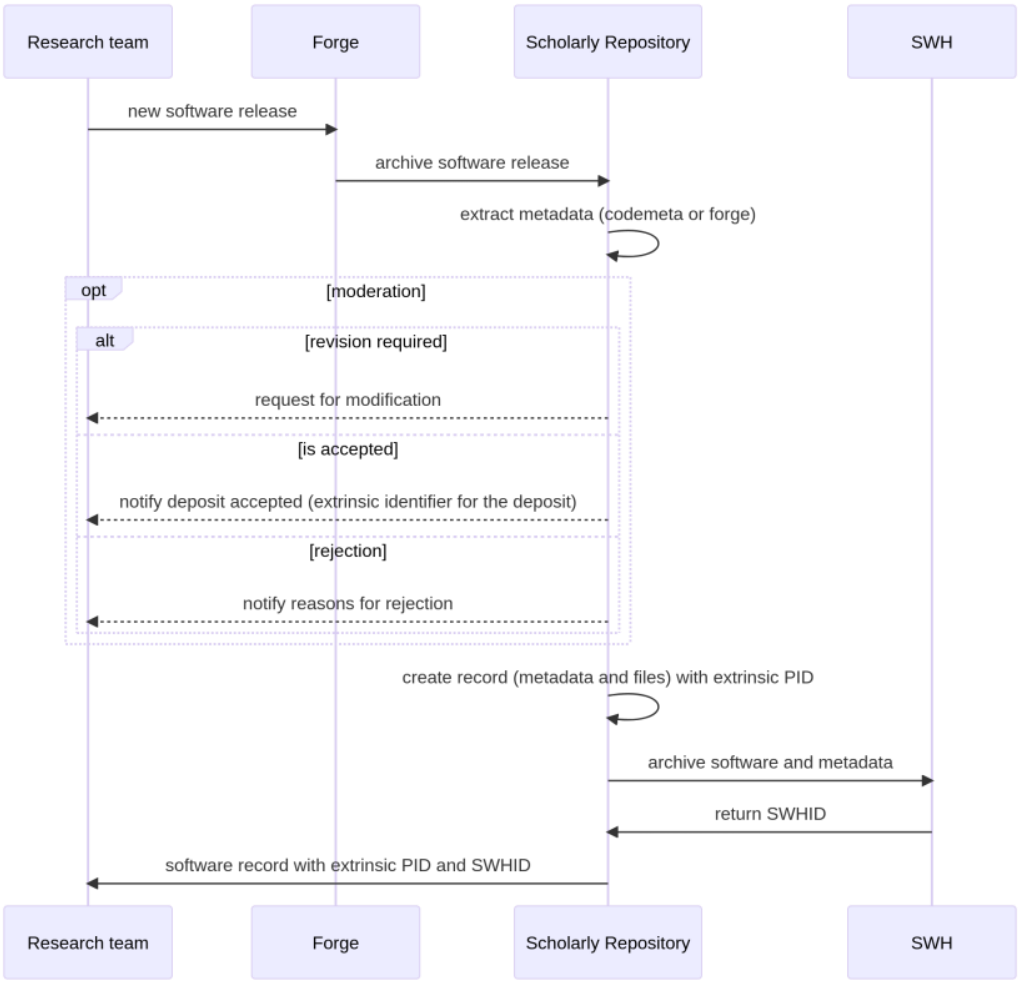

Figure 26

Sequence diagrams UC3

UC3 Registering Already Archived Software in an InvenioRDM repository

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

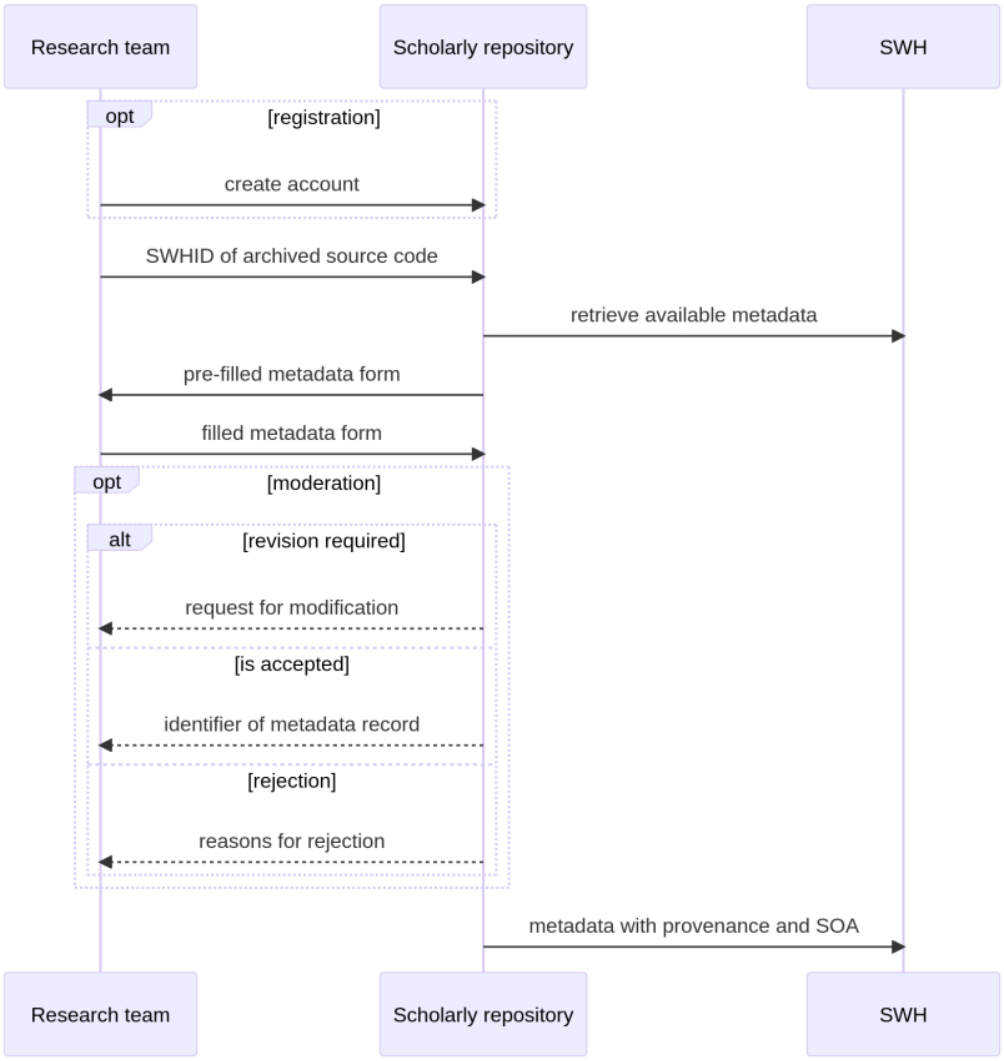

Figure 27

Sequence diagrams UC4

UC4 Updating Existing Metadata in an InvenioRDM repository

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

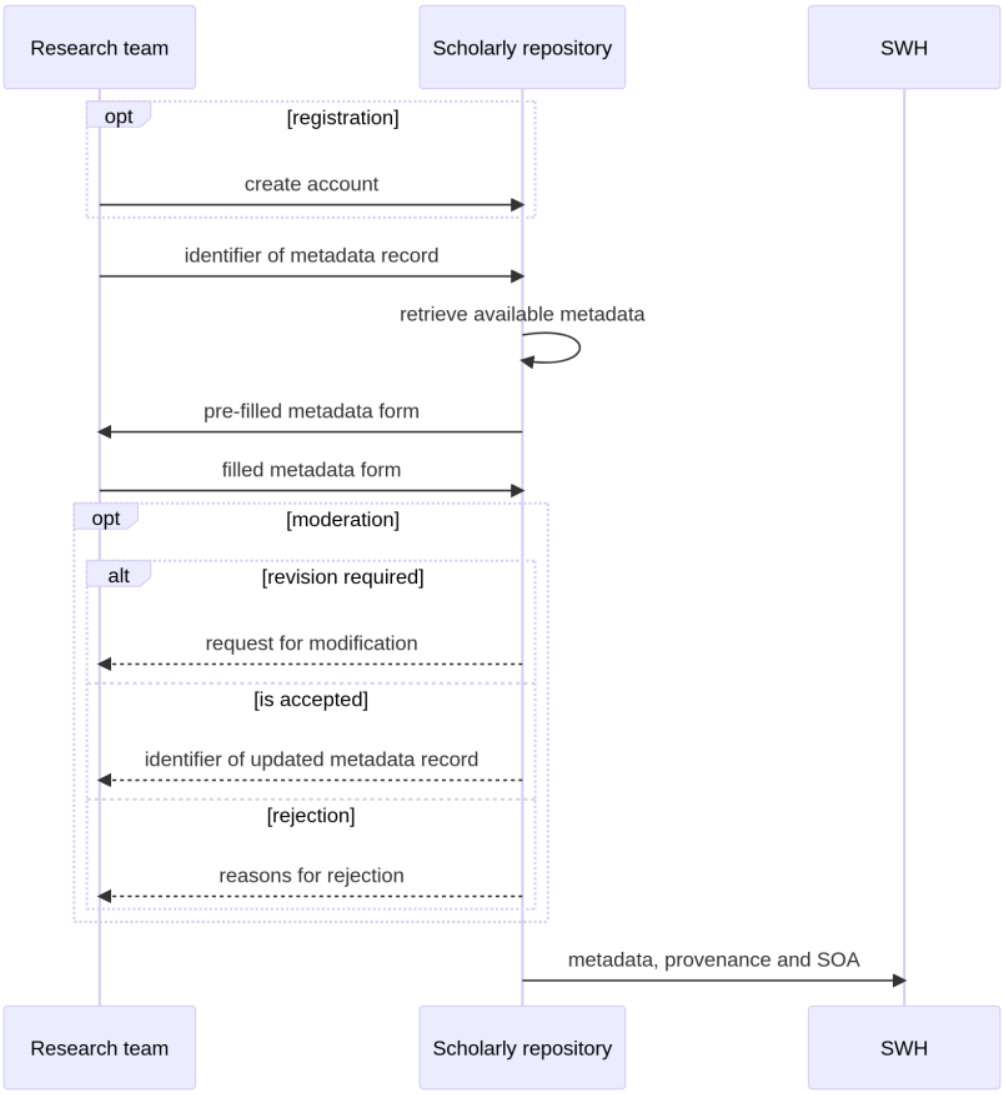

Figure 28

Functional specifications

#	Short description	Priority	Reference
1	Feature flag, configuration and documentationConfiguration The Software Heritage integration in InvenioRDM will need to be an optional add-on as not all InvenioRDM instances may want to enable the integration, nor is an InvenioRDM instance even guaranteed to host software source	H	

Functional specifications			
	code. Software Heritage further requires authenticated access, thus as minimal a system administrator needs to provide their SWH credentials to the InvenioRDM application. Documentation We will ship InvenioRDM with Software Heritage support directly integrated, but users will need to enable the integration similar to how users need to enable the DataCite integration for registering DOIs.		
2	Marking resource types as software source code A scholarly repository may host many different resource types, and only software source code should be sent to SWH. Tag resource types InvenioRDM needs to enable a system administrator to tag which deposit resource types that should be sent to SWH. Only public records Further, only public records with public files are sent to SWH.	H	
3	Software-specific metadata Codemeta for interoperability The [Codemeta standard](https://codemeta.github.io/terms/) based on Schema.org is supported by both SWH, Zenodo and InvenioRDM as a common metadata exchange format. Further, InvenioRDM registers DOI via DataCite and thus provided DataCite metadata format as an exchange format for repositories and aggregators Supported terms	H	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	The following Codemeta terms have existing fields in InvenioRDM or requires only minor adaptions to be supported: - `downloadUrl` - `fileSize` - `releaseNotes` - `author` - `citation` - `contributor` - `copyrightHolder` - `copyrightYear` - `creator` - `dateCreated` - `dateModified` - `datePublished` - `editor` - `fileFormat` - `funder` - `keywords` - `license` - `producer` - `provider` - `publisher` - `sponsor` - `version` - `isAccessibleForFree` - `isPartOf` - `hasPart` - `position` - `description` - `identifier` - `name` - `sameAs` - `url` - `relatedLink` - `givenName` - `familyName` - `affiliation` - `identifier` - `name` - `maintainer` - `embargoDate` - `funding` - `referencePublication` - `readme` The following Codemeta terms requires		

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	new fields: - `codeRepository` - `programmingLanguage` - `runtimePlatform` - `developmentStatus` - `operatingSystem` The following Codemeta terms will not be supported: - `targetProduct` - `applicationCategory` - `applicationSubCategory` - `installUrl` - `memoryRequirements` - `permissions` - `processorRequirements` - `softwareHelp` - `softwareRequirements` - `softwareVersion` - `storageRequirements` - `supportingData` - `email` - `address` - `softwareSuggestions` - `contIntegration` - `buildInstructions` - `issueTracker` SWHID InvenioRDM must also be able to store a Software Heritage Identifier (SWHID). The SWHID will be a fully system managed property that users will not be able to edit. The identifier will be managed using InvenioRDM's internal persistent identifier management system which further enforces that we cannot store duplicate SWHID's.		
4	Import software specific vocabularies SPDX is the default license vocabulary. Users may describe their record using software-specific metadata (as described above). The following fields will be supported by vocabularies that help drive auto-completion, search	M	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	filtering and display: - `programmingLanguage` - TODO_TODO _(identify suitable vocabulary) - `runtimePlatform` - TODO (identify suitable vocabulary) - `developmentStatus` - repostatus.org - `operatingSystem` - TODO (identify suitable vocabulary)		
5	Software metadata on record landing page The record landing page for a repository record should expose: - The Software Heritage Identifier (SWHID) - including the permalink to the full SWHID access the code in it's context - Software specific metadata.	H	
6	Deposit form software fields The deposit form (i.e. form used for depositing records and/or updating metadata) will add support for the new fields defined above. The form will group software-specific fields.	H	
7	Deposit form upload SWH ID The "Files"-section of the deposit form will add support for uploading/registering files from multiple sources include "from Software Heritage". Choosing "from Software Heritage", will enable users to provide a SWH ID of software source code already registered in SWH. Also, this feature can detect if a SWH is already deposited in the repository		
8	Metadata formats (import/export) The following metadata formats will be adapted to support software-specific metadata according to the supported terms: - BibTeX - CSL (Citation Style Language)	M	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	- CFF (Citation File Format) - Codemeta - DataCite The formats will be supported as: - Export option on record landing pages - REST API metadata formats		
9	Automatic extraction of metadata The automatic extraction of metadata from software source code will happen in two cases: - via the GitHub integration - via the depositing of already SWH-registered software In both cases, the automatic metadata extraction services to prepopulate the deposit form and possibly automatically publish the record. In both cases metadata from Codemeta or CFF will be supported as the source of the metadata being imported.	L	
10	Deposit from repository to universal source code archive We deposit in SWH published records that must fulfill all of the following conditions: - Resource types tagged as software source code - Public accessible metadata and files - Record must have one or more associated files - Max size of all files does not exceed 100 Mib (SWH deposit limit) A new version of a record in InvenioRDM is technically a new record, and thus must comply with above conditions. Metadata updates to an existing record is allowed after the record is published.	H	https://docs.softwareheritage.org/devel/swh-deposit/api/user-manual.html#push-a-deposit https://docs.softwareheritage.org/devel/swh-deposit/api/index.html

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	Event triggers The moments when software must be pushed to SWH is when: - A draft record is published. - A record metadata update is published. - An embargo period expires. Multiple scenarios exist where a user changes resource type, access rights or embargo period. These scenarios are explicitly excluded from the scope (see 2.4) Background publishing to SWH A record is published to SWH when one of the above events are triggered and the record fulfill all conditions. The publishing to SWH uses the [SWH SWORD deposit API]() to send a compressed file and metadata file. Files The SWORD deposit requires a single compressed file to be sent. In case a record has a single compressed file (zip, tar, tar.gz, tar.bz2 or tar.lzma) this file will be sent directly. In case a record has multiple files, they will be compressed and sent as a single file. Technically, SWORD allows multiple files, but it is best not to use this feature. A zip within a zip is forbidden with the SWH deposit even though the verification on InvenioRDM side would be hard to enforce. To do: specify error handling, when SWH API may fail ingestion, and that the repository is able to inform the user that the ingestion failed. State management Internally, InvenioRDM will keep track of records sent to SWH and their submission state. The following state are		

Functional specifications			
	defined (similar to SWH's [deposit statuses]) - Created - Event triggered and record matches - In progress - Sending started (SWH partial) - Sent - Record was sent to SWH (SWH deposited, verified and loading states) - Done - Record succeeded ingestion in SWH (SWH done state) - Failed - Record failed ingestion in SWH (SWH rejected, expired or failed states)		
11	Feature flag, configuration and documentation Configuration The Software Heritage integration in InvenioRDM will need to be an optional add-on as not all InvenioRDM instances may want to enable the integration, nor is an InvenioRDM instance even guaranteed to host software source code. Software Heritage further requires authenticated access, thus as minimal a system administrator needs to provide their SWH credentials to the InvenioRDM application. Documentation We will ship InvenioRDM with Software Heritage support directly integrated, but users will need to enable the integration similar to how users need to enable the [DataCite integration] for registering DOIs.		[DataCite integration]- https://inveniordm.docs.cern.ch/customize/dois/
12	Deposit from repository to universal source code archive We deposit in SWH published records that must fulfill all of the following conditions: - Resource types tagged as software source code - Public accessible metadata and files	H	[SWH SWORD deposit API](https://docs.softwareheritage.org/devel/swh-deposit/api/index.html) [deposit statuses](https://docs.softwareheritage.org/devel/swh-deposit/api/user-manual.html#push-a-deposit)

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	- Record must have one or more associated files - Max size of all files does not exceed 100 Mib (SWH deposit limit) A new version of a record in InvenioRDM is technically a new record, and thus must comply with above conditions. Metadata updates to an existing record is allowed after the record is published. Event triggers The moments when software must be pushed to SWH is when: - A draft record is published. - A record metadata update is published. - An embargo period expires. Multiple scenarios exist where a user changes resource type, access rights or embargo period. These scenarios are explicitly excluded from the scope (see 2.4) Background publishing to SWH A record is published to SWH when one of the above events are triggered and the record fulfill all conditions. The publishing to SWH uses the [SWH SWORD deposit API] to send a compressed file and metadata file. Files The SWORD deposit requires a single compressed file to be sent. In case a record has a single compressed file (zip, tar, tar.gz, tar.bz2 or tar.lzma) this file will be sent directly. In case a record has multiple files, they will be compressed and sent as a single file_technically, SWORD allows multiple files, but it makes everything more complicated, so		

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	it is indeed best not to use this feature.._Should we specify that a zip within a zip is forbidden ?This would be hard to enforce. Example: User greates a git repository on GitHub, archives in Zenodo, end result is a zip with zip inside will be sent to SWH. I'd rather see this as part error handling, that in some cases the SWH API may fail ingestion, and that the repository is able to inform the user that the ingestion failed. State management Internally, InvenioRDM will keep track of records sent to SWH and their submission state. The following state are defined (similar to SWH's) - Created - Event triggered and record matches - In progress - Sending started (SWH partial) - Sent - Record was sent to SWH (SWH deposited, verified and loading states) - Done - Record succeeded ingestion in SWH (SWH done state) - Failed - Record failed ingestion in SWH (SWH rejected, expired or failed states)		
13	Feature flag, configuration and documentation Configuration The Software Heritage integration in InvenioRDM will need to be an optional add-on as not all InvenioRDM instances may want to enable the integration, nor is an InvenioRDM instance even guaranteed to host software source code. Software Heritage further requires authenticated access, thus as minimal a system administrator needs to provide their SWH credentials to the InvenioRDM application.		[DataCite integration](https://inveniordm.docs.cern.ch/customize/dois/) for registering DOIs.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
	Documentation We will ship InvenioRDM with Software Heritage support directly integrated, but users will need to enable the integration similar to how users need to enable the [DataCite integration] for registering DOIs.		

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

8.2.1.3 External references

External references		
#	Short description	Reference
1	InvenioRDM Infrastructure Architecture	https://inveniordm.docs.cern.ch/develop/architecture/infrastructure/
2	InvenioRDM Records Data Model	https://inveniordm.docs.cern.ch/develop/architecture/records/
3	SWORDv2	http://swordapp.github.io/SWORDv2-Profile/SWORDProfile.html#protocoloperations_creatingresource_entry
4	Software Heritage deposit documentation	https://docs.softwareheritage.org/devel/swh-deposit/api/index.html#deposit-api

8.2.2 DataVerse - SWH SPECS

Component	DataVerse connector and API to Software Heritage
Category	**RSAC**
Contact person	Wim Hugo
Contributors	Wilko Steinhoff
Version	2.0
Date	20.01.2023

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Overview
Repository software will interact with the services created in FAIRCORE4EOSC in multiple ways, and at multiple points during specific workflows. These workflows are described as individual user stories below. The user stories are specific to repository operations, and not all of the requirements expressed will be satisfied or addressed by FAIRCORE4EOSC components. The user stories are documented as requirements - functional and otherwise - and a high-level architecture/ specifications are provided. In the detailed requirements and specifications, the elements that can be provided by FAIRCORE4EOSC components and services are identified clearly. Specifically, some of the services, as indicated, are provided directly by WP6.

Omissions and Work to be Done
1. Review of the use cases against the work of RDA a. Publishing Data Workflows b. Standardised RDA Use Cases c. Matrix of Use Cases d. FAIR Workflows 2. RDA Repository Terminology Verification (Case Statement)

Objectives	
#	Description
1	Implement workflow steps and interfaces to integrate FC4E services into the main tasks of a repository: 1.1 Manual and automated deposit of research output 1.2 Curation - manual and automated 1.3 Discovery and re-use
2	Provide a High-Level Architecture 2.1 Context - including design considerations 2.2 Containers and Components -indicating the contribution from FC4E services and components
3	Specifications 3.1 Process flows for major use cases 3.2 Links to applicable specifications for APIs and interfaces

Out of Scope	
#	Short description

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Out of Scope	
1	The workflow steps from a repository perspective are reproduced below for sake of completeness, but will not all be in scope for the FC4E project. Those that are not in scope are indicated explicitly

8.2.2.1 Requirements

User stories		
#	Description of the user story	Reference
1	Manual/ automated deposit of research output A user (researcher at a RPO) wants to deposit a collection[1] of research outputs to an appropriate repository for the type of output, and in alignment with institutional and disciplinary practices. The repository should be certified as trustworthy, or, alternatively, provide for some important aspects of trustworthy repositories - such as long-term preservation. Ideally, researchers would want to engage the deposit pipeline from within their own working environments: VREs, institutional or disciplinary websites, scientific workflows and code, etc. On deposit, the researcher would like to receive guidance in respect of the most appropriate repository for each object in the collection, while providing the minimum required metadata (FAIR, reproducibility[2], schema compliance, PID registration compliance, disciplinary norms, ...) only once across all of the different types of research outputs in the collection. In addition, guidance in respect of minimum metadata, appropriate licence selection, and FAIRness of objects in the collection will be very useful. The researcher would want to receive confirmation of deposit and associated PIDs at the earliest opportunity, and be able to see an inventory of published works across different repositories. 1. Typical collection can comprise of one or more of the following: a. Input/ raw dataset(s) b. Derived/ processed dataset(s) c. Code d. Semantic artefacts e. Reports, papers, and other scholarly publications 2. Reproducibility requires additional PID references for a. Instruments b. Methodology c. Samples, Physical Objects, and Specimens d. Computing Platforms	RDA Use Cases RDA Publishing Workflows FAIR Workflows RDA Repository Attributes PresQT

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
2	Curation Tasks Curators require development and integration of new tools to assist with the processing of new and legacy deposits, covering the following broad requirements: 1. Evaluation of FAIR compliance, PID Policy compliance, reproducibility, and alignment with disciplinary norms - ethics, formats, etc. 2. Verification of and linking to controlled vocabularies and registries, and recording PID relations implied or explicitly provided in the metadata. 3. Maintaining provenance and versioning information 4. Updating external semantic artefacts, such as PID and Research Graph services, with information about the deposited object.	Signposting OAI-PMH Linked Data Notifications
3	Enhanced Discovery 3.1 Indexing catalogues with additional facets obtained from external semantic artefacts - primarily but not limited to vocabulary or registry services, and PID/ research graph services. 3.2 Provision of contextual information to end users in respect of repository objects (deposits) - for example compliance or user feedback - from services such as PID and research graphs. 3.3 Embedding graph-based UI components into repository discovery tools to improve utility for the end user.	

User requirements				
These are specific user requirements that can be addressed by FAIRCORE4EOSC components or products				
#	Short description	Priority	Feasibility	Reference
1	An API for the provision of metadata and references to code for long-term preservation in Software Heritage. The code is linked to a dataset and/ or other research objects being deposited. Needs to return a PID (SWHID) for inclusion into PID graphs and provenance information in Dataverse. PresQT as a candidate specification or facade. Note: For the SWHID to be included in the PID Graph, the SWHID needs to be included as a related identifier in DOI metadata	H	4	WP6 Component PresQT
2	An API for retrieving metadata in an agreed schema (CodeMeta) from SoftwareHeritage based on a SWHID, and obtain a citation in one or more standardised formats (specific formats to be confirmed)	H	5	WP6 Component
3	An API for updating PID Graph information based on the LOD references obtained during the deposit process. The source of the LOD contribution needs to be persisted in the target PID Graph.	H	3	

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
4	An API for updating Research Graph information based on the LOD references obtained during the deposit process. The source of the LOD contribution needs to be persisted in the target PID Graph.	H	3	
5	An API for referencing semantic artefacts in a consistent and sustainable way - can be a brokering platform as envisaged in part for FC4E	M	2	
6	An API for querying and accessing information stored in the Research Graph service developed by FC4E	H	3	
7	An API for querying and accessing information stored in the PID Graph service developed by FC4E	H	3	
8	An API for querying and assessing level of PID compliance of the PIDs included into a metadata record.	H	3	
8	An API for transformation of a metadata record - schematically and/ or semantically - by using the metadata crosswalk service envisaged for FC4E	M	2	
9	An API for registering a concept or type, with a PID returned, in a vocabulary or type registry.	M	3	
10	...			

Functional requirements			
#	Short description	Priority	Reference
1	Ingest Workflow 1. Selection of appropriate pipelines, target repositories for collection of research outputs 2. Verification of metadata completeness 3. Assessment of deposited outputs in respect of compliance with multiple sets of criteria a. FAIR b. Reproducibility c. PID Policies d. Format and schema compliance e. Optional format conversions for dissemination f. Optional metadata schema crosswalks g. Ethics clearance and protection of subject's rights (CARE) 4. Licence assessment and evaluation a. Rights to allocate or select a licence	[H|M|L]	<JIRA issue number> or <URL to external reference>

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
	b. Extent of private or sensitive data c. Licence selection and applicability		
2	Long-Term Preservation Workflow 1. Creation of preservation metadata 2. Assessment of deposited outputs in respect of compliance with multiple sets of criteria a. Format and schema compliance b. Optional format conversions for preservation		
3	Curation Workflow - New Deposits		
4	Curation Workflow - Legacy Deposits		
5	Indexing and Faceting Workflow		
6	Contextualisation Workflow		
7	Graph-Based Discovery Assistance Workflow Repository software such as Dataverse needs to be extended to allow externally provided search and discovery UIs to be embedded into the main application. These extensions invoke the standard Dataverse Search API after assisting the user with query formulation in any of a number of ways. 1. Select Graph-Based Search Option as an advanced search method in the Dataverse UI. 2. Navigate a graph of choice (e.g. Research Graph, PID Graph) and formulate a query based on nodes and relations. 3. Translate the query int corresponding search API facets. 4. Execute via standard Dataverse Search API.		Dataverse Search API

8.2.2.2 External references

External references		
#	Short description	Reference
1	Dataverse open soure repository	https://github.com/IQSS/dataverse
2	Software Heritage deposit documentation	https://docs.softwareheritage.org/devel/swh-deposit/api/index.html#deposit-api

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

8.3 RSAC component for Publishers

8.3.1 Dagstuhl - SWH SPECS

Component	Dagstuhl connector and API to Software Heritage
Category	**RSAC**
Contact person	Michael Wagner Ramy-Badr Ahmed
Contributors	Ramy-Badr Ahmed
Version	2.0
Date	20.01.2023

Overview
<Provide a high-level description of the component/service> Develop APIs and connectors between Dagstuhl and Software Heritage to support: • Archival: automate the archival in Software Heritage of the source code of artifacts associated to 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 to publications; export citation information in one or more of the common open citation formats (BibLaTeX, CSL, codemeta.json).

Objectives	
#	Short description
1	Archival: • Ensure and promote publicity of source code of artifacts associated to research articles. • Offer instructions on research code submission to either, a Forge, a Scholary Repository or to Dagstuhl directly as a software bundle. • Automate the archival process in Software Heritage of the source code of artifacts associated to research articles.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Objectives	
	• Build a database of SWHIDs of the associated source code for 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 artifact on research document. • 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 integerity 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 to publications.
4	Citation: • Provide guidelines and best practices regarding citation of software so that software is eventually cited with correct attributions by Reseachers. • Deposit and retrieve the preferred citation information for software associated to 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 decides 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. If a user is not willing to do that from their own initiative, the archive pillar won't be satisfied.

8.3.1.1 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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

- **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
1	**As a** Researcher **I can** submit a non-public source code to Publisher **so that** it will be publicised as Software Origin for the universal source code archive, SWH and added to the 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 non-public 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** reference the record and source code artifacts on 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 artifacts.	Reference: Use Case R1
5	**As a** Publisher **I can** add a reference to the archived software artifact on document **so that** the related software is clearly linked to the text.	Reference: Use Case R2

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
	Summary: Dagstuhl adds reference on research articles to clearly map the software artifact 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 integerity 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 metadata export for a software deposit on a scholarly repository from Dagstuhl.	Cite: Use Case C1
10	**As a** Publisher **I can** provide guidelines and best practises regarding citation of software, **so that** Researchers are to correctly cite their software.. **Summary:** Dagstuhl provides best practices and guidelines addressing citation of software. Dagstuhl scans for potential missing references and/or citations to ensure correct citations by Researchers.	Cite: Use Case C2

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements (user = Reasearchers, 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 Automating new archivals, updating swh archived version of sw (if necessary), updating swh metadata (if necessary)	Priority	Reference
1	**Title:** Get information about a software origin stored in SWH. Explaination: API utilising SWH "Archive" endpoint.	H	https://archive.softwareheritage.org/api/1/
2	**Title:** Get information about the metadata of some software origin stored in SWH. Explaination: API utilising SWH "MetaData" endpoint.	H	https://archive.softwareheritage.org/api/1/
3	**Title:** Trigger Archival to SWH Explaination: API utilising SWH "Request archival" endpoint.	H	https://archive.softwareheritage.org/api/1/

- 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

8.3.1.2 Specifications

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Architectural design

Publication and Propagation

- Publisher notifies authors of their article (with source code archived) acceptance.
- Publisher simultaneously triggers an API propating metadata to Catalogs (e.g. dblp).
- Publisher considers various citation formats (BibLaTex, CSL, codemeta.json).

The workflow ensures that Software is archived, describable, referenceable, and citeable.

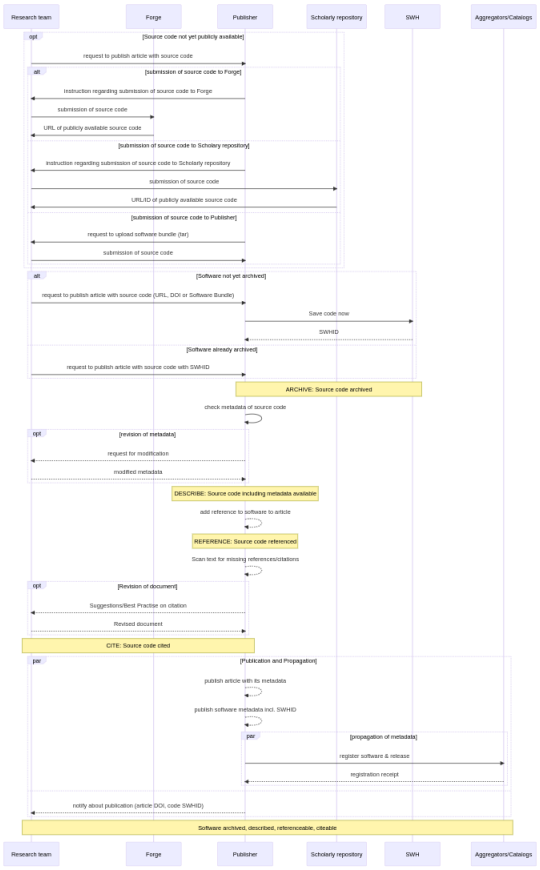

Figure 29 – Sequence diagram publisher workflow

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.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
#	Short description	Priority	Reference
	<Provide a short description of functional technical specification> Given the above workflow of the Archiectural Design, we aim to provide several back- and front-end serives that handle the four SIRS report pillars each. Bundling these servics altogether into an operational infrastructure used by Dagstuhl; it should be possible to promote software archiving to SWH, enriching metadata stored in SWH. We also aim to implement a web client (php-based) that interacts with the SWH Merkle DAG data model on the SWH server-side.		<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 relevant SWHID. Then proceed either to point number 5) or proceed to point number 3).	H	SWH Endpoints: /api/1/origin/visit/ /api/1/origin/get/ /api/1/origin/resolve/
3	Validate latest version in SWH is compatible with the submitted artefact. If necessary, proceed to point number 5) on inconsistent comparison. With the help of an internal DB, we can track where the changes have been introduced to the repository.	H	SWH Endpoints: /api/1/snapshot /api/1/release /api/1/revision /api/1/directory/ /api/1/content/
4	Validate the stored metadata in SWH is compatible with that of the submitted artifact.	H	SWH Endpoints: /api/1/raw-extrinsic-metadata /api/1/origin/intrinsic-metadata
5	Trigger (Re-)Archival of the submitted source code.	H	SWH Endpoint: /api/1/origin/save /api/1/origin/save/webhook/

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

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	Every subcomponent may interact with Dagstuhl's workflow		

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

8.3.1.3 External references

External references - T6.2 API and Connectors Between Open Access Publishers and Software Heritage		
#	Short description	Reference
1	Links will be provided for testing after the system has reached Beta state.	
2	Software Heritage deposit documentation	https://docs.softwareheritage.org/devel/swh-deposit/api/index.html#deposit-api

8.3.2 Episciences - SWH SPECS

Component	Episciences connector and API to Software Heritage
Category	**RSAC**

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Contact person	Alain Monteil Maud Medves
Contributors	Alain Monteil Maud Medves
Version	2.0
Date	20.01.2023

Overview
<Provide a high-level description of the component/service> Develop APIs and connectors between Software Heritage and Episciences • Archival: Source code is already archived on Software Heritage • 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 to publications; export citation information in one or more of the common open citation formats (BibLaTeX, CSL, codemeta.json).

Objectives	
#	Short description
1	Integrate and showcase the Overview into operational infrastructures used by Episciences.
2	Archival: ensure the archival of source code in HAL and/or in Software Heritage associated to research articles;
3	Reference: expose the corresponding SWHID on the journal's publication record;
4	Description: enable the deposit and retrieval in Software Heritage of curated metadata for software associated to publications;
5	Description: ensure that modifications in the article's metadata will be deposited in Software Heritage.
6	Citation: deposit and retrieve the preferred citation information for software associated to publications; add to citation information the reference to software source code.
7	Citation: export citation information in one or more of the common open citation formats - BibTeX

Out of Scope	
#	Short description

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Out of Scope	
1	automate archival from online public repository using a url in Software Heritage on the Episciences platform (this step will be done by the user and not by the infrastructure)
2	export citation information in one or more of the common open citation formats (CSL, codemeta.json).

8.3.2.1 Requirements

User stories		
#	Description of the user story	Reference
...	<Provide 1 or more user stories which are example to using the component/service>	<URL to an external reference describing the user story in more detail>
1	A researcher can specify a SWHID for the associated source code, when submitting an article to Episciences.	Archive: Use Case A3
2	Episciences adds a reference to the archived software artifact so that: 2.1 reviewers have access to the source code in the peer-reviewing form following the permalink SWHID 2.2 readers can access the code via a link on the record landing page 2.3 exported metadata contains the SWHID	Reference: Use Case R2
3	Researcher adds a SWHID reference of the source code on the article's pdf , following requirements of the Episciences journal	Reference: Use Case R1
4	Researcher can use Scholary Repository (HAL, etc.) to enrich metadata with specific software metadata. The metadata will be then visible on the Episciences journal record when using the identifier to reference the record and source code artifacts	Describe: Use Case D1
5	Episciences provides export format in BibTeX with software source code reference	Describe: Use Case D2
6	Episciences provides best practices and guidelines addressing citation of software.	Cite: Use Case C2

User requirements (user = Episciences)				
#	Short description	Priority	Feasibility	Reference
1	An API to retrieve metadata from scholarly repository	H	5	TBD - link to github Issue

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements (user = Episciences)				
2	An API to retrieve metadata from Software Heritage	M	5	TBD - link to github Issue
3	An API to access source code via SWHID resolution	H	5	TBD - link to github Issue
4	An API to show iframe of the software source code available in Software Heritage	L	3	TBD - link to github Issue

Functional requirements			
#	Short description	Priority	Reference (https://github.com/CCSDForge/episciences)
1.1	**Title: accept SWHID reference on Episciences platform for software artifact** Explanation: add a "related material / software" (or equivalent) field in the submission form where the authors can enter the SWHID of a source code associated with the submitted paper.	H	TBD - link to github Issue
1.2	**Title: accept HAL reference on Episciences platform for software artifact** Explanation: add a "related material / software" (or equivalent) field in the submission form where the authors can enter the HAL identifier of a source code associated with the submitted paper.	H	TBD - link to github Issue (same as above)
2.1	**Title: add a reference to the archived software artifact on reviewer's peer-reviewing form** Explanation: expose the archived software artifact on the peer reviewing form, so that the reviewer has access to it when peer reviewing the submitted paper.	H	TBD - link to github Issue
2.2	**Title: add a reference to the archived software artifact on record landing page** Explanation: once the article has been published, a specific section should be visible on the article's landing page and display the reference (some metadata) and a link to the archived software	M	TBD - link to github Issue

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements			
	artifact (whether it is on SWH or on a scholarly repository).)		
2.3	**Title: add a reference to the archived software artifact on exported metadata** Explanation: Record's exported metadata (harvestable through an API) should include a reference to the archived software.	H	TBD - link to github Issue
5.1	**Title: Add @software / @softwareversion BibTeX entry to the BibTeX export.** Explanation: For example, in this export add the entry below the @article BibTeX entry: https://jtcam.episciences.org/10110/bibtex link to BibLaTeX TODO: add example	M/L	TBD - link to github Issue
x	**Title: Push/send notification (from Publisher) to SWH upon publication.** Explanation: Send notification or push updated metadata (from Publisher) to SWH or scholarly repository, when a source code is related to a an article, which has just been published.	M	TBD - link to github Issue

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

Non-functional requirements			
#	Short description	Priority	Reference
1	**Title : request from the authors submitting a manuscript that they first archive the associated source code in SWH** Explanation : Add a section in the Episciences documentation to explain to authors that, if they have a related source code, it must be first (self-) archived in SWH and then linked with the paper's submission (via the Episciences interface - functional requirement 1.1).	H	TBD - link to github Issue
2	**Title: add on the documentation or/and requirements of the Episciences journal how to add a reference to the source code on the article's pdf**	M	TBD - link to github Issue

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Non-functional requirements			
	Explanation: Add a section in the Episciences documentation dedicated to referencing the source code in the article's pdf, so that the journals running on the platform know how to expose related software artifacts directly in the article's pdf.		
3	**Title: add on the documentation of the Episciences journal a section explaining how to cite software.** Explanation: providing best practices and guidelines addressing citation of software.	L	TBD - link to github Issue

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

8.3.2.2 Specifications

Architectural design

Researcher submit to a journal Episcience an identifier from Scholary Repository + SWHID for source code archived and related to publication. After reviewing and publication Episciences show publications metadata and at least link to source code archived on software Heritage.
One publication could have several SWHId.

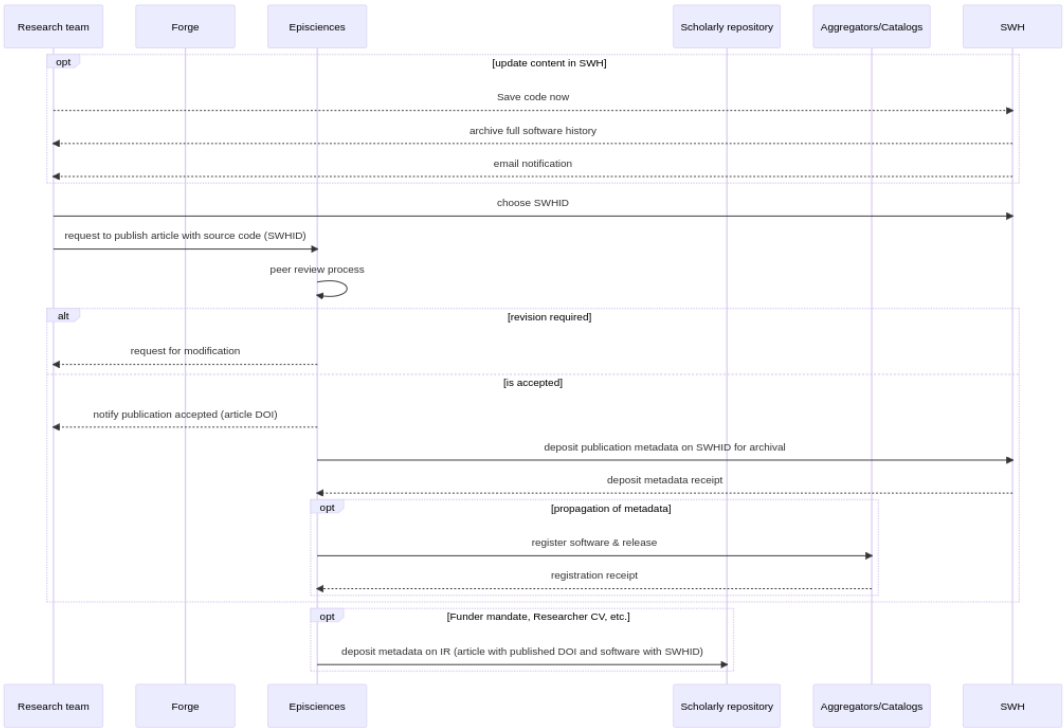

Figure 30

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
#	Short description	Priority	Reference
1	Episcience should use the same workflow between HAL and software Heritage already implemented and in production	H	

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

8.3.2.3 External references

External references		
#	Short description	Reference
1	Episciences: documentation	*https://doc.episciences.org/en/welcome/*
2	*Software Heritage deposit documentation*	*https://docs.softwareheritage.org/devel/swh-deposit/api/index.html#deposit-api*

8.4 RSAC component for aggregators

8.4.1 swMATH - SWH SPECS

Component	swMath connector and API to Software Heritage
Category	**RSAC**
Contact person	Maxence AZZOUZ-THUDEROZ
Contributors	Maxence AZZOUZ-THUDEROZ & Moritz Schubotz
Version	2.0
Date	20.01.2023

Overview
Develop API and connectors between aggregators and Software Heritage to support archival, reference, description and citation as follows: extract references to software from research articles and trigger archival of source code known to aggregators and missing in Software Heritage, expose the corresponding SWHID in the software artifact record maintained by the aggregator, deposit and retrieve metadata for the software artifacts in Software Heritage, deposit and/or retrieve preferred citation information for the software artifacts in Software Heritage; export citation information in one or more of the common open citation formats (BibLaTeX, CSL, codemeta.json). Integrate and showcase the above into operational infrastructures that are used by research communities, and notably: swMath and OpenAire.

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Objectives	
#	Short description
1	*Archive: extract references to software from research articles and trigger archival of source code known to swMATH*
2	*Reference: expose the corresponding origin (url) and SWHID*
3	*Describe: deposit and retrieve metadata for the software artifacts in Software Heritage*
4	*Cite: deposit and/or retrieve preferred citation information for the software artifacts in Software Heritage; export citation information in one or more of the common open citation formats (BibLaTeX, CSL, codemeta.json)*

Out of Scope	
#	Short description
1	Cite: export citation information in one or more of the common open citation formats (CSL)

8.4.1.1 Requirements

User stories		
#	Description of the user story	Reference
1	*As a scholar I can trigger an archival request in SWH so that the software in the record is safely archived.*	Archive: Use case A1
2	*As a scholar, I can find a specific software referenced on swMath and the URL link containing the SWHID so that I can find the source code on SHW*	Archive: Use case A2
3	*As a scholar, I can find for a given software on SWH the research articles citing it coming from the swMath database.*	Reference: Use case R1
4	*As a scholar, I can connect on swMATH platform, look for a software description and find on SWH the repository with the same description so that I know what the software source code is about.*	Describe: Use Case D1

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories		
5	*As a scholar, I can find research mathematical software on swMATH and its associated metadata, so that I can download the metadata in a BibTex or Codemeta format and use it for citation in my work.*	Cite: Use case C1

User requirements				
#	Short description	Priority	Feasibility	Reference
1	*We assume that the user searches for open source mathematical softwares. He identifies the software on swMATH catalog, finds the archived source code on SWH even if the current version is no longer available.*	H	5	

- 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	*Enrich SWH with software descriptions, research articles citing these softwares, authors, ids and other metadata from the swMATH database*	[H|M|L]	<JIRA issue number> or <URL to external reference>
2	*Trigger archival on SWH platforme for software artifact URL*	H	
3	*Display a URL link on swMATH containing the SWHID bringing to the mathematical archive on SWH platform*	H	
4	*Deploy a functionality to let the user download the metadata of a software in a BibTex or Codemeta format*	H	

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

Non-functional requirements			
#	Short description	Priority	Reference
1	*Build a documentation of the service*	M	
2	*Write a guideline*	L	
3	*Legal aspects (Data transfer agreements, SLA, zbMATH Open data legacy... etc.)*	H	
4	*Tutorials for users*	L	

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

Funded by the European Union
FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

8.4.1.2 Specifications

Architectural design

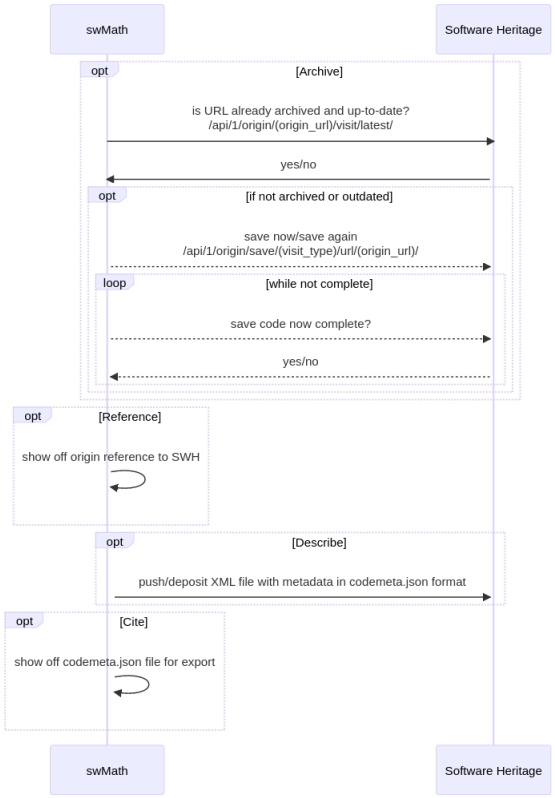

Figure 31 – Sequence diagram - swMath to SWH

See diagram designed in August 2022 during the swMath - SHW meeting

This diagram is inspired from push and pull workflow proposed in the SIRS report.

https://hedgedoc.softwareheritage.org/fc4e-wp6-t6-3-specs-template?both#

Functional specifications

#	Short description	Priority	Reference
	<Provide a short description of functional technical specification>	[H|M|L]	<JIRA issue number> or <URL to external reference>
1	*Create a script to trigger archival using search API of SWH*	H	*https://docs.softwareheritage.org/devel/swh-search/index.html#swh-search*

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
			/api/1/origins/ /api/1/origin/save /api/1/origin/save/webhook/
2	*Create new front end features for the swMATH website to display link to the SWH software repository*	H	https://zbmath.org/software/
3	*Create an API to enrich with software metadata (descriptions, articles citing the softwares, swMATH id, etc...)*	H	https://oai.zbmath.org/v1
4	*Identify URL and verify if most recent version is in SWH*	*H*	/api/1/release/ /api/1/snapshot/
5	*Use SWH API's save endpoint*	*H*	/api/1/origin/save/
6	*Use SWH API's webhook endpoint*	*M*	/api/1/origin/save/webhook/gitlab/

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

Service specifications			
#	Short description	Priority	Reference
1	*User support: update reviewer guide with information about SWH*	M	https://zbmath.org/reviewer-service/
2	*update FAQ with SWH transfer information*	*L*	*https://zbmath.org/frequently-asked-questions/*
3	*establish SLA*	*L*	

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

Operational specifications			
#	Short description	Priority	Reference
1	*Deployment in production:* *We will use MediaWiki as a software platform.*	H	
2	*Admin to contact for the service developed by zbMATH Open:* *edititor@zbmath.org*	L	
3	*Monitoring of logs of the deployed service*	M	

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	zbMATH Open will deploy this services such they can be intgrated with EOSC Core components like OpenAIRE, Datacite and Dagsthul	M	

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

8.4.1.3 External references

External references		
#	Short description	Reference
1	Links will be provided for testing after the system has reached Beta state.	
2	*Software Heritage deposit documentation*	*https://docs.softwareheritage.org/devel/swh-deposit/api/index.html#deposit-api*

8.4.2 OpenAIRE - SWH SPECS

Component	
Category	**RSAC**
Contact person	Paolo Manghi
Contributors	Paolo Manghi , Thanasis Vergoulis
Version	2.0
Date	20.01.2023

Overview
This component implements an API and connectors between OpenAIRE and Software Heritage to support archival, reference, description and citation of research software artifacts leveraging the CodeMeta standard, and the Software Heritage intrinsic identifiers (SWHID).

Objectives	
#	Short description

Objectives	
1	**Archival:** To trigger the archival of software artifacts which are already included in OpenAIRE (since they have been identified based on their mentions in research articles) but missing from SWH. SWHIDs will be acquired for the respective software artifacts.
2	**Reference:** To maintain and expose the SWHIDs for all the software artifact records maintained by OpenAIRE, where applicable.
3	**Description:** To deposit missing software artifact metadata in SWH and to retrieve software artifact descriptions from SWH to be included in the respective OpenAIRE records.
4	**Citation:** To export citation information in BibTex and codemeta.json.

Out of Scope	
#	Short description
1	To export citation information in CSL or other alternative citation formats.

8.4.2.1 Requirements

User stories		
#	Description of the user story	Reference
1	A researcher that examines a software artifact record in OpenAIRE can then find the archived source code from SWH exploiting the connection.	https://hedgedoc.softwareheritage.org/fc4e-wp6-t6-3-specs-template#212-Use-case-OpenAIRE
2	A researcher that examines a software artifact record in OpenAIRE can find the respective SWHID in the record metadata.	https://hedgedoc.softwareheritage.org/s/fc4e-wp6-t6-3-specs-template#222-Use-case-OpenAIRE
3	A researcher that examines a software artifact record in OpenAIRE can find metadata coming from SWH and vice versa.	https://hedgedoc.softwareheritage.org/s/fc4e-wp6-t6-3-specs-template#232-Use-case-Describe-software-on-OpenAire
4	A researcher exports citation information for a desired software artifact record in OpenAIRE in Bibtex or codemeta format.	https://hedgedoc.softwareheritage.org/s/fc4e-wp6-t6-3-specs-template#242-Use-case-Cite-software-on-OpenAIRE

User requirements				
#	Short description	Priority	Feasibility	Reference

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements				
1	Functionality to get the source code related to an OpenAIRE software artifact record	M	3	Ticket TBD
2	Functionality to get the SWHID of a given OpenAIRE software artifact record	M	4	Ticket TBD
3	Functionality to get metadata for a given SWH software artifact record including OpenAIRE metadata	M	3	Ticket TBD
4	Functionality to get metadata for a given OpenAIRE software artifact record including SWH metadata	M	3	Ticket TBD
5	Functionality to get the citation information of a given OpenAIRE software artifact record	M	3	Ticket TBD

- 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	Trigger archival of an OpenAIRE software artifact record on SWH (and return SWHID).	H	Ticket TBD
2	Retrieve metadata from SWH (including citations) to OpenAIRE.	H	Ticket TBD
3	Retrieve metadata from OpenAIRE (including citations) to SWH.	H	Ticket TBD

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

Non-functional requirements			
#	Short description	Priority	Reference
1	Offer documentation pages for the component	M	Ticket TBD
2	Tutorial for users	L	Ticket TBD

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

8.4.2.2 Specifications

Functional specifications			
#	Short description	Priority	Reference
1	API endpoint to trigger archival of an OpenAIRE software artifact record on SWH. Its response will contain the acquired SWHID.	H	TBD

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional specifications			
2	API endpoint to retrieve metadata from SWH to OpenAIRE.	H	TBD
3	API endpoint to retrieve metadata from OpenAIRE to SWH.	H	TBD
4	API endpoint offer OpenAIRE software artifact citation information to SWH	H	TBD
5	API endpoint to offer SWH software artifact citation information to OpenAIRE	H	TBD

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

Service specifications			
#	Short description	Priority	Reference
1	Update OpenAIRE APIs/dumps documentation accordingly	M	TBD
2	Prepare tutorial for the new functionalities	L	TBD

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

Operational specifications			
#	Short description	Priority	Reference
1	Prototype of the component will be hosted in OpenAIRE premises	H	TBD

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

Integration with EOSC Core components			
#	Short description	Priority	Reference
1	The component will be part of a core FAIRCORE4EOSC component (RSAC) to be included in EOSC core	H	N/A

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

8.4.2.3 External references

External references		
#	Short description	Reference
1	OpenAIRE Guidelines	https://guidelines.openaire.eu/en/latest/
2	OpenAIRE Dumps	https://graph.openaire.eu/develop/graph-dumps.html
3	OpenAIRE APIs: general info	https://graph.openaire.eu/develop/overview.html
4	OpenAIRE APIs: authentication & limits	https://graph.openaire.eu/develop/authentication.html

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

External references		
5	OpenAIRE APIs: response metadata format	https://graph.openaire.eu/develop/response-metadata-format.html

9 WP7 – User Requirements from the Case Studies

9.1 Climate Case Study

Overview

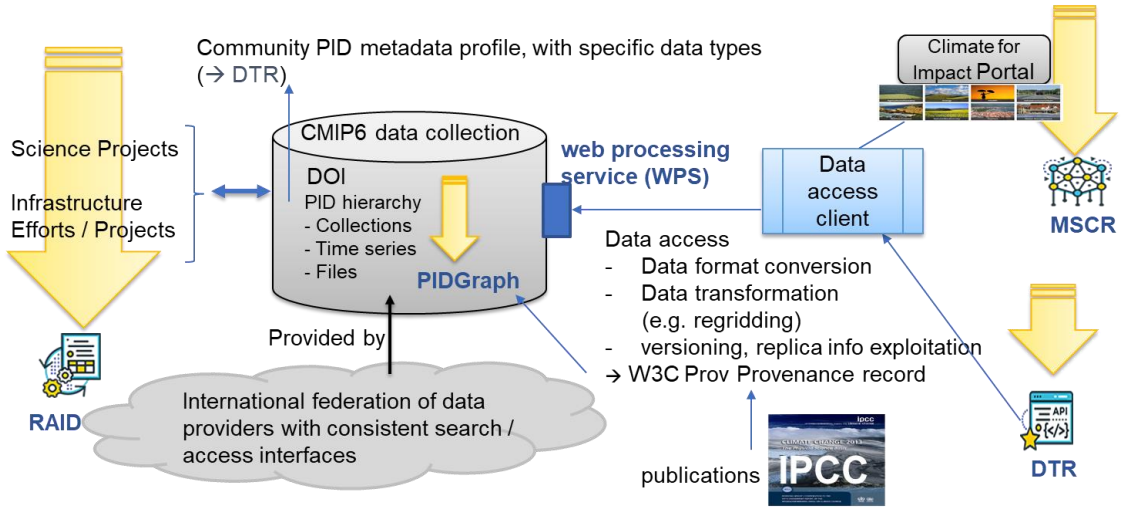

Figure 32

User stories

#	Description of the user story	Relevant for ENES component	FC4E Component(s)	Reference
1	As a researcher I would like to explore the broader context of a specific project/activity e.g. to explore which project/activity level to use to reference in a publication. The researcher starts with a specific CMIP6 activity which is referenced from a CMIP6 data collection in the portal (given as a specific RAID) and then can follow the activity graph up (broader activity context) or down (specific activity context).	ENES portal(s)	RAID	<URL to an external reference describing the user story in more detail>

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories				
2	As a researcher I would like to get the (directly) to the data collection(s) produced in a specific activity context (e.g. a specific CMIP6 sub-project given as a specific RAID). I would also like to explore the broader or more specific data collections associated to related activities in the RAID graph (this relies on User story 1).	ENES portal(s) or tool which can be integrated in jupyter notebooks or used stand alone	PIDGraph, RDGraph	
3	As a forest manager, I would like to be able to find, utilize and interprete climate variables (e.g. indices on 'days with average temperature > 25° C) in order to plan reforestration of a destroyed forest area.	MSCR provides crosswalks (semantic mappings) of climate parameters (described by CF conventions) to vocabularies interpretable in the application contxt	T4.2.2 MSCR - Functional specifications	
4	As a user of a downstream community data portal (e.g. the C4I portal) I want to access some pre-processed data (not the original data provided from the ENES data portals). For this the downstream portal calls a web processing service doing data reduction and data transformation operations on the original data. This call is done by a data access client (SW agent) integrated in the downstram community portal. To be able to develop, reuse and sustainably provide such data access clients these clients need specific type information for the data sources as well as the requested pre-processed results (and mybe in the future also for the web processing services themselves).	Downstream community portal (e.g. C4I portal) "Intelligent" Data access clients using data type information to trigger/parameterize computions	PIDGraph, DTR	
5	As a user of a downstram community (e.g. climate impact) data portal I want to access data based on the specific terms used in my domain. Thus data access involves 1) vocabulary crosswalks with respect to ther terms used by the data provider infrastructure and 2) triggering the needed data transformation steps to derive the requested data out of the original data. (Thus user story thus links user story 3 and 4).	downstream community portal	MSCR, PIDGraph, DTR	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories				
6	to add later: IPCC author user story - relies to some degree on user story 1, 2, 4, 5)			

User requirements				
#	Short description	Priority	Feasibility	related user story
1	Call the RAID service via API to 1) retrieve all direclty related activities/projects and 2) retrieve metadata for these	H	4	User story 1
2	Given a PID, go up and down the PID collection hierarchy (PIDGraph) to find associated RAIDs	H	3	prerequisite for ser story 1, 2 if user starts from a known data collection and wants to explore the RAID context
3	Given a PID (of the input data) retrieve type information from DTR - then use this type information in combination with the requested type information of the output data (and type information of the processing service) to trigger a computation of an OGC web processing service.	L	2	User story 4

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

9.1.1 Specifications

Architectural design
see sketch under this section's overview. Detailed (e.g. UML based) design will be added in future versions

9.1.2 External references

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

9.2 Mathematics Case Study

Component	
Category	CAT, RDGRAPH, PIDGRAPH, MSCR, DTR, RAID, PIDMR, RSAC, SWHM
Contact person	Maxence Azzouz-Thuderoz
Email address	
Contributors	Maxence Azzouz-Thuderoz, Moritz Schubotz
Version	
Data	

9.2.1 Requirements

User stories			
#	Description of the user story	FC4E Component(s)	Reference
1	**As a** mathematician, **I would like to** be able to ensure that the PID provided in a URL link is a **valid identifier so that** I can use the PIDMR service to resolve it and get informations on the origin of the PID.	PIDMR	WP5 - EOSC PID Meta Resolver and PID Kernel Information Profiles
2	**As a** mathematician, **I would like to** be able to find available **crosswalks** for a given metadata schema **so that** I can provide zbMATH metadata in other useful shemata.	MSCR	T4.2.2 MSCR - Functional specifications
3	**As a** mathematician, **I would like to** be able to control **type of inputs** in the metadata of research articles or softwares thanks to the API of DTR **so that** we can ease the conversion of metadata. • Cordra leaves the possibility to control schema types and we expect it to help for schema conversion	DTR	T4.3 EOSC Data Type Registry
4	**As a** mathematician, **I would like to** be able to make relationships between **zbMATH PIDs** (people identifiers, publications/software identifiers) and other related scientific projects references in RAiD **so that** we can improve the discoverability of relevant research entities in the PID Graph	RAID	T4.4 EOSC Research Activity Identifier Service (RAiD)
5	**As a** mathematician, **I would like to** be able to make relationships between **zbMATH people identifiers** and ORCIDs available in the	PIDGRAPH	WP3 - EOSC Federated Search

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories			
	PID Graph, **so that** PID Graph queries can be made in the PID graph.		Service and PID Graph Service
6	**As a** mathematician, **I would like to** be able to make relationships between **zbMATH publications/software identifiers** and **zbMATH people identifiers** available in the PID Graph, **so that** PID Graph queries can be made in the PID graph	PIDGRAPH	WP3 - EOSC Federated Search Service and PID Graph Service
7	**As a** mathematician, **I would like to** be able to make relationships between **zbMATH publications/software identifiers** and **DOIs** be available in the PID Graph, **so that** PID Graph queries can be made in the PID graph.	PIDGRAPH	WP3 - EOSC Federated Search Service and PID Graph Service

User requirements				
#	Short description	Priority	Feasibility	Reference
1	<Provide a short description of the user requirement>	[H|M|L]	[1-5]	<JIRA issue number> or <URL to external reference>
…	Any mathematicians that can access softwares or articles and their metadata on zbMATH Open	M	4	

- 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	FC4E Component(s)	Reference
	<Provide a short description of functional requirements>	[H|M|L]		<JIRA issue number> or <URL to external reference>
1	Build an API that ensures swMATH PID validity with PIDMR services.	M	PIDMR	WP5 - EOSC PID Meta Resolver and PID Kernel Information Profiles
2	Build an API that ensures swMATH metadata format conversion	M	MSCR	T4.2.2 MSCR - Functional specifications

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

Functional requirements				
3	Build an API that ensures possibility to control types of inputs in the metadata of research articles or softwares	M	DTR	T4.3 EOSC Data Type Registry
4	The API will return the related digital object for a given PID coming from swMATH	M	RAID	T4.4 EOSC Research Activity Identifier Service (RAiD)
5	Build an API that ensures zbMATH people identifiers to be matched with ORCIDs	H	PIDGRAPH	WP3 - EOSC Federated Search Service and PID Graph Service
6	Build an API that ensures zbMATH people identifiers to be matched with zbMATH publications/software identifiers	H	PIDGRAPH	WP3 - EOSC Federated Search Service and PID Graph Service
7	Build an API that ensures zbMATH publications/software to be matched with DOIs	H	PIDGRAPH	WP3 - EOSC Federated Search Service and PID Graph Service

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

Non-functional requirements			
#	Short description	Priority	Reference
	<Provide a short description of non-functional requirements>	[H|M|L]	<JIRA issue number> or <URL to external reference>
1	Establish a documentation for each component implied in the development of this use case	H	
2	Come with guidelines to highlight how to take advantage of the deployed services	M	
3	Legal aspects (Data transfer agreements, SLA, zbMATH Open data legacy... etc.)	M	
4	Build tutorials for mathematicians	M	

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

9.2.2 Specifications

9.2.3 External references

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

9.3 Service Providers and Research Data Management Communities Case Study

The case study will demonstrate and showcase some of the developed components and tools in other tasks especially in WP4 into the EUDAT's publishing and sharing service B2SHARE as well as to propagate some of this integrations to B2FIND to enrich and to improve the contents displayed and provided by the B2SHARE & B2FIND services.

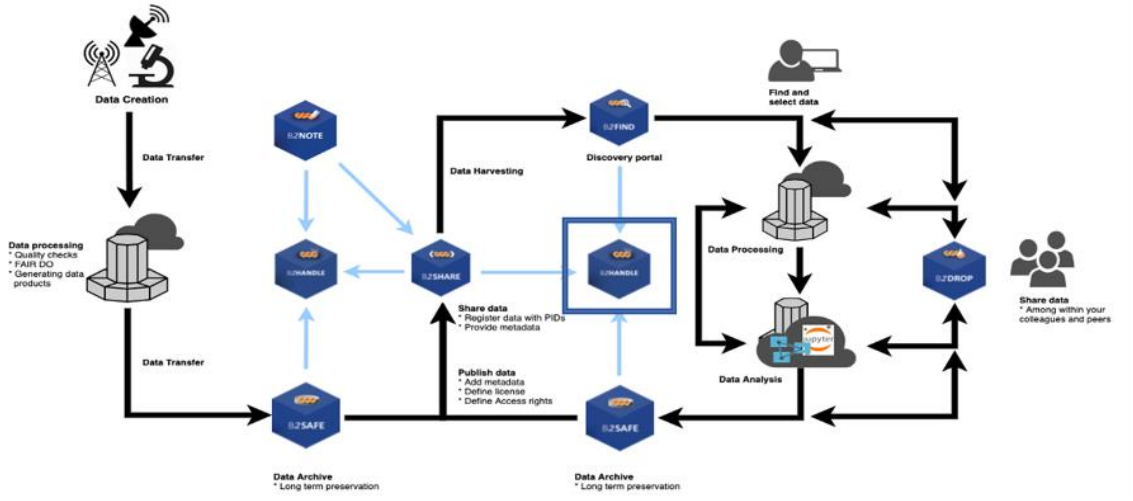

Figure 33

The case study will demonstrate and showcase the following:

- EOSC **Data Type Registry** (DTR) to provide user friendly APIs for metadata imports and access to different data types and metadata mappings
    - Integration of Data Type registry and adding type-identifiers to metadata schema
    - Make use of developed tools and components to automatically register and retrieve data types to/from data type registry
- EOSC **Metadata Schema and Crosswalk Registry** (MSCR) to support publishing, discovery and access of metadata schemas and provide functions to operationalize metadata conversions by combining crosswalks
    - Explore the possibility to facilitate a two way traffic so that B2SHARE can derive specific community extensions from EOSC registry hosted metadata schema and push own metadata schema to MSCR

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

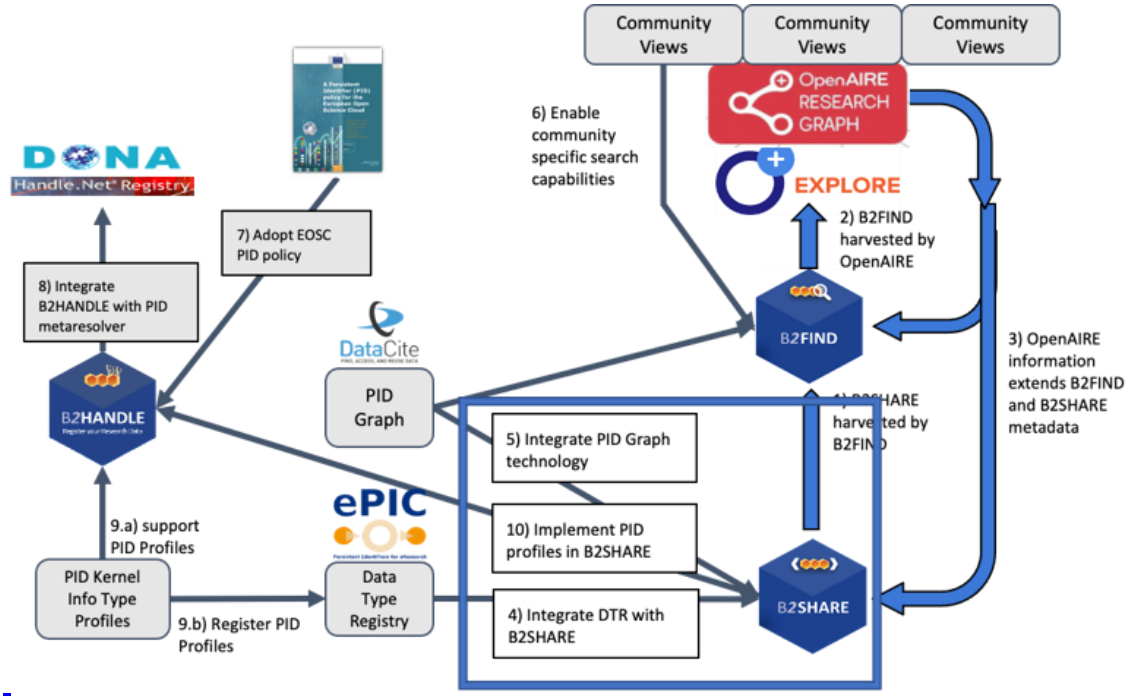

Figure 34

User stories				
#	Description of the user story	Relevant for EUDAT component(s)	FC4E Component(s)	Reference
1	Make available the necessary APIs to query, update, upgrade and integrate with B2SHARE. Having the possibility to explore and see available schemas as well as to be able to update and upgrade them	B2SHARE, B2FIND	MSCR	
2	Make available the available the necessary APIs to query available types, upload new types	B2SHARE, B2FIND	DTR	
...				

9.3.1 Specifications

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

9.3.2 External references

9.4 Social Sciences and Humanities Case Study

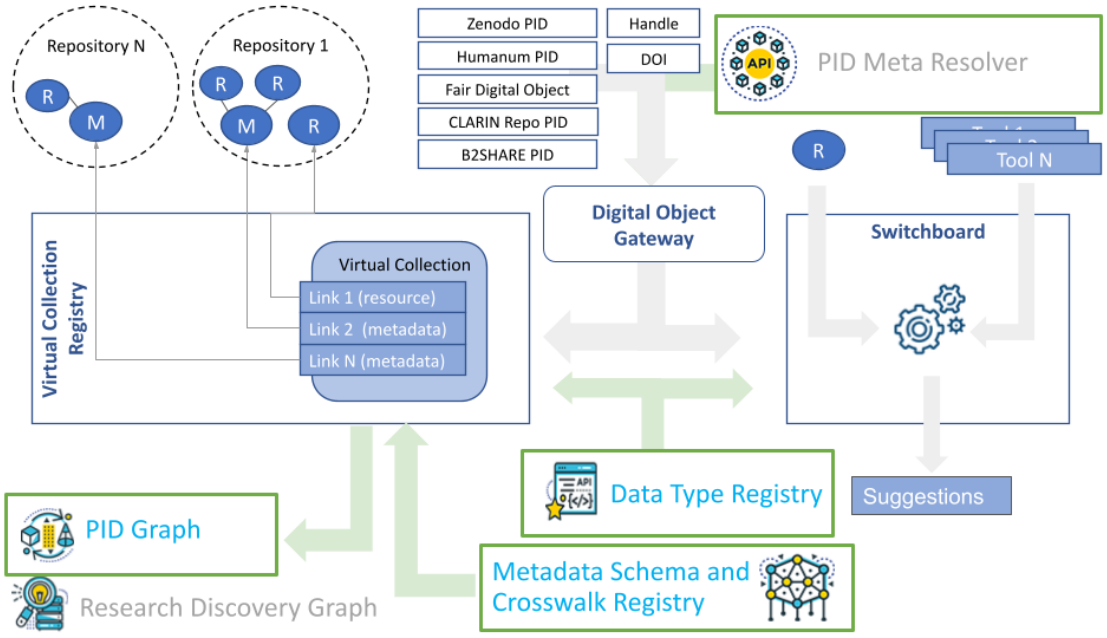

Figure 35

User stories				
#	Description of the user story	Relevant for CLARIN Component	FC4E Component(s)	Reference
1	A client (Switchboard, VCR, …) wants to be able to resolve a given identifier into an url pointing to the associated object. • Resolve: PID → URL • PIDMR allows resolution of identifiers not directly supported in the DOG. I.e. PIDMR will augment DOG functionality.	Digital Object Gateway	PIDMR	
2	A client (Switchboard, VCR, …) wants to be able to query if an identifier exists for a given url (reverse lookup, improves references used within clients). • Reverse resolve: URL → PID	Digital Object Gateway	PIDMR	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories				
	• PIDMR allows reverse resolution of urls not directly supported in the DOG. I.e. PIDMR will augment DOG functionality.			
3	A client (Switchboard, VCR, …) wants to be able to get the url pointing to a resource, even if the identifier points to a landing page or metadata record. • If a crosswalk is available from the MSCR, given the input schema of the landing page / resource, this can be used to transform into an understood format.	Digital Object Gateway	MSCR	
4	A researcher wants to know what tools are available within the CLARIN infrastructure to process a given (set of) resource(s). • o Given the data type of the resource and the supported data type of the tool make a suggestion regarding the tools than can process the resource. o Using well defined data types from a shared registry can reduce potential mismatches o Improve the matching based on intent/context. I.e. a researcher wants to perform text analysis on images with text content. The data type will be recognized as image/<subtype>, instead we would like to be able to signal image/<subtype>+written text and have a toolchain that first performs OCR on the image and then allows for text analysis of the result.	Switchboard	DTR	
5	• A researcher, without in-depth knowledge regarding data types, wants to override the auto-detected data type of a resource (auto-detect can be wrong, failed, …). o Media → video → mpeg4 o Text → analysis → tei → tei subtype o Text → xml → cmdi • Offer guidance to arrive at the most specific relevant data types • Requires a taxonomy which can be queried from the DTR • Same approach can be used to help tool owners select the proper data types for their tools (in case there is any doubt)	Switchboard	DTR	

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User stories				
6	We would like to increase the discoverability of virtual collections • ○ Extend the information collected in the minted DOI to improve integration with the PIDGraph	Virtual Collection Registry	PIDGraph, RDGraph	
7	Provide better suggestions for link metadata to user's while editing a collection • If a user adds a link by PID or URL, the associated object is analysed to extract relevant information to fill in basic fields such as title, description, .. stored with each link in the collection • The MSCR can provide functionality to add support for more schemas by providing a crosswalk to a common schema.	Virtual Collection Registry		
8	Provide a uniform view of the collection with proper links to the referenced resources • When using the collection as input for machine based workflows, the MSCR can provide crosswalks to support more schemas where we can extract links to the actual bitstreams of the resources	Virtual Collection Registry, Digital Object Gateway, Switchboard		
9	Allow to store intent with a resource • Properly store information about the itent of a collection and/or link, based on functionality offered by the DTR (see story 4)	Virtual Collection Registry		

9.4.1 Requirements

User requirements						
#	Story	Short description	Relevant for CLARIN Component	FC4E Component	Priority	Reference
1	1	Given an identifier return the url or error	Digital Object Gateway	PIDMR	[H|M|L]	<JIRA issue number> or <URL to external reference>
2	1	Return list of resolvable identifiers (by scheme?)	Digital Object Gateway	PIDMR		

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.

User requirements						
3	2	Given an url return the identifier pointing to this url.	Digital Object Gateway	PIDMR		
4	3,7,8	Fetch available crosswalks for a landing page / metadata record	Virtual Collection Registry, Digital Object Gateway, Switchboard	MSCR		
5	3,7,8	Transform landing page / metadata record given a crosswalk	Virtual Collection Registry, Digital Object Gateway, Switchboard	MSCR		
6	4	Standard CRUD actions for data types • o Proper documentation with the DTR and CRUD functionality	Switchboard	DTR		
7	5,9	Support extended data types to capture intent. • o I.e image/jpeg+written text o Have support for base crud functionality for these extended types	Switchboard, Virtual Collection Registry	DTR		
8	5	Support a taxonomy on top of the specific data types • o Query data type taxonomy o Have support for basic crud functionality for taxonomys	Switchboard	DTR		
9	6	Store author information in DOI metadata	Virtual Collection Registry	PIDGraph		
10	6	Store links to other objects in DOI metadata	Virtual Collection Registry	PIDGraph		

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

FAIRCORE4EOSC has received funding from the EU's Horizon Europe research and innovation programme under Grant Agreement no. 101057264.