Project Title	European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructure
Project Acronym	ESCAPE
Grant Agreement No	824064
Instrument	Research and Innovation Action (RIA)
Topic	Connecting ESFRI infrastructures through Cluster projects (INFRA-EOSC-4-2018)
Start Date of Project	01.02.2019
Duration of Project	42 Months
Project Website	www.projectescape.eu

D3.3 - CONCEPTUAL DESIGN REPORT ON THE SOFTWARE AND SERVICE REPOSITORY

Work Package	WP3, OSSR
Lead Author (Org)	Thomas Vuillaume (LAPP, CNRS)
Contributing Author(s) (Org)	Enrique García (LAPP, CNRS) Kay Graf (FAU)
Due Date	31.07.2020, M18
Date	31.07.2020
Version	1.0

Dissemination Level

X	PU: Public
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

Versioning and contribution history

Version	Date	Authors	Notes
0.1	10.07.2020	Thomas Vuillaume, CNRS-LAPP	Conceptual design report on the software repository
0.2	13.07.2020	Kay Graf, FAU	added comments
0.3	20.07.2020	Thomas Vuillaume, CNRS-LAPP	review of the comments
0.4	30.07.2020	Thomas Vuillaume, CNRS-LAPP	Review of the comments, minor chagnes
1.0	31.07.2020	Thomas Vuillaume, CNRS-LAPP	Final version

Disclaimer

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Executive Summary

This document constitutes deliverable D3.3 of the ESCAPE project, the *Conceptual design report on the software and service repository*.

Project Summary

ESCAPE (European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures) addresses the Open Science challenges shared by ESFRI facilities (CTA, ELT, EST, FAIR, HL-LHC KM3NeT and SKA) as well as other pan-European research infrastructures (CERN, ESO, JIVE and EGO) in astronomy and particle physics. ESCAPE actions are focused on developing solutions for the FAIRness of large data sets handled by the ESFRI facilities.

These solutions shall: i) connect ESFRI projects to EOSC ensuring integration of data and tools; ii) foster common approaches to implement open-data stewardship; iii) establish interoperability within EOSC as an integrated multi-probe facility for fundamental science. To accomplish these objectives, ESCAPE aims to unite astrophysics and particle physics communities with proven expertise in computing and data management by setting up a data infrastructure beyond the current state-of-the-art in support of the FAIR principles. These joint efforts are expected to result in a data-lake infrastructure providing an open science cloud-based analysis facility linked with the EOSC. ESCAPE supports already existing infrastructures such as astronomy Virtual Observatory to connect with the EOSC. With the commitment from various ESFRI projects in the cluster, ESCAPE will develop and integrate the EOSC catalogue with a dedicated catalogue of open-source analysis software. This catalogue will provide researchers across the disciplines with new software tools and services developed by the astronomy and particle physics communities. Through this catalogue, ESCAPE will strive to provide researchers with consistent access to an integrated open-science platform for data-analysis workflows. As a result, a large community "foundation" approach for cross-fertilisation and continuous development will be strengthened. ESCAPE has the ambition to be a flagship for scientific and societal impact that the EOSC can deliver.

Table of content

1. Introduction

The ESFRI projects in ESCAPE will provide open access to their data, software and science tools to allow reproducibility of science results and the development of open science analyses across domains and infrastructures, necessary for opening the window to multi-messenger and multi-probe research.

The aim of ESCAPE WP3-OSSR (Open-source Scientific Software and Service Repository) is to provide the tools necessary for the communities to share their science products in a harmonised way respecting the FAIR principles, promoting open science and maximising cross-fertilisation by software re-use and co-development. One of the key components to achieve this goal is the software repository. The present conceptual design report documents the work to build a demonstrator for the ESCAPE repository and provide an integrated development platform.

2. Goals and Objectives

Within the context of the ESCAPE project, the need for creating 'standard work-flows' or common practices to develop, share and store software and services, following the FAIR principles (Findable, Accessible, Interoperable and Reusable), is essential. The multitude of development platforms employed by the involved ESF/RIs (GitHub, GitLab, Bitbucket among many others), and all the services that these platforms provide (continuous integration, continuous deployment, continuous delivery, cloud storage, etc.) make some of the FAIR principles difficult to be fulfilled. Thus, rather than imposing certain coding habits to very heterogeneous groups, the inclusive solution chosen by OSSR is to converge into more general common practices that will be evolving with time and needs, which satisfy the use-cases of the involved communities. In the case of the task 3.5 (Repository Implementation and Deployment - RIAD), the two main immediate needs are:

1. providing an open-source development platform available to all the consortium members and to the communities they are representing
2. a long-term repository insuring the implementation of the FAIR principles and the connection with other ESCAPE services such as ESAP (ESCAPE Science Analysis Platform) on one hand and the EOSC portal/marketplace on the other hand

These tools shall be equipped with documentation, guidelines and real use-case examples to help users. A survey was performed among the partners in order to have a representative overview of the needs. Questions and answers to this survey are provided in the appendix.

3. Conceptual design

A repository is first and foremost a trusted place where researchers can safely archive their data and software for long-term preservation and retrieval. But to fully support open science, the repository must provide more services and be integrated in the ESCAPE virtual environment. A schematic representation of this environment is provided in Figure 1.

The repository must be the entry point for any newcomer looking for software through a webportal integrated in the EOSC. It should therefore provide a clear interface, with filtering and search options. Moreover, any contribution should be identifiable without ambiguity, thus allowing citation. The software and analysis themselves must come with a clear documentation and tutorials to maximise reusability. The connection to the analysis platform will allow researchers to easily test the software and (re-)run the archived analysis. It is foreseen that this connection will require the use of containers technologies to ease software preservation, installation and use by the platform. On the other end of the software lifecycle, a development platform is an important service to allow collaborative software development. An automated integration of the development platform with the repository should allow developers to push fixed states of their code to the OSSR (releases).

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

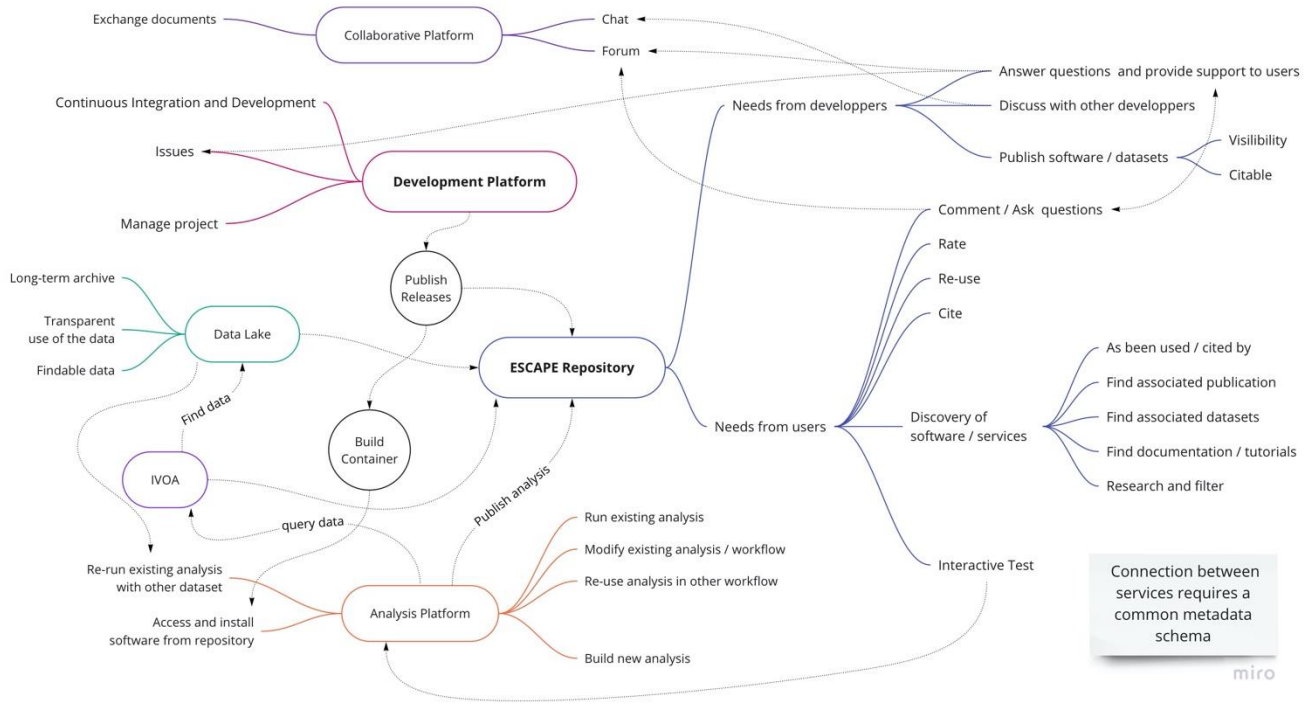

Figure 1: Conceptual design of the OSSR and its connection to the ESCAPE virtual environment

4. Description on the software and service repository

4.1 Development platform

The GitLab development platform hosted at the French National Institute of Nuclear and Particle Physics - in2p3 (1) - was chosen and made available for the whole ESCAPE community. The idea of establishing this service and making it available was not to substitute the development platforms already used by each institution/group. It provides however a common place to gather the common developments, ideas, guidelines and templates for the community, as well as a development platform for new developments if needed by an institution/group and shows the potential integration of this development platform in the software lifecycle. Two main reasons led us to choose the self-hosted GitLab platform rather than GitHub (despite its popularity). First, GitHub's services are based on a closed source and are privately owned (the platform was bought by Microsoft in 2018), and second, the GitLab platform is hosted by one of the ESCAPE consortium members.

Table 1: comparison of GitHub or GitLab as web services and a self-hosted GitLab solution

	Advantages	Drawbacks
GitHub or GitLab as a service	- Free for open source - No setup - Integration with lot of services - Huge community	- Data stored outside the E.U. - No control over the data & conditions of use - No control of accounts
Self-hosted GitLab	- Private projects - Control - Open source	- Private projects - Control - Open source

A GitLab group was created (2) where the developments under the ESCAPE2020 project WP3 are currently being collected. Within this group a 'template project' was created (3). The objectives of this template project are:

1. Provide the main guidelines for developing a new Open Source project (within the ESCAPE context),
2. Have a web platform where the new implementations discussed within the ESCAPE work packages can be tested, and
3. Show a 'real' use case development example.

The project will be reused and customized as per the individual needs of ESCAPE partners. The idea behind a template is that any user can make a copy of it (clone or fork the repository) and start making changes / continue developing from the state of this work. The main components a repository must contain are:

- An open source license file (the license and provenance model will be covered in D3.7);
- Various *README* files with instructions/explanations of the different packages within the project and their usage. Within these files the contribution guidelines (instructions) are also explained in detail;
- A *setup.py* file to install the project,
- an example of code within a repository (a very simple function making an operation),
- a unitary test to test the correct performance of the developed code,
- other secondary standard git files (a *.gitignore* with examples and a *environment.yml* files...)
- and a continuous integration-continuous delivery/deployment (CI/CD) file (described later).

4.2 Repository

The development platform handles the need to develop and build the software, insuring its re-usability and interoperability of code whereas the repository handles the findability, accessibility and long-term archiving of the software. Various alternatives have been investigated to achieve these goals, in the respect of the FAIR principles, and with a vision beyond the duration of the ESCAPE project (costs, maintenance, …).

The Zenodo repository service (4) was created in May 2013 in collaboration between CERN (5) and the OpenAIRE project (6), and is hosted by CERN. It is developed to respect the FAIR principles and is compliant with the recommendations from OpenAIRE. Its main features are to provide a long-term, safe and secured archive. The service is able to deal with many kinds of content (software, datasets, documents...) and publication versioning, providing a unique DOI for each version, thus allowing precise citation of the used software or dataset. The use of keywords and communities will allow us to filter and organize the content of the repository.

Zenodo is developed as an open source project and the developer team is now developing a generic open source research data management platform, Invenio RDM (7), that will be the future foundation of Zenodo. A videoconference was set on February 2020 with the main developers and project leaders of the Invenio software to explain ESCAPE needs and have a better overview of the Zenodo service. From these discussions and study of the advantages and drawbacks, summarized in the following table, of using Zenodo as a web-service (hosted by CERN), versus self-hosting a digital repository, it was decided to use Zenodo as a web-service. The main reasons for this choice were that the Zenodo

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

repository has the continuous support of the CERN infrastructures and resources, thus ensuring the long-term availability. Moreover, CERN is an ESCAPE consortium member.

Table 2: comparison of Zenodo as a web service and a self-hosted repository solution

	Advantages	Drawbacks
Zenodo as a service	- Free - No setup - CERN support - Secure,very long time, archive - Always up to date - Integration with OpenAIRE	- Size limitation to 50GB per upload (an agreement is possible for bigger datasets) - No multiple ownership as of today - No multiple curators as of today - Limited customization
Self-hosted zenodo (or like)	- Better branding - Own the data - Custom front-end - Custom A&A	- Time and costs to develop/setup/install - Computing infrastructure required - Maintenance - Long-term planning required

Zenodo makes use of DOIs (digital object identifier) to identify uniquely every entry that is uploaded to its database. In other words, a DOI is the fingerprint of a deposit. The deposited entries can be all form of scientific products: datasets, source code/software, publication, posters etc. All the deposited entries must contain certain compulsory information (title, description of the entry, authors, publication date, access right and license, etc.). This information will be converted into human and machine-readable metadata within the repository, as explained later. Once an entry is published, a DOI will be automatically assigned to the publication and the entry will be integrated into the repository. An upload and its corresponding DOI cannot be erased, although a publication can always be updated with a new version (and therefore assigned with a new DOI that will be linked to with the old one(s)). Thus, the repository is conceived to be the warehouse of stable releases (in the case of software) and not the place to store minor changes or tests of a project. This last purpose is precisely the function of the development platform.

Among different services, Zenodo provides a full connection with the GitHub platform. If both services are connected (which must be done manually the first time), the creation of a release within the GitHub platform will automatically create an upload to Zenodo and an automatic assignation of a DOI. This application is extremely useful from the ESCAPE development context and will be extended to the GitLab instance in the near future. A stable solution, for the connection between the GitLab development platform and the Zenodo repository was developed within WP3 and described in the next section.

Zenodo also allows the creation of communities to gather together publications and entries under a similar subject/interest. An ESCAPE2020 community was created in the repository (8), where all the different partners have already started to upload stable versions of some of their projects. The upload of entries to the community is extremely important for two main reasons. First, the need of momentum for this initiative (the consortium needs to use this repository), and second, each new upload will be a new use-case that potentially will identify new needs within the developed service.

Zenodo allows to harvest the whole repository through the Open Archives Initiative Protocol for Metadata Harvesting - OAI-PMH (9). This protocol is extensively used by other repositories and was

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

developed precisely to harvest metadata descriptions of records (an uploaded entry to a repository). The harvested metadata for each record are available in various formats; like *oai_datacite* (v4.0 and previous versions), and others. These formats are based on the DataCite Metadata Schema (10). When a new entry is uploaded to Zenodo the provided information/description is directly mapped to the DataCite schema representation. In addition, the repository allows uploading other types of standard metadata schema representations. This means that besides from being able to harvest metadata from the repository in the DataCite standard, Zenodo allows the upload of other machine-readable metadata standards files with the goal of improving the software metadata in general; Zenodo participates in the development and support other standards like the CodeMeta project (11).

4.3 Connection of services

We have seen that the Zenodo repository offers some excellent connection facilities between some applications/services (Zenodo - GitHub), but it is not the case yet between the repository and GitLab. To fulfil this lack of connectivity - various libraries and existing services were implemented to create a pipeline (integrated within the CI/CD chain) that allows uploading a specific state of a repository (together with a containerised image of the source code if desired) to Zenodo within every new software release that it is done to a project.

The automation process to upload a specific state of a repository (from now on, we will just refer to stable releases, as is the only state that it is intended to be uploaded) is composed of two parts:

- The use of the continuous integration service that GitLab offers (12), and
- the use of the *.zenodoci* library to handle the communication between GitLab and Zenodo (what it is called an application program interface - API).

The GitLab CI service is a powerful built-in tool that allows performing numerous automated operations every time a modification is applied to a project. It can therefore be used to automatize the containerisation of code. The GitLab CI service follows the continuous methodologies (continuous integration, delivery and deployment), and is configured through a `*.gitLab-ci.yml*` file. Following the FAIR principles, an open source library (13) containing the code and the instructions to automate the creation of a Singularity image container including a source code. Thus, by providing a Singularity receipt (that creates an image of the project if built by the Singularity code) to the *gitLab-ci.yml* file, together with the implementations of the previously mentioned repository, the GitLab instance will create a Singularity image of the repository and store it as an 'artefact' in the project's repository. In other words;

- the CI runner calls a remote container hosted in a third-party service (Docker Hub) that contains the Singularity source code,
- the remote container builds the corresponding Singularity image by using the Singularity receipt provided,
- and the runner stores the image in the GitLab project instance.

A CI job can contain various stages. This way, after the creation of the image container, the runner can run other stages, and for example, upload the code to different services or platforms.
The library *zenodoci* (14) has been developed to automatize uploads from GitLab to Zenodo (15) via its REST API (16). It can be used to upload any type of content.

These libraries have been integrated and adapted into the ESCAPE template project (3). This way, a real use-case could be tested and implemented to the template as a potential guideline. The first stage is almost straightforward (calling a remote image).

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

The *zenodoci* library is then used to automatically upload the source code (as a compressed archive) to the repository, and if desired and used, the singularity image and the recipe to create the container. The template project contains all this detailed information in the form of *readme* files within the repository (with the detailed instructions for each stage and how to create environment variables to connect the two services for a completely automatic upload). Furthermore, the documentation also explains the specific case of the repository licensing and why open source code from other projects could be included in the project. A schematic diagram of the implemented workflow is depicted on the following Figure 2;

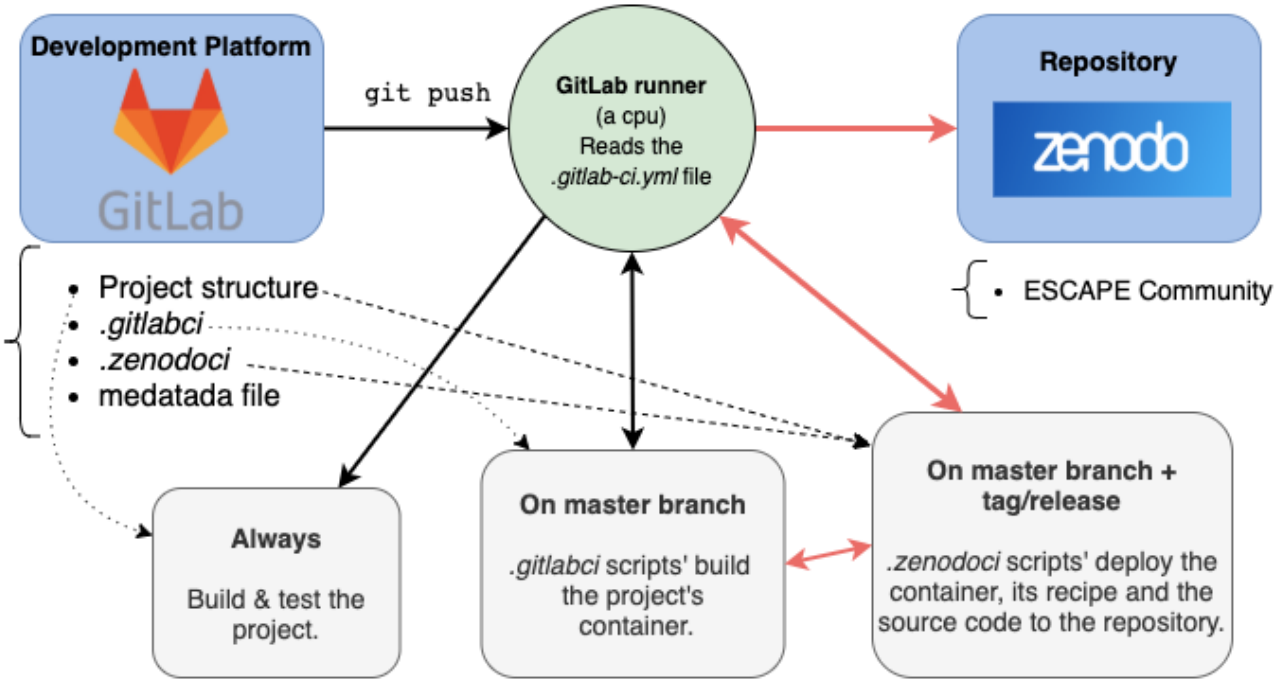

Figure 2: Diagram showing the pipeline between the development platform and the repository through the GitLab-ci stages.

Every entry that has been developed within the ESCAPE context - or is provided by the ESCAPE partners - and has been uploaded to the repository, must follow the FAIR principles. The way to make any software Findable, Accessible, Interoperable and Reusable is highly governed by the correct and complete use of metadata. If an entry is correctly tagged it can be easily connected with other services and or platforms. For example, in case of the ESCAPE analysis platform, another template project was created in the ESCAPE development platform; the ESCAPE metadata template (17). This repository contains the documentation and the file to create an ESCAPE metadata file to be added to every ESCAPE project so that it can be correctly harvested once it is in the Zenodo repository. There have been various format suggestions and a final solution is currently being investigated. One of the solutions is to create a JSON-LD file following the CodeMeta schema mentioned above (18). This solution is being tested because on the one hand Zenodo supports this standard schema, i.e., they can capture metadata from these kind of files, and on the other, the schema maps all the metadata fields that were defined by the FG1 Software requirements and with the collaboration WP3. This means, if any project/entry that is going to be uploaded to the ESCAPE2020 community in Zenodo includes a `*codemeta.json*` file with all the specified fields filled, other external services will be able to correctly find, access and reuse the publication. As a result, the repository fulfils FAIR principles.

5. Conclusion and Future work

The work presented here constitutes the conceptual design and a working demonstrator for the ESCAPE repository and its integration with the development platform, allowing researchers to develop their code in a collaborative environment, publish it respecting the FAIR principles and providing them, and others, the possibility to refer to a stable version of the published software with a DOI. It also provides them with an opportunity to automatically see their software added to the rest of the ESCAPE landscape and its services.Future work should focus on a stronger integration with other ESCAPE services, notably with the analysis platform. This will be possible thanks to a clear definition of the software metadata and of a workflow to retrieve them.

Even though not directly depending on the repository, the development of a landing platform seems necessary at this point to bring together the different ESCAPE services in an integrated environment and provide them better visibility. It must be the entry point in the future for any researcher looking for software, data, documentation and tutorials, either as a user or as a developer. ESCAPE should provide the ways and means to start a new science project easily, with all the modern tools and services, as well as with computing resources supporting them. This landing platform will be integrated with the EOSC portal (as portal of portals).

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

References

1. **[Online] https://in2p3.cnrs.fr/.**
2. *escape development platform.* [Online] https://gitlab.in2p3.fr/escape2020.
3. [Online] https://gitLab.in2p3.fr/escape2020/escape/template_project_escape.
4. Zenodo. [Online] https://zenodo.org/.
5. [Online] https://home.cern.
6. [Online] https://www.openaire.eu.
7. [Online] https://github.com/inveniosoftware/invenio.
8. ESCAPE repository. [Online] https://zenodo.org/communities/escape2020.
9. [Online] http://www.openarchives.org/pmh/.
10. [Online] https://schema.datacite.org/.
11. [Online] https://codemeta.github.io/.
12. [Online] https://docs.gitlab.com/ee/ci/.
13. [Online] https://gitlab.com/singularityhub/gitLab-ci.
14. zenodoci. [Online] https://gitlab.in2p3.fr/escape2020/escape/zenodoci.
15. [Online] https://github.com/SiLeBAT/zenodo-python.
16. [Online] https://restfulapi.net/.
17. ESCAPE template project. [Online]
https://gitlab.in2p3.fr/escape2020/escape/escape_metadata_template.
18. [Online] https://github.com/codemeta/codemeta.
19. [Online] https://www.openaire.eu.

Appendix

WP3-OSSR survey on the repository

This survey was completed in July 2019 to have early feedback from the partners and define their needs concerning the ESCAPE repository. The following report the questions and answers.

Q: What features do you expect from the OSSR repository?

Answers on the left. On the right, the fulfilment of this need by the WP3-OSSR services (development platform and repository)

Supported for a long time	v
Mirror services such as GitHub or GitLab	
Continuous Integration	v
Bug and issue tracking	v
Documentation	v
Storage	v

Common templates for documentation	
Quality review of documentation	v
Keyword-based system for detecting and advertising possible overlaps among projects that may lead to co-developed solutions	v
GitLab-like features with an easy upload/sync possibility with the available solutions	v
A template for software publication (licensing, provenance, A&A,...)	v
Code review	v
Project management	v
Integration with other tools (travis, slack, codacy,zenodo…)	v
Team management	v
Code hosting	v
Code publication and releases archival	v
Github-like features	v
Accessiblity (easy also from within the data-analysis program running on my pc)	v

Q: Do you have one or several repositories in place within your ESF/RI and how are they implemented?

- *Several instances, most of them being GitLab or GitHub*
- *Often scattered repositories*

Q: How would your ESF/RI transfer the software and services to the OSSR (transferring the full development, adding stable releases or similar?)

- Ideally as a mirror of another service
- Adding stable releases
- Adding tagged releases
- stable releases of (header-only libraries) of common interest
- do not know

Q: What software licenses, versioning and means of preservation do you employ?

- Codes mainly in IDL with additions of python. Local versioning system. Latest snapshots published publicly
- Major version + minor version + patch version. FairRoot is stored on GitHub, reduce dependency on third-party software, cross-platform compilation and execution
- GitLab for versioning. Containers for preservation
- Usually MIT license, versioning via GitLab tags/releases (semantic versioning), containerization of software
- BSD (and other open source) licenses - version releases

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

- BSD 3-Clause License. Semantic versioning. Zenodo.
- I used IDL, switched to Python to avoid licenses. Versioning is not a problem to me, so far. Preservation of data is done by NAS and duplication.

Q: How does your ESF/RI implement the FAIR principles to software, services and relevant data to test those, what means are in place for long-term preservation, maintenance and curation?

- New territory in high-resolution solar physics with the exception of solar space missions
- I don't know this at the moment
- FairRoot software framework is an OpenSource software which is publicly available and is used by many experiments. It has dedicated web-page with documentation and news, dedicated forum, bug and issue tracker, continuous integration tests, variety of examples of physics simulation and data analysis and an automated template for new project creation.
- CTA: the software is open-source, findable, accessible, interoperable and re-usable. Some example data is given with the software for development and tests. maintenance and curation is done via git pull-request. No long-term preservation plan right now
- EST is still not producing any data. The issue is being discussed by the community at present time

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Acronym list

Partners

AIP	Leibnitz-Institut für Astrophysik Potsdam
CERN	European Organization for Nuclear Research
CNRS-LAPP:	Laboratoire d'Annecy de Physique des Particules (CNRS)
CNRS-CPPM:	Centre de Physique des Particules de Marseille (CNRS)
NWO-I-CWI:	Centrum Wiskunde & Informatica (NWO-I)
CTA:	Cherenkov Telescope Array
CTAO:	Cherenkov Telescope Array Observatory
EGO-Virgo:	European Gravitational Observatory
CERN:	European Organization for Nuclear Research
EST:	European Solar Telescope
ESO:	European Southern Observatory
ELT:	Extremely Large Telescope (was E-ELT)
FAIR:	Facility for Antiproton and Ion Research
FAU:	Friedrich-Alexander University Erlangen-Nuremberg
GSI:	GSI Helmholtzzentrum für Schwerionenforschung
HITS:	Heidelberg Institute for Theoretical Studies
HL-LHC:	High-Luminosity Large Hardon Collider
IFAE:	Instituto de Fisica de Altas Energias
INFN:	Istituto Nazionale di Fisica Nucleare
JIVE:	Joint Institute for VLBI ERIC
AIP:	Leibnitz-Institut für Astrophysik Potsdam
MPG-MPIK:	Max-Planck-Institut für Kernphysik (MPG)
KM3NeT:	multi-km3 sized Neutrino Telescope
NWO-I-Nikhef:	Nationaal instituut voor subatomaire fysica (NWO-I)
OROBIX:	OROBIX SRL
SKA:	Square Kilometre Array
SKAO:	Square Kilometre Array Organisation
UCM:	Universidad Complutense de Madrid
UNITOV:	Universita degli Studi di Roma Torvergata

General

ASTERICS:	Astronomy ESFRI & Research Infrastructure Cluster
E-EB	ESCAPE Executive Board
EOSC:	European Open Science Cloud
EOSC-Hub:	Integrating and managing services for the European Open Science Cloud
ESCAPE:	European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures
ESFRI:	European Strategy Forum on Research Infrastructures

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

ESF/RI:	ESFRIs and major RIs as projects within ESCAPE
FAIR:	Findable, Accessible, Interoperable, Reusable
IVOA:	International Virtual Observatory Alliance
OSSR:	Open Science Software and Service Repository (ESCAPE WP3 itself and the final product within the EOSC catalogue of services)
RDA:	Research Data Alliance
RI:	Research Infrastructure
VO:	Virtual Observatory
WP:	Work Package

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.