Project title: Radiation risk appraisal for detrimental effects from medical exposure during management of patients with lymphoma or brain tumour (SINFONIA)

Grant Agreement: 945196

Call identifier: NFRP-2019-2020

Topic: NFRP-2019-2020-14 Improving low-dose radiation risk appraisal in medicine

D5.1: Requirements and design of the prototype architecture

Lead partner:	CESGA
Author(s):	Jorge Fernández, Carlos Mouriño (CESGA), Mercedes Riveira, Antonio López (SERGAS), John Stratakis (UoC), Tom Van Herpe (QAELUM)
Work Package:	WP5
Delivery as per Annex I:	M16 (December 31st, 2021)
Actual delivery:	M16 (December 16th, 2021)
Type:	Report
Dissemination level:	Public

Index of contents

Index of figures

Index of tables

Abbreviations

3D	3-dimensional
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
BMI	Body Mass Index
BSD	Berkeley Software Distribution
CBCT	Cone Beam Computed Tomography
CERN	Organisation européenne pour la recherche nucléaire (European Org. for Nuclear Research)
CESGA	Centro de Supercomputación de Galicia (Galicia Supercomputing Centre)
COINS	Collaborative Informatics and Neuroimaging Suite
CMS	Content Management System
CPU	Central Processing Unit
CRUD	Create, Retrieve, Update, Delete
CSV	Comma-Separated Values
CT	Computed Tomography
CTDI	Computed Tomography Dose Index
CTP	Clinical Trial Processor
CTV	Clinical Target Volume
BRICS	Biomedical Research Informatics Computing System
DB	Database
DBMS	Database Management System
DCM	DICOM file format
DERIVA	Discovery Environment for Relational Information and Versioned Assets
DICOM	Digital Imaging and Communication on Medicine
DICOM-RT	Digital Imaging and Communication on Medicine - Radiation Therapy
DKTK	Deutsche Konsortium für Translationale Krebsforschung (German Cancer Consortium)
DLS	Digital Library Software
DMB	Data Management Board
DMP	Data Management Plan
DOI	Digital Object Identifier
DR	Digital Radiography
EEA	European Economic Area
EEG	Electroencephalography
EOB	End of Business
EU	European Union
FACS	Fluorescence-Activated single Cell Sorting
FAIR	Findability, Accessibility, Interoperability, Reusability
FTP	File Transfer Protocol

FUSE	Filesystem in User Space
GB	Gigabyte
GDPR	General Data Protection Regulation
GIF	Graphics Interchange Format
GPL	General Public License
HDFS	Hadoop Distributed File Systems
HIPAA	Health Insurance Portability and Accountability Act
HL7	Health Level Seven
HPC	High Performance Computing
HTML5	HyperText Markup Language 5
HTTP	HyperText Transfer Protocol
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IGRT	Image-Guided Radiation Therapy
iRODS	Integrated Rule-Oriented Data System
ISO	International Standards Organisation
IT	Information Technology
JPEG	Joint Photographic Experts Group
JPG	JPEG file format
JSON	JavaScript Object Notation
kV	Kilovoltage
kVp	Kilovoltage peak
LDAP	Lightweight Directory Access Protocol
LTS	Long-Time Support
LORIS	Longitudinal Online Research and Imaging System
mA	Milliamperage
MEG	Magnetoencephalography
MCNP	Monte Carlo N-Particle
MIT	Massachusetts Institute of Technology
MRI	Magnetic Resonance Imaging
MIRC	Medical Imaging Resource Center
MB	Megabyte
MS<x>	Milestone <x> (in project context)
M<x>	Month <x> (in project context)
nDPA	Swiss Federal Data Protection Act
NIH	US National Institutes of Health
NITRC-R	Neuroimaging Tools & Resources Collaboratory - Registry
NFS	Network File System
OAuth	Open Authorization

OARs	Organs At Risk
ORCID	Open Researcher and Contributor ID
PACS	Picture Archiving and Communication Systems
PBL	Peripheral Blood Lymphocytes
PDF	Portable Document Format
PET/CT	Positron Emission Tomography
PHI	Protected Health Information
PHP	PHP: Hypertext Pre-processor
PTV	Planning Target Volume
RadPlanBio	Radiation Dose Plan and Image/Biomarker Outcome platform
RAM	Random Access Memory
REST	Representational State Transfer
RGB	Red, Green, Blue
ROI	Region of Interest
RFC	Request for Comments
RSNA	Radiology Society of North America
RT	Radiation Therapy
RTSTRUCT	Radiotherapy structure set
SCK CEN	Studiecentrum voor Kernenergie Centre d'Étude de l'énergie Nucléaire
SICAS	Swiss Institute for Computer Assisted Surgery
SERGAS	Servizo Galego de Saúde (Galician Health Service)
SKANDION	Skandionkliniken - Kommunalförbundet Avancerad Strålbehandling
SCO	Świętokrzyskie Centrum Onkologii (Świętokrzyskie Oncology Centre)
SFTP	Secure File Transfer Protocol
SMN	Second Malignant Neoplasms
SU	Stockholms Universitet (University of Stockholm)
UoC	University of Crete
UI	User Interface
UID	Unique Identifier
UJK	Uniwersytet Jana Kochanowskiego (Jan Kochanowski University)
UNIGE	Université de Genève (University of Geneva)
UGENT	Universiteit Gent (Ghent University)
URL	Unique Resource Locator
US	United States (of America)
TB	Terabyte
TCP/IP	Transmission Control Protocol / Internet Protocol
TFS	Teaching File System
VM	Virtual Machine
WED	Water Equivalent Diameter

WP	Work Package
XML	Extensible Markup Language
Y<x>	Year <x> (in project context)

Executive summary

This deliverable presents the software development method of the SINFONIA data repository, focusing on its application during Y1 in order to generate the first version of the repository released in M11 (July 2021).

A brief insight of the purpose of the SINFONIA project is presented first, along with the key objectives that the implementation of a data repository intends to fulfill and correlating them with the relevant work packages of the project and the tasks within those work packages. Then, the general SINFONIA methodology is recalled as the foundation for the software engineering process driving the development of the data repository.

This development process is also described, detailing its steps and the relationships among them. The first step is an analysis divided in two parallel tasks: the capture of the users' needs and requirements and an analysis of those needs and the underlying regulatory constraints; and a survey about existing technologies able to provide features similar to those expected for the SINFONIA data repository, either as finished software products or as development frameworks to compose it.

The second step comprises the design of the prototype architecture. It is composed of a review of the needs, requirements, and constraints in order to compose a model of how the information generated within the SINFONIA project could be organized, and to define a basic architecture for the data repository. Then, the main features and characteristics of the different software solutions reviewed are mapped to all the needs, requirements and constraints identified, and to the design foundations outlined. As a result, Girder, one of the file storage servers studied, arises as the software solution selected to implement the core of the repository.

The prototype was implemented in the third step of the development process. An enumeration of the software tools deployed are given, along with some details about the underlying hardware platform and how the regulatory constraints identified during the analysis step are effectively enforced. The look-and-feel customizations and features extensions applied on the Girder instance deployed are also introduced, supported by an extended description enriched with screenshots of the main features that Y1 prototype offers.

Once the prototype of the repository was implemented and released, a validation process was launched. That process was separated in two consecutive phases, first a WP5 internal phase and then a consortium-level phase. In the internal phase WP5 partners made their own experiments to get familiar with the prototype and to think about new features, improvements, or eventual issues to solve. Some bugs and confusing features identified by the partners, and major improvements and additional features proposed for further iterations, are enumerated. A similar validation process launched for the whole consortium is described too. The whole validation is closed with a self-assessment about up to which extent the prototype implemented addresses the users' needs and requirements, and the regulatory constraints identified.

The document is concluded with a summary of the work covered by this deliverable.

1 Introduction

The main objective of the 4-year SINFONIA project is to develop novel methodologies and tools that will provide a comprehensive risk appraisal for detrimental effects of radiation exposure on patients, workers, caretakers, and comforters, the public and the environment during the management of patients suspected or diagnosed with lymphoma and brain tumors.

One of the key objectives of the project is to develop and operate a repository to collect and pool data from imaging and non-imaging examinations and radiation therapy sessions, histologic results and demographic information related to individual patients with lymphoma and brain tumors. This platform will be used in developing methods and techniques described in other WPs for patient-specific dose estimation, risk assessment and uncertainty determination. It will provide services such as data uploading/downloading, data preview, information search information, or implementation and allocation for execution of AI algorithms. The European (GDPR[1]) and other multi-national regulations on data protection obliges the repository to ensure the anonymization or removal of personally identifiable information. Thus, several guidelines and best practices regarding patient data privacy and protection will be prepared and distributed among project partners. This will enable them to focus on their core tasks since data storing and protection will be managed centrally.

The tasks for achieving the aforementioned aims are included in SINFONIA Work Package 5 (WP5). Specifically, the WP is organized in Tasks 5.1 to 5.3. Task 5.1 will focus on the definition of the platform architecture after gathering the requirements and demands about the repository from the different project partners. This gathering will continue throughout the project in order to improve the platform and release evolved versions thanks to the application of a continuous improvement methodology. Task 5.2 is dedicated to the creation of the platform including data storage, computational capabilities, and protocols according to the requirements gathered in Task 5.1. At the end of Year 1 (Y1, M12), a first prototype will be released, and improved versions are expected to be released every year until the end of the project in Y4 (M48). Task 5.3 will conclude the development with the integration of the features to support AI algorithms. This deliverable, D5.1, titled “Requirements and design of the prototype architecture” and expected for M16 (December 2021), covers the WP5 work in Task 5.1 and the first steps of Task 5.2. Regarding Task 5.1, the content of the document is focused on the procedure for capturing the users’ needs and requirements and the analysis of those needs and the technologies available for achieving a suitable design that sets the foundation for implementing the SINFONIA data repository. However, as the first version of the repository has been released before preparing this deliverable, for the sake of completion the implementation and validation phases of its development process are also covered.

The rest of this document is structured as follows: Section 2 describes the software engineering methodology that drives the development of the SINFONIA data repository, with a special focus to the steps followed to develop the Y1 prototype. Section 3 introduces the method for capturing the users’ needs, summarizes the analysis of their answers and includes an extensive survey of the software technologies available to compose a solution to satisfy such needs. Section 4 covers the design of the repository, defining a theoretical model for organizing the information that is going to be stored, establishing the foundations of an architecture for the prototype, and justifying which technologies from the survey have been chosen to implement the proposed architecture. The hardware and software infrastructure set to deploy the prototype, along with the main features it offers is described in Section 5. Section 6 describes the validation of the prototype performed within WP5, how that internal validation led to an improved version of the prototype, and the external validation open for the Consortium. Finally, Section 7 concludes the document by summarizing its content and advancing some insights on the definition of next iterations of the repository architecture and its implementation. Additionally, some annexes are added for illustrating the process of capturing needs from the repository users: the questionnaire circulated among the partners to collect such needs, and the answers received for that questionnaire, are included as Annex A and Annex B, respectively.

[1] Regulation 2016/679 of the European Parliament and of the Council of 27 April 2016

2 Development methodology

The SINFONIA methodology, defined in the Grant Agreement, Annex A, Part B, Section 1 “Excellence”, Subsections 1.3 “Concept and methodology”, describes the development of a data repository for imaging and non-imaging data and deployment of the developed AI algorithms on the online platform as one of the main topics on which SINFONIA will concentrate to reach its objectives. This topic is initially addressed with a definition of the initial requirements of the platform and the composition and deployment of an initial architecture. A continuous improvement, releasing several versions of the infrastructure, considering constraints and new requirements will be pursued. Finally, the developed AI algorithms will be integrated into the platform. This section describes the software engineering methodology proposed to follow these development guidelines.

2.1 Yearly prototyping methodology

The aforementioned requirements from the Grant Agreement sets the foundations for the software engineering process that must drive the development of the SINFONIA data repository. This process will be based in a prototyping methodology based in the generation of yearly software prototypes.

First, a general analysis of the users’ needs and other external constraints (e.g.: data protection regulations) is performed. A basic subset of these needs is selected to sketch an architecture and implement a first usable prototype after Y1 (M12). This Y1 prototype, which is available since M11 (July 2021) in https://sinfonia.cesga.es, serves as test platform for the users, who are expected to get familiar with it and make their own experiments for detecting problems and proposing improvements. Lessons learned from Y1 version are the input for producing a Y2 version, which will set the foundations of the final version of the repository. Both the initially identified needs and further requirements and improvements eventually expressed by the users will be included in next Y3 and Y4 versions. Thus, version after version, the repository will become a tool not only for internal data sharing but also for disseminating the data generated and the findings obtained from the research activities of the project, including several AI algorithms for predicting radiation effects, with the intention of remaining beyond the project lifetime as a useful tool for the scientific and clinical communities interested in dosimetry.

The yearly prototyping methodology is a hybrid software engineering approach that combines two very common software development models, the rapid prototyping model and the incremental model.

The rapid prototyping model, as described by John Crinnion in [1]:

> *“[…] involves the creation of a working model of various parts of the software at a very early stage and after a relatively short investigation. That model becomes the starting point from which users can re-examine their expectations and clarify their requirements. When this goal has been achieved, the prototype model is thrown away, and the system is formally developed based on the identified requirements.”*

The incremental model, as described by Roger S. Pressman in [2, 3]:

> *“[…] applies linear sequences in a staggered fashion as calendar time progresses […] Each linear sequence produces deliverable ‘increments’ of the software […] Delivers a series of releases, called increments, that provide progressively more functionality for the customer as each increment is delivered […]”*

The idea behind combining these two models is to organize the process with clear development increments in form of usable versions, but at the same time allow a continuous and iterative flow of information about eventual issues and proposal of new or improved features. In the context of the SINFONIA data repository development process, the rapid prototyping method is applied for obtaining the Y1 prototype, considering the first year of a four-year project as the aforementioned “early stage”. The “relatively short investigation” comprises a survey of the needs and requirements of the users and of the different technologies available to implement the platform. The information collected during that investigation is the input for the design and implementation of a first software approach to satisfy those needs. That first software approach is a “starting point” that is submitted for review within the WP5, producing an improved version after that internal review. The result is the Y1 prototype, that is released to provide all the partners with a usable data sharing tool and, at the same time, a mechanism to identify needs, suggestions, and issues that might not be covered yet or not clearly manifested during the users’ needs collection process (i.e., a mechanism to “re-examine expectations” and “clarify requirements”).

The incremental model would drive the development of the repository from Y2 to Y4, as successive releases or “increments” of the SINFONIA data repository will each occur at the end of Y2 (M24), Y3 (M36) and Y4 (M48). The main input to the development process of the Y2 increment of the repository would be all the knowledge acquired during the development of the Y1 prototype, although Y2 increment will be a different, more-evolved product based on its own architecture and with a richer exploitation of current technologies. Let us also remark that despite the Y1 prototype will be eventually discarded, as [1] advances, it will also serve the consortium as the data sharing tool of the project during the development of the Y2 increment. Pressman also considers that “*the process flow for any increment can incorporate the prototyping paradigm”* [3]. In our context, that means that the development of Y2, Y3 and Y4 increments will be driven in a similar way to that of Y1, incorporating both internal and external review processes and mechanisms to collect issues and suggestions.

2.2 Y1 prototype development process

As mentioned in subsection 2.1, the development of the Y1 prototype will follow a rapid prototyping model inspired by the definition from [1]. Here we detail the different steps of that development process, which are depicted in Figure 1:

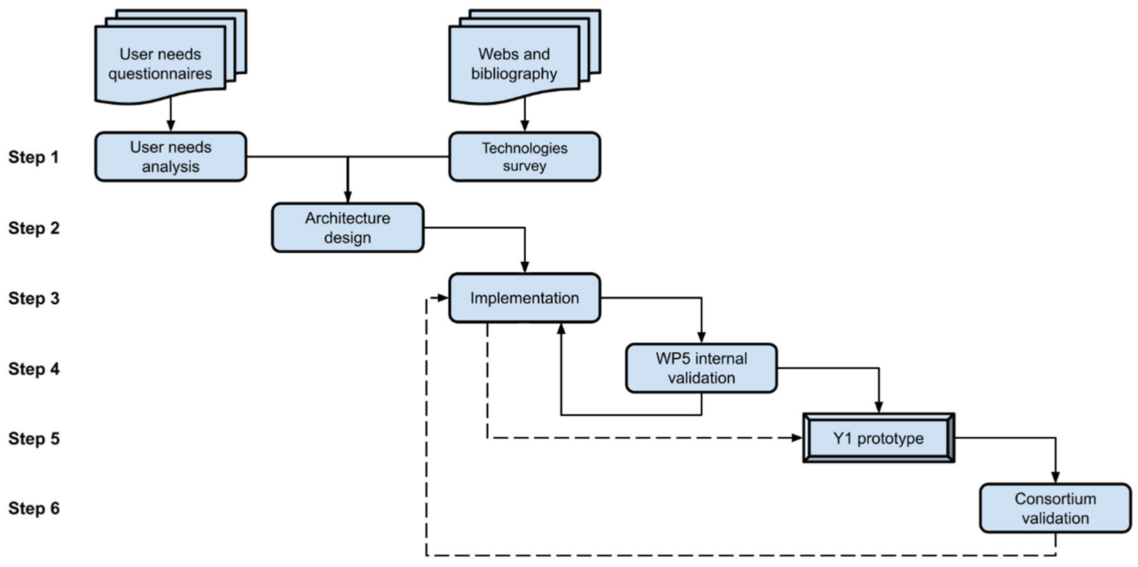


Figure 1: Prototype development process steps

1a. **User needs analysis**: A user needs questionnaire is composed and circulated among the partners. The answers to that questionnaire are analyzed to identify the needs the repository is expected to fulfil, and to select a subset of basic ones to be included in the Y1 prototype.

1b. **Available technologies survey**: A survey of webs, source code repositories and computer science bibliography about the current technologies available for implementing data repositories is performed to indicate the more suitable options for addressing the user needs.

2. **Architecture design**: A design of an architecture for the Y1 prototype is proposed after the conclusions drawn from the analysis of the answers and the information gathered from the survey of available technologies.

3. **Implementation**: The architecture becomes real after a software implementation process.

4. **WP5 internal validation**: Once the implementation from Step 3 is ready, it is presented to the rest of the WP5 members for validation. The feedback given by the WP5 members is considered in order to solve any issues they may find and to apply the improvements they may propose.

5. **Y1 prototype**: After a number of internal validation rounds, the Y1 prototype is presented to the whole SINFONIA consortium and released for its members' use. Those rounds are represented by a loop between "Implementation" and "WP5 internal validation" boxes in Figure 1.

6. **Consortium validation**: After the prototype release, the Consortium members are encouraged to conduct their own individual validation by means of their daily use of the Y1 prototype of the repository. WP5 shall focus on any issues identified by the users or the suggestions they might make. Changes and suggestions within the scope of the prototype design are applied to the implementation and a new updated version is released. This external validation workflow is represented by the boxes connected with dashed arrows in the figure.

3 Analysis of needs and technologies

The first step of the SINFONIA data repository development process consists of two related activities. First, the needs and requirements that users expect to be covered by the repository are collected and analyzed, along with a study of the external regulatory constraint that the platform must observe. From this analysis some basic features are selected to be included in the Y1 prototype. Second, a detailed survey about some available technologies for designing and implementing a valid solution for covering such needs is provided. This section describes, in that order, how those activities have been planned and performed.

3.1 User needs collection and analysis

This subsection describes the method followed to collect from users their needs and requirements regarding the SINFONIA data repository. Namely, those needs and requirements were collected by composing a questionnaire within WP5 and circulating it among partners. The answers received, along with the regulatory constraints that EU-wide projects that manage health information should observe, were analyzed to establish a viewpoint that will guide the design and implementation of Y1 prototype of the repository and set a foundation for the development of the successive yearly increments of the repository.

3.1.1 Requirements capture questionnaire

In the Project Kick Off meeting held in M1 (September 2020) the consortium agreed on circulating among partners a questionnaire to collect the requirements that the SINFONIA data repository should meet. From M1 to M5 (January 2021), WP5 worked in the preparation of a draft questionnaire and the subsequent refinement of both its structure and the specific questions posed on it. After a couple of meetings and a few emails exchanged during those months, a final version of the questionnaire that was more understandable to the rest of the project partners was obtained, having in mind that most of them do not have a deep software engineering background.

The WP5 work during those months resulted in the questionnaire included in this document as Annex A. The questionnaire is composed of an introduction, on which the partners are informed about the purpose of the questionnaire and given some instructions for answering to it; and five sections:

1. Data Summary: The partners are asked about the file types and data formats they expect to upload to the repository, the expected number of elements and size for the different types of data, the identification needs and naming conventions for their data objects, and an estimation of from when and during how long each partner plans to upload information. The questions of this section are numbered as 1.1 to 1.8 in Annex A.

2. Accessing Data: This section intends to capture the upload, download, searching, and visualization needs in terms such as interaction methods, online previewers and editors, external tools for handling data or relevant search criteria and options for the presentation of search results. The questions of this section are numbered from 2.1 to 2.7 in Annex A.

3. Data Security: Here the partners can state their concerns about data anonymization, regarding the regulatory constraints imposed by the EU General Data Protection Regulation (GDPR) and the ethical obligations derived from the informed consents collected from the subjects of their experiments. They are also asked about who should be authorized to access their data, during how long and up to which extent those permissions might me adjustable. The questions of this section are numbered from 3.1 to 3.5 in Annex A.

4. AI Algorithms: For this topic, just a preliminary survey is performed in the form of a few questions about the intention of developing AI algorithms and eventually integrating them into the, as such integration is planned for later versions of the repository. The questions of this section are numbered from 4.1 to 4.3 in Annex A.

5. Other: This section intends to offer the partners a space to provide any other initial requirements for the repository that might not be covered (or just partially) in the previous questions. The answering for such an open question is guided through a couple of tables on which the partners can assess up to what level they consider that some functional[2] and non-functional[3] requirements must be satisfied by the repository. The levels are three, from higher to lower, MUST/MUST NOT, SHOULD/SHOULD NOT, and MAY. Tables for functional and non-functional requirements are numbered as Table A.1 and Table A.2, respectively, in Annex A. The levels are explained in a terminology section just after the tables.

The blank questionnaire was circulated among the rest of the SINFONIA partners on M5 (January 2021). SERGAS and UoC prepared in advance their answers to the questionnaire, that were also circulated in order to give some guidance about the information WP5 was expecting to collect. A deadline of 2 weeks, up to the beginning of M6 (February 2021) was set and, in general, project partners answered in a timely manner.

In parallel to the composition, circulation, and answers' collection of the questionnaire, WP1 worked in the document "D1.2 First Version of the Data Management Plan". This deliverable introduces some insights on the purpose of the data collection within the project, the types and formats of data collected, and some guidelines about how the repository could meet FAIR (Findability, Accessibility, Interoperability, and Reusability) principles [4]. D1.2 also introduces some relevant aspects about security and ethics to be considered; namely, the document states the need for finding a trade-off between data protection and integrity for research usefulness, and for committing to the application of proper de-identification methods in order to ensure local anonymization and thus mitigate the risk for sensitive information leaks. Additionally, D1.2 also includes a detailed survey for each partner about the data types they manage and how the data are used, their needs on identifiability, accessibility, interoperability, and data security, and a summary of the ethical aspects they must consider when processing project data.

3.1.2 User needs analysis

The work of analysis of these two sources of information (partners' answers to the questionnaire and deliverable D1.2) led to the identification of WP-based profiles in the features that the users expect the repository to have. For each profile, we extracted information about the kind of data resources the users handle, which kind of viewing or editing tools they expect in the web interface, and how those resources should be stored in the repository. A general view of the needs of each WP and their common interface requirements are summarized in Figure 2.

As the research conducted by WP2 users is mainly focused on personalized patients' dosimetry in radiological and radiotherapy imaging, PET-CT imaging and radiotherapy, they plan to store DICOM image series generated from several modalities (CT, CBCT, PET/CT, DR…) and different kinds of DICOM-RT objects (images, structure sets, plans, doses), and DICOM metadata. DICOM[4] (Digital Imaging and Communications in Medicine) is the international standard (ISO 12052) for medical images and related information. It defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use. It is implemented in radiology, cardiology, and radiotherapy devices, and it is increasing its stake in medical devices from other domains such as ophthalmology or dentistry. In short, medical devices that implement the DICOM standard generate DICOM files, usually known as instances. A DICOM file can be seen as an envelope that embeds the medical image together with the clinical meta-data associated with this image. Metadata is essentially encoded as a hierarchy of standardized attributes with a name and a value. DICOM-RT was born as the extension of the DICOM standard that defines how radiotherapy data (such as external beam and brachytherapy treatment plans, doses, and images) can be transferred among pieces of equipment [5]. These are the most usual DICOM RT objects: RT Structure Sets, that contain information about delineated structures of interest, such as tumor volume, CTV (Clinical Target Volume), PTV (Planned Target Volume), or OARs (Organs At Risk); RT Plans, that contain

[2] Product features or functions that the developers must implement to enable users to accomplish their tasks.
[3] Design constraints about security, reliability, performance, maintainability, regulations, etc.
[4] https://www.dicomstandard.org/about

geometric and dosimetric data (beam angles, collimator openings, beam modifiers, etc.); RT Images, that specify radiotherapy images usually obtained on a conical imaging geometry; and RT Doses, that contain dose data generated by a treatment planning system. Non-DICOM files may be somehow linked to the DICOM ones, and conventional files like PDF reports or CSV sheets are also expected to be stored. Visualization of DICOM objects through an appropriate DICOM viewer is expected too, along with readers/editors for the aforementioned types of conventional files.

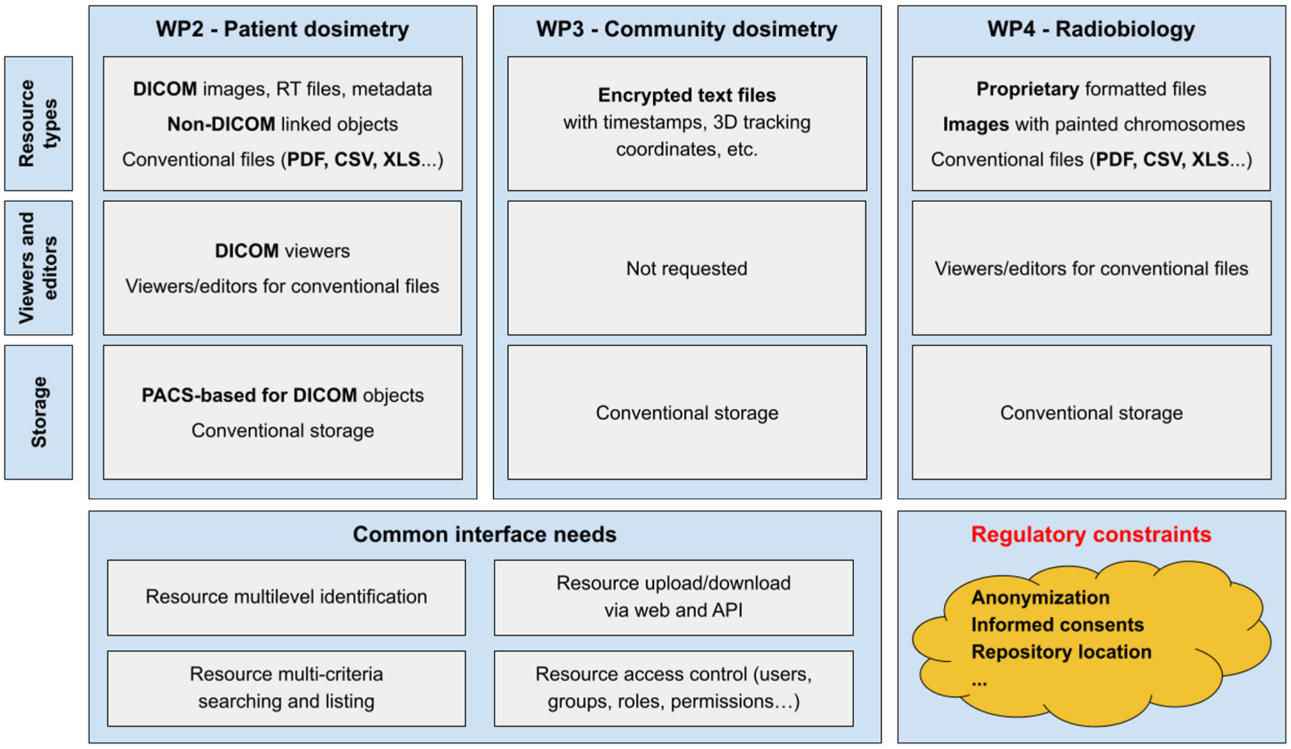

WP2 - Patient dosimetry
WP3 - Community dosimetry
WP4 - Radiobiology
Resource types
Viewers and editors
Storage
DICOM images, RT files, metadata
Non-DICOM linked objects
Conventional files (PDF, CSV, XLS...)
Encrypted text files with timestamps, 3D tracking coordinates, etc.
Proprietary formatted files
Images with painted chromosomes
Conventional files (PDF, CSV, XLS...)
DICOM viewers
Viewers/editors for conventional files
Not requested
Viewers/editors for conventional files
PACS-based for DICOM objects
Conventional storage
Conventional storage
Conventional storage
Common interface needs
Resource multilevel identification
Resource upload/download via web and API
Resource multi-criteria searching and listing
Resource access control (users, groups, roles, permissions…)
Regulatory constraints
Anonymization
Informed consents
Repository location
...

Figure 2: Summary chart of identified needs and constraints

WP3 researchers are focused on community dosimetry activities (staff, comforters, environment). In principle they need a private storage for encrypted text files in CSV or JSON format with tracking data for people and objects, and calculations and measurements of organ and effective doses, and for conventional files like JPEG images of real-time generated dose rate maps or Excel spreadsheets with time series of activity in water and sediment and resulting doses to human or biota.

The radiobiological research[5] performed by WP4 partners relies on laboratory machinery such as the Isis Fluorescence Imaging System from MetaSystems or the sequencing and array-based solutions from Illumina[6], that generates files in their own proprietary formats. They plan to upload images of cells with painted chromosomes, and also conventional files like text reports, PDFs with images of FACS gates and numerical tables, or scoring results as Excel spreadsheets or CSV files. The manipulation of those proprietary formats is linked to proprietary software that might be difficult to integrate into the repository, but readers/editors for the conventional files mentioned will be useful.

On top of these specific research needs, we have also identified a set of common needs regarding the interaction with the SINFONIA repository: resource upload and download through both a web portal (for human users) and an application programming interface (API) to allow other programs, tools or services to exchange data; support of several options for identifying the resources stored (repository internal identifiers, project-level identifiers composed following a common template, partner-private identifiers that users may have assigned before the

[5] https://metasystems-international.com/en/products/isis/
[6] https://www.illumina.com/

upload, etc.); a search engine flexible enough to attend queries based on multiple criteria (resource identifiers, names and descriptions, DICOM tags…) and filter results by those criteria; and a full system of user accounts, groups, roles and permissions, that is needed to keep a proper resource access control.

Table 3.1 summarizes the relevance given by the partners to the functional and non-functional requirements, respectively in subtables a) and b), according to the levels they assigned to them (available in Table B.1 and Table B.2 of Annex B). The relevance has been computed by assigning 3 points to each MUST/MUST NOT answer, 2 points to each SHOULD/SHOULD NOT answer, 1 point to each MAY answer, and 0 points if the partner left that level unanswered, and then summing up the points corresponding to the level each partner gave to each requirement. Since we received 11 filled-in questionnaires the maximum score for a requirement is 33, which would be obtained when a MUST level (3 points) was set for that requirement in all the questionnaires. In general, the needs expressed by the partners in their written answers to sections 1 to 4 of the questionnaire are consistent with the levels assigned to the requirements.

Functional requirement	**Score**
Authorization and authentication	33
Identifiers	28
Data access	28
Metadata	25
Integration / Interoperation	24
User interface	23
Data organization	22
Sharing data in a collaborative environment	20
Availability of fully annotated data	18
Cross-referencing of data	18
User interface (sandbox)	16
Ranking based on metadata	16
Integration of tools developed by other partners	15
Preview	12

a) Ordered summary of functional requirements

Non-functional requirement	**Score**
Capacity / scalability	33
Security	32
Volumetric / performance	30
Compliance with privacy and protection standards	29
Data security	28
Storage needs	27
Human-friendly interfaces	23
Availability/failover	22
Deployment	22
Recoverability	21
Data preservation	21
Serviceability	20
# of simultaneous accesses greater than # of partners	17
Location	16
Test environment	15
Uploading time lesser than 2 minutes/500MB	8

b) Ordered summary of non-functional requirements

Table 3.1: Ordered summary of functional and non-functional requirements

All the datasets that the partners expected to collect and/or generate when WP1 asked them to prepare D1.2 came to an accumulated size in the scale of 10TB. However, from the answers given by the partners for Section 1 (Data Summary) of the user needs questionnaire we can infer just an amount in the scale of 2TB to be uploaded to the repository. As not every component of every dataset described in D1.2 is going to be uploaded to the repository, we will tentatively stick to that lower size of 2TB for estimating the users' needs of physical storage.

The content of this subsection is just a summary of all needs expressed by the consortium during the first months of the project. A similar summary was presented for discussion to WP5 in a periodic meeting held in the beginning of M7 (March 2021). The reader is encouraged to review both the answers to the questionnaire included in Annex B and the content of D1.2 in order to get a full detailed picture of the specific needs of each partner.

3.1.3 Regulatory constraints

As mentioned in the introduction of subsection 3.1, apart from the needs collected from users, there are also regulatory constraints that must be integrated as requirements for development of the SINFONIA data repository. This situation is represented in the bottom right-hand corner of the summary chart from Figure 2.

In the SINFONIA Grant Agreement, Annex A, Part B, Section 5 “Ethics and Security”, Subsection 5.1 “Ethics”, there is an overview about the personal data related issues that the project must address. Namely, it is stated that processing of data will be performed in anonymized format, and that information will be handled confidentially and in accordance with the rules provided by GDPR. Images must be also previously deidentified on partners’ local infrastructure to remove all the patient-identity related information. De-identification processes must be performed before transferring data to any server such as the repository developed in WP5, in order to avoid any mitigation of personal data from the local site to the remote repository. It is also stated that the anonymization process will follow a strategy that must be previously defined by the consortium.

The main origin for these constraints is the Regulation 2016/679[7] of the European Parliament and of the Council of 27 April 2016, on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. The title of this legal text is officially shortened as General Data Protection Regulation and usually known by the acronym GDPR. The GDPR is composed of 99 articles and 173 recitals. The articles constitute the legal requirements, whereas the recitals provide additional information and supporting context that help people and organizations to comply with the articles [6]. There are GDPR recitals that are notably connected to the design, implementation and use of the SINFONIA data repository. For instance, recitals 23 and 24 are about the application of GDPR to non-EU subjects when managing data from EU subjects, recitals 28 and 29 address data pseudonymization, recitals 32 and 33 specify the conditions needed for consenting data processing both in general and for certain areas of scientific research, recitals 34 and 35 set the definitions for genetic and health data in the GDPR context, recitals 53 and 54 specify additional requirements for processing of sensitive data in public and private health, and in social sectors, and whereas recital 159 does the same for processing data for scientific research purposes.

The GDPR affects operations based not only in the EU territory but also in those non-EU states that participate from the European Economic Area (EEA), which are Iceland, Liechtenstein, and Norway. Switzerland (home country of UNIGE partner) is neither an EU nor an EAA state, but it has followed a parallel process for enacting their own updated data protection regulation. The Swiss regulation on this topic, known as Federal Data Protection Act (nDPA), will be entry into force in mid-2022, date which is within the project lifetime. In general terms, it does not go beyond the requirements of the EU GDPR [7], since the latter follows a prohibition approach (no data processing without explicit authorization), whereas the former grants a general permission for data processing "no unlawful violation of the personality" arises [8]. Consequently, we can assume that if the data repository is developed according to GDPR-mandated regulations, it should be also nDPA-compliant.

We have identified three main regulation-derived constraints that must be integrated in the development of the SINFONIA repository. First, the data collected, generated and/or related to the SINFONIA project should not be processed out of the GDPR territorial scope when interacting through the repository. Secondly, data navigation, upload and download must be somehow restricted in those cases on which data owners are mandated to limit the access to their resources. Such control gives rise to an additional requirement as it implies the creation of an account-based authentication and authorization system that collect some basic users’ personal information that must be processed according to a GDPR-compliant Privacy Policy. Regarding the assurance of the proper anonymization of the data resources managed by means of the repository, in principle it is users’ responsibility rather than of repository administrators, but some basic technical guidelines about de-identification of files and images must be prepared to help the consortium to minimize the chance of any personal information leak when sharing data through the repository.

[7] https://eur-lex.europa.eu/eli/reg/2016/679/oj

3.2 Analysis of available technologies

One of the main needs for the SINFONIA partners is the storage and manipulation of different types of medical images and research data files. Because of that, this analysis of the available technologies started with a review of several lists that gather a vast amount of medical image collections and repositories, like:

- Open-Access Medical Image Repositories[8]: a collaborative list open to community contributions maintained by Stephen R. Aylward, Senior Director of Strategic Initiatives of Kitware[9], which is a US company specialized in the research and development of open-source software in the fields of computer vision, medical imaging, visualization, 3D data publishing and technical software development.
- CVonline Image Databases - Bio/Med[10]: a collated list of image and video databases for computer vision research and algorithm evaluation maintained by Prof. Robert Fisher, full professor at the School of Informatics of the University of Edinburgh, Scotland, United Kingdom.
- Neuroimaging Tools & Resources Collaboratory - Registry (NITRC-R)[11]: a free web-based community led by University of Massachusetts Medical School that promotes software tools and resources, vocabularies, test data, and databases for neuroimaging informatics.

The medical image collections and repositories found in the aforementioned lists are supported by a number of software architectures and programming frameworks and languages that may be useful for implementing the SINFONIA repository or, at least, to inspire the actual implementation of some of its functionalities. Most of these repositories are instances of different server-based platforms developed for sharing and describing data, analyses, and other contents.

Some of these platforms are devoted to store medical imaging data, whereas others provide support for the whole lifecycle of data, from experiment designs to the publication of results. Some cover a more general scope and are designed for no specific field, and others are tailored to disciplines like neuroimaging. Moreover, references to full or minified software for managing PACS (Picture Archiving and Communication Systems), and to repositories based on Digital Library Software were found in those lists, so that a mention to them is deserved in this analysis of available technologies.

3.2.1 Medical imaging repositories and platforms

The SICAS Medical Image Repository[12] is a medical image repository run by The Swiss Institute for Computer Assisted Surgery (SICAS) for its founding members and other related institutions. It allows its users to set a folder-based flexible structure for their information, rather than a fixed project-based organization. The resources may be searched by means of semantic predicates and protected under access levels and permissions. It also offers a full featured API to connect to the repository from any external software. On top of the platform features, it provides additional services like anonymization of patient information and image features or expert curation of high-quality image data. A demonstration portal is available, but accounts for using the repository are only provided to affiliated institutions. The source code of the platform is not publicly available and there is no information about the underlying programming languages and libraries.

ePAD[13] is a free imaging platform developed by the Rubin Lab at Stanford Medicine Radiology at Stanford University. One of its main objectives is to encourage the adoption, collection, dissemination, and use of quantitative imaging data among radiologists, in order to provide the community with the benefits of image

[8] https://www.aylward.org/notes/open-access-medical-image-repositories
[9] https://www.kitware.com/
[10] https://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm#biomed
[11] https://www.nitrc.org/
[12] https://www.smir.ch/
[13] https://epad.stanford.edu/

metadata sharing and interoperability in their clinical workflows. Its architecture is based on a core that can be extended with community-developed plugins, making it flexible for supporting a wide range of imaging-based projects. These plugins extend the platform with features like image fusion and subtraction, federated machine learning or generation of segmentation objects from contour annotations. The platform can be downloaded for installation after accepting its terms of use or compiled from the Java core and the Java and JavaScript interfaces available in GitHub[14].

XNAT[15] is an open-source platform developed by the Neuroinformatics Research Group at Washington University that can be used to support a wide range of imaging-based projects. It supports a core set of functional data tasks such as uploading, organization and sharing, viewing, and downloading, securing, anonymization and permission management, searching and exploring, and running complex computing processes. XNAT has been proven to support a vast number of use cases [9]: it can be deployed as an institutional repository, for storing clinical research data, for clinical translation, for sharing data, or for conducting multi-site studies. It is mainly written in the Java programming language with an underlying PostgreSQL database, and it offers both web and API interfaces through an Apache Tomcat server. It can be installed in both virtual and physical machines on local premises.

COINS[16] (Collaborative Informatics and Neuroimaging Suite) is an open-source information system developed by The Mind Research Network and the Lovelace Biomedical and Environmental Research Institute, two non-profit research institutions, and the University of New Mexico. It is composed of web-based tools to manage studies, subjects, imaging, clinical data, and other assessments [10]. It provides tools for creating and managing questionnaires and images, with DICOM support, as well as EEG and MEG sessions. Flexible user management and roles allow administrators to include new users easily and create a collaborator portal for each in order to place their metadata and study documentation there, like radiology review reports. It also includes a set of HTML5/JavaScript search, export and sharing tools [11], with specific safeguards for preventing PHI leaks. Their maintainers claim that COINS hosts more than 800 studies of institutions from all over the world. It is offered as a cloud-based service, and despite having a GitHub page, neither the source code of the platform main components nor instructions to deploy a private instance seem to be publicly available.

LORIS[17] (Longitudinal Online Research and Imaging System) is an open-source web-based data and project management software for neuroimaging research studies. Developed by several research groups from the McGill University in Canada, it provides storage and processing tools for behavioral, clinical, neuroimaging and genetic data for both longitudinal and large multi-site studies [12]. Its key features include data collection with support for a variety of sources and formats including DICOM, querying and dissemination tools for data sharing, workflows for verifying clinical and imaging data, workflows and integrated 3D visualization and quality annotation tools for neuroimaging, genetic and epigenetic data, or a module for a secure online questionnaire completion from home for study participants. Its open-source nature and a user-friendly configuration module allow researchers to add custom features and to tailor the platform to any study design. The platform is programmed mainly as a PHP/JavaScript web application. The source code is available in GitHub and instructions to deploy a LORIS instance are provided by the maintainers.

RadPlanBio[18] (Radiation Dose Plan and Image/Biomarker Outcome platform) is a platform developed in the German Cancer Consortium (DKTK, Deutsche Konsortium für Translationale Krebsforschung) to provide their non-commercial radiotherapy trials unit with a web-based software infrastructure to operate. It supports the collection and exchange of radiotherapy research data from multi-center clinical and preclinical studies, with extensions for securing PHI information and uploading DICOM images and treatment plans [13]. Other important

[14] https://github.com/rubinlab
[15] https://www.xnat.org/
[16] https://coins.trendscenter.org/
[17] https://loris.ca/
[18] https://dktk.dkfz.de/en/research/core-facilities/radplanbio

features are study subject randomization, support of electronic case report forms for clinical data collection, linking of DICOM images with DICOM-RT treatment plans, DICOM data de-identification and RTSTRUCT ROI naming harmonization, or a DICOM viewer with a DICOM-RT plugin. Its Java codebase is available under a GPL-3.0 License in GitHub[19], along with additional tools for connecting the platform to a PACS or a desktop application for uploading DICOM files.

The MIRC[20] (Medical Imaging Resource Center) project of the Radiology Society of North America (RSNA) intends to provide the radiology community with open-source medical imaging tools. MIRC software lets radiology departments and research laboratories create, manage, and share data libraries, storing and serving all kinds of digital information like teaching files, clinical and technical documents, electronic presentations, and imaging datasets for research and clinical trials. Its main components are the Clinical Trial Processor (CTP) and the Teaching File System (TFS). The CTP intends to facilitate the aggregation of research data between research sites, also including several pipelines to process imaging datasets in order to prepare them for research usage, such as personal information removal. TFS instances can run in a distributed manner in different research sites and they can be exploited in multiple ways: as a whole for indexing, searching and storing data, and authoring a radiology teaching file; as an additional layer on top of other existing teaching file system with their own internal database to enable it to respond to MIRC queries; and, finally, just connected to any other commercial teaching file system that supports the MIRC query mechanism. All the software is released under the RSNA-MIRC Public License[21], that allows anyone to use, reproduce, and distribute the original software and any derivative work. The different components and tools are distributed as Java packages to be run in Java Virtual Machines installed in a server or virtual machine. The aforementioned packages contain the Java source code of those components and tools.

3.2.2 Scientific data repositories

BRICS[22] (Biomedical Research Informatics Computing System) is an extensible web-based platform to implement collaborative web-based bioinformatics repositories. The platform was born from the joint effort of several US public institutions, under the coordination of the Center for Information Technology of the National Institutes of Health (NIH). Repositories implemented with BRICS can be used for collecting, validating, harmonizing, and analyzing datasets from research studies and clinical trials. The platform is structured in a modular architecture that intends to cover all the stages of the data lifecycle, and that can be customized to meet specific research needs and objectives. Apart from a graphical web interface, it also offers an API for programmatic import and export of data. Despite BRICS is a public-funded development, there is no information about the license source or the source code availability.

DERIVA[23] (Discovery Environment for Relational Information and Versioned Assets) is a platform developed by the Informatics Systems Research Division of the Information Science Institute of the University of South Carolina. It intends to integrate lightweight and user-friendly tools to capture, manage, and curate data throughout the full lifecycle of scientific data (experiment designs, data acquisition, analyses, publication). It allows users to store diverse scientific assets, organize them by means of metadata models, retrieve them quickly and easily, and enforce access controls as needed. The architecture of DERIVA is structured in three levels (server components, client libraries and client applications) which can be composed in multiple ways in order to adapt to changes in the data models or interface-related needs (web and desktop applications, command line access, API for programmatic accesses, etc.).

[19] https://github.com/ddRPB/rpb
[20] http://mircwiki.rsna.org/index.php?title=Main_Page
[21] http://mirc.rsna.org/rsnapubliclicense.pdf
[22] https://brics.cit.nih.gov/
[23] http://isrd.isi.edu/deriva/

Synapse[24] is a collaborative research platform created by Sage Bionetworks and funded by several institutions of the US National Institutes of Health for promoting reproducible research and responsible data sharing throughout the biomedical community. As usual in such platforms, it allows researchers to share and to describe their data and the analyses performed on them, where they come from, and how to use them. It also provides mechanisms for adding and retrieving data, analyses, and their respective descriptions. This is achieved by storing metadata about the content, descriptive wiki pages, and provenance. The actual data is not stored by the Synapse instance, but it can be configured to store data in many types of locations, from local hard drives to private servers and cloud storage (remote URLs, SFTP servers, AWS S3, etc.). The developers and funding institutions of Synapse keep a central instance of Synapse with the philosophy of maximizing collaboration within the research community, although individual users are always allowed to control up to what extent their resources are shared. Nevertheless, as a public-funded work it is released under the Apache 2.0 License and the source code is available on GitHub. The core of the platform is programmed in Java, its web interface is written in JavaScript, and client libraries are available for multiple programming languages (R, Python, etc.) in order to allow programmatic access to Synapse instances.

IEEE DataPort[25] is a cloud-based data sharing service promoted by the Institute of Electric and Electronic Engineers (IEEE) for storing, searching, accessing, and managing data. It intends to make datasets easily accessible for readers of their research papers: the datasets may be linked to IEEE Xplore articles and to the ORCID asset list of their authors or curators, Digital Object Identifiers (DOI) may be provided automatically for each dataset and each analysis stored, and citations are formulated and provided in multiple formats for users. Moreover, it also facilitates data analysis by enabling access to data from AWS Cloud and by enabling the downloading of datasets. It is a subscription service (free for individual users, whereas institutional subscribers are charged depending on their size) that cannot be downloaded and deployed in a private server or virtual machine, so no information about the source code and its license and availability is provided.

HUBZero[26] is an open-source software platform born at the Network for Computational Nanotechnology with the sponsoring of the National Science Foundation, for building collaborative websites for hosting analytical tools, publishing scientific data, and sharing resources. Use cases of HUBZero instances go across an important number of scientific disciplines, including cancer research, pharmaceuticals, biofuels, microelectromechanical systems, climate modelling, water quality, volcanology, and more. The core of HUBZero is a CMS (Content Management System) that allows members to write blog entries, participate in discussion groups, work together in projects, publish datasets and computational tools with digital object identifiers (DOIs), and make these publications available for others. Researchers using a HUBZero instance may create datasets and interactive simulation tools with RStudio, Jupyter Notebooks, and other webapps. Spaces for discussion, progress tracking, and file sharing may be also provided by means of services such as Google Drive, GitHub, or Dropbox. Simulation/modelling tools hosted in a HUBZero instance can run on cloud computing resources, clusters, and other high-performance computing (HPC) facilities, and generate attractive visualizations. HUBZero production instances must be installed on tailored Linux physical servers with hardware resources enough, although a VirtualBox appliance is provided for development purposes. The source code is mainly written in PHP, which is consistent with its web-based nature, and it can be accessed in the GitHub[27] of the project. Most packages are released under the MIT License, but some remain licensed under the GPLv2 license or other open-source licenses as derivatives from other works.

The Integrated Rule-Oriented Data System (iRODS)[28] is open-source data management software used by research, commercial, and governmental organizations, with a special orientation to mission critical distributed environments. It is indeed a middleware that sits between the file systems that actually store data and the domain-

[24] https://www.synapse.org/
[25] https://ieee-dataport.org/
[26] https://hubzero.org/about/aboutus
[27] https://github.com/hubzero/hubzero-cms
[28] https://irods.org/

specific applications that exploit data. All the servers in an iRODS deployment must run the same code and acknowledge each other as peers. The middleware can be deployed on top of an existing heterogeneous data infrastructure in order to construct a flexible data grid. Data becomes virtual, allowing access to distributed storage assets regardless of where and on what device the data is stored. Data discovery is enabled by a metadata catalogue to describe data objects and collections. Data workflows are automated by means of a rule engine that permits any action to be initiated by any trigger on any server or client within the platform. Thanks to its plugin-based architecture, it also supports features like microservices, multiple kinds of storage resources, authentication mechanisms, network protocols, rule engines, extension of API endpoints, and databases. Secure collaboration mechanisms between collaborating or distributed teams are provided too.

The Midas[29] platform is an open-source toolkit developed by Kitware, Inc. for rapid creation of web-enabled data storage for computational scientific research. Repositories implemented with Midas are able to provide multiple data access methods: web, remotely mounted filesystems, and DICOM server interfaces, and allows developers to extend the methods in which data is stored to other relational and non-relational databases. Midas is optimized for centralizing, indexing, storing, and processing massive collections of data, integrating them seamlessly into the workflow of both individual research projects and large publication efforts. It also provides a self-documenting API and a Python client library for accessing data programmatically through HTTP requests or scripts. The Midas platform has been superseded by Resonant[30], a more modern toolkit that inherits and extends most of the features of Midas for providing storage, analysis, and visualization solutions.

3.2.3 File storage servers and clouds

Dropbox[31] and Google Drive[32] are two very popular services of cloud-based file storage and synchronization, with both offering several modalities of paid subscriptions (Dropbox Business, Dropbox Enterprise, Google Workspace) for professional uses at several scales regarding the number of members within the subscription and the amount of data managed through it. Their core features are quite similar, as both platforms store the uploaded files in their respective clouds and offer mechanisms for file synchronization across devices and for file sharing with other users (either registered or anonymous, depending on the specific preferences set by the owner of data). Users can access to their files by means of client applications installed in their devices (computers, smartphones, etc.) and of the respective web portals of the platforms. Both Dropbox and Google guarantee that all the data managed through their paid-subscription cloud services is processed according to GDPR[33,34] and high-level standards on security and privacy regarding health information like the Health Insurance Portability and Accountability Act[35,36]. As mentioned before, these services are only cloud-based and they cannot be deployed on local computing premises, so that users have to trust in the privacy and security measures established by the providers and in their observance of applicable regulations.

ownCloud[37] is an open-source software suite for file synchronization and sharing, and content collaborative edition. The file server can be deployed in private data centers, at common cloud storage providers, or in a Germany-hosted collaboration platform controlled by ownCloud GmbH, the company that develops and maintains the software. Users can find, edit, and share documents on mobile apps, through a web portal or by means of a desktop client. The data storage and maintenance are GDPR-compliant[38] and security is enforced through

[29] https://web.archive.org/web/20200929102106/http://midasplatform.org/
[30] https://resonant.kitware.com/
[31] https://www.dropbox.com/
[32] https://www.google.com/drive/
[33] https://aem.dropbox.com/cms/content/dam/dropbox/www/en-us/security/privacy_data_protection_whitepaper_04-2020.pdf
[34] https://workspace.google.com/terms/dpa_terms.html
[35] https://www.dropbox.com/en/business/trust/compliance/HIPAA
[36] https://support.google.com/a/answer/3407054
[37] https://owncloud.com/
[38] https://owncloud.com/gdpr/

measures like multi-factor authentication, encryption, and file lifecycle management (users can set retention periods after which files are automatically either archived or deleted if they are no longer needed). It also offers a reinforced sharing method for confidential documents that allow users to set limitations on files and folders like preventing copying, downloading, and editing. Basic features are supported for free, with the source code available for download and installation in a private server and for collaboration in GitHub[39] under the GNU Affero license. However, commercial usage (both in private servers and the ownCloud hosting) with additional features and support services requires a paid subscription billed monthly in function of the number of users.

Nextcloud[40] is another software suite with similar features whose core was forked from ownCloud in 2016. It offers very similar features to those of its ancestor, with free basic and paid enterprise modalities. The source code is also available in GitHub[41] under the GNU Affero license. One of the differentiating traits of Nextcloud with respect to ownCloud is a very active ecosystem of community-developed apps and plugins[42]. For instance, there is a DICOM viewer[43] that allows users to display and manipulate DICOM images with a streamlined sidebar and viewer seamlessly integrated in the Nextcloud web interface [14].

Girder[44] is the web-based data management tool from Resonant. It enables a quick and easy construction of web applications for data organization and dissemination. Girder offers a single RESTful web API behind which data is transparently stored, served, and proxied from heterogeneous backend storage engines, including local filesystems, MongoDB databases, Amazon S3 stores, and Hadoop Distributed File Systems (HDFS). It supports multiple options for managing and authenticating users, from self-stored credentials itself to third-party services such as OAuth or LDAP. Both user- and role-based access controls can be set on the resources managed by a Girder instance. Along with the Girder server, which is written in Python, a JavaScript single-page web application can be built and served to browsers in order to allow users to interact with the Girder instance. This application relies on the aforementioned RESTful web API. Despite being part of a tool more oriented to data analysis and visualization, all these features make Girder resemble more to a file storage server than to a scientific data repository. Additionally, the Girder core behavior can be modified and extended by means of plugins. For example, there is a DICOM plugin available that makes Girder capable of parsing metadata from DICOM files in order to index them, to allow the search engine to inspect DICOM metadata and to preview the images of several types of DICOM series in the web application. The source code of Girder is available in GitHub[45] under the Apache 2.0 License, for downloading and installing it in private servers with all the features available. Additionally, Kitware offers a paid support service for solving issues and questions and develop prototypes.

3.2.4 PACS and DICOM servers

The improvements on imaging computer technology during the last decades led to an important decrease in usage of analogue film-based systems in both radiology clinical departments and research laboratories. Such physical support was gradually replaced by digital images that must be efficiently stored, rapidly retrieved, captured from multiple acquisition modalities, and simultaneously accessed from multiple sites. Electronic picture archiving and communication systems (PACS) were developed to meet these needs [15]. Such a system consists of image acquisition devices, software for data management, storage devices, the transmission network, computers for user interaction, and devices to produce hard-copy images.

As mentioned in section 3.1.2, DICOM is the international standard for medical images and related information. Hence, an important stake of the files stored and indexed in a PACS are DICOM files acquired from a variety of medical imaging devices. DICOM files are one of the main data assets that SINFONIA partners manipulate and

[39] https://github.com/owncloud
[40] https://nextcloud.com/
[41] https://github.com/nextcloud
[42] https://apps.nextcloud.com/
[43] https://apps.nextcloud.com/apps/dicomviewer
[44] https://girder.readthedocs.io/en/latest/
[45] https://github.com/girder/girder

also expect to share through the repository. Because of that, a review about up to what extent PACS software may meet their needs seems convenient.

PACS make sense in the context of large hospital departments and research centers, and because of that their commercial implementations may exceed both actual needs and budget of smaller groups, which are devoted to specific research tasks. Moreover, some of their components such as the DICOM viewers they may include have to be certified as medical devices in order to integrate them in the patient care workflow of a medical institution. Such requirements increase the complexity of the underlying software engineering process and, thus, the final cost of these systems. However, there are other pieces of software called DICOM servers or minified PACS that try to cover some PACS' basic features but focusing on the management of DICOM files and avoiding the constraint of having to be certified as medical devices.

Orthanc[46] is an open-source lightweight implementation of a DICOM server for healthcare and medical research. It is rooted in the Sébastien Jodogne's research work [16] at the Department of Medical Physics of the University Hospital of Liège (Belgium). It was born with supporting of research about automated analysis of medical images in mind, although nowadays it is also integrated in the DICOM workflow of many healthcare institutions throughout Europe and the United States [17]. It has a quite simple architecture and can be run with neither complex database administration nor the installation of third-party dependencies, as by default it deploys seamless its own lightweight database engine. It also provides a RESTful API that allows users to access DICOM data by means of any programming or scripting language able to make HTTP requests. This API is also used by Orthanc Explorer, an out-of-the-box web interface served to navigate through the data stored in the system. The tags of the stored images can be easily downloaded, and standard PNG previews can be generated on-the-fly. Additionally, it features a plugin-based extension mechanism to add modules like more DICOM web viewers, backends for PostgreSQL/MySQL databases and cloud storage services, or an implementation of the DICOMweb standard. It is a cross-platform software packaged to be installed in Linux, Windows, and OS X systems. As an open-source project, its C++ source code is available in a self-hosted Mercurial repository[47] under a GPLv3 license. Nowadays Orthanc is developed and maintained by Osimis[48], a spin-off company of the University Hospital of Liège that provides paid support services according to an open-source business model.

3.2.5 Digital Library Software

Another way to implement such repositories is by means of Digital Library Software frameworks. Digital libraries are online databases of digital objects that can include text, images, audio, video, and other digital documents or media formats. In addition to storing content, digital libraries provide means for organizing, searching, and retrieving the content contained in the collection. There are many examples of software solutions to implement this kind of repositories, like DSpace, E-Prints, Fedora Commons, Greenstone, Invenio, MyCoRe, Opus, SimpleDL or SobekCM.

For instance, the Invenio Framework[49] is an open-source project initially developed by CERN as a code library to build large-scale repositories with support for API-based interaction, multiple storage backends and an authentication system. It is the base for other two software solutions: InvenioRDM for generic institutional repositories, and InvenioILS for the digital management of libraries, with support for cataloguing, circulation, acquisitions, and interlibrary loans. All these tools are programmed mainly in Python with web interfaces in HTML5/JavaScript, and they are available under MIT license in the GitHub[50] page of Invenio Software. The

[46] https://www.orthanc-server.com/
[47] https://hg.orthanc-server.com/
[48] https://osimis.io/en/
[49] https://inveniosoftware.org/products/framework/
[50] https://github.com/inveniosoftware

Invenio tools have been the base for implementing Zenodo[51], a very-well known platform for sharing and publishing papers and related research assets.

DSpace[52] is another open-source solution for implementing institutional repositories. It was initially released in 2002 as a joint initiative of Hewlett-Packard and the MIT. Operations of administration, deposit, ingest, search, and access are performed through a web interface. The asset store is built on top of a filesystem, whereas the metadata, including access and configuration information, is stored in a relational database. It also offers a granular group-based access control that allow resource owners to set permissions down to the level of individual files. The source code is mainly written in Java, with its web interfaces in HTML5/Angular, and it is available under a BSD license in their GitHub[53] repository. Institutions like The World Bank, University of Cambridge, or the MIT use DSpace to implement some of their institutional repositories. There has been also an attempt to build a DICOM medical image repository [18] on top of DSpace. However, and despite the highly customizable nature of DSpace, the proposal from [18] is still a nascent approach that required very specific modifications in order to support the storage and classification of DICOM files.

[51] https://zenodo.org/
[52] https://duraspace.org/dspace/
[53] https://github.com/DSpace

4 Design of prototype architecture

This section details the process followed for designing the architecture for the Y1 prototype. First, a quick review of the users' needs and regulatory constraints identified in section 3.1 is performed in order to compose an information model for the prototype. After that, a very basic architecture able to support the needs and constraints and the information organization described in the model, is proposed. Finally, the different technologies analyzed in section 3.2 are reviewed in order to check up to what extent they would be suitable for implementing the prototype, selecting Girder as the best candidate and introducing a description of its features.

4.1 Information model

A quick review of the user's needs concluded in section 3.1 is performed in order to clearly identify the actors that participate on those needs, how they interact, and the kinds of data objects they actually produce during their research activities. The result of this process is a theoretical model about how the information generated within the SINFONIA project (and thus, is expected to be stored in the repository) might be organized. The diagram in Figure 3 intends to represent this theoretical information model.

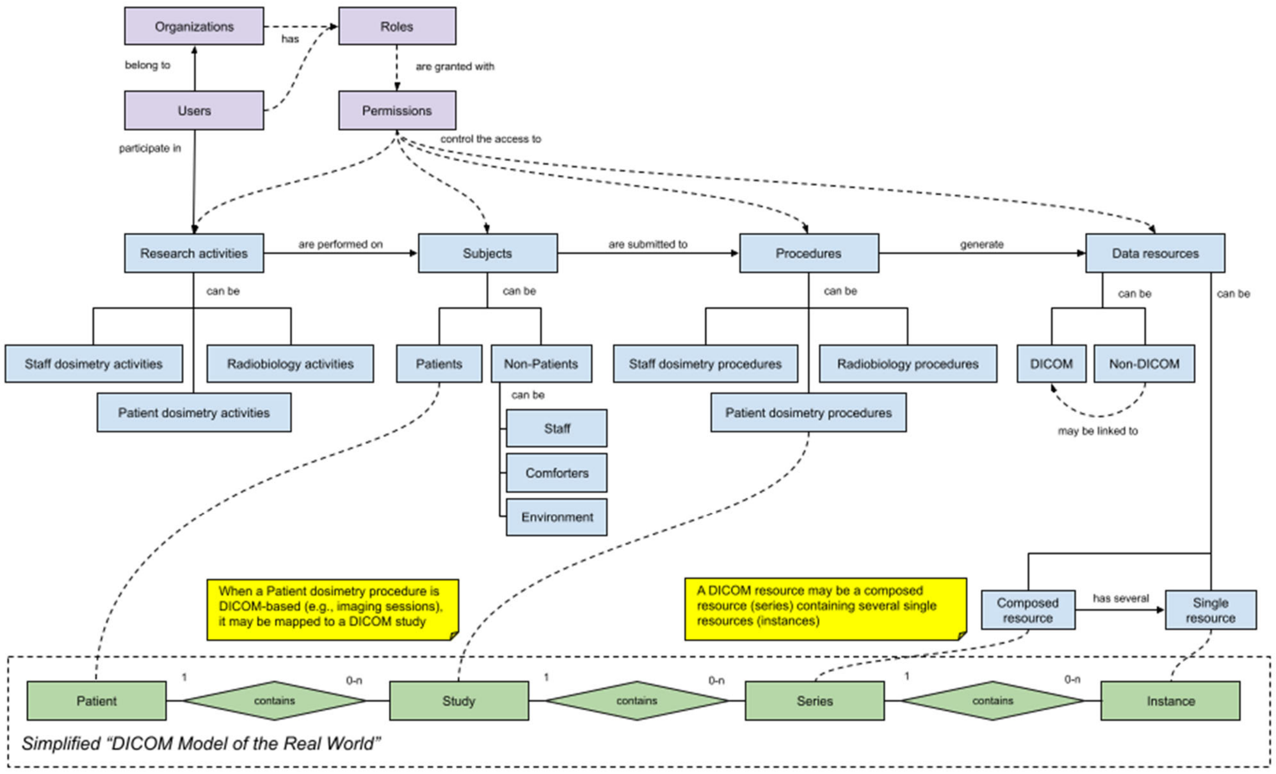


Figure 3: Information model diagram

Each WP conducts its own type of **research activities**, which may be grouped in patient dosimetry activities for WP2, staff dosimetry activities for WP3, and radiobiology activities for WP4. All these research activities are performed on **subjects**: **patients** that undergo RT studies and treatments, **non-patient** hospital staff that are monitored to know the radiation dose they receive when conducting those studies and/or administering those treatments, and **subjects** for which tissue samples are being studied in terms of sensitivity to radiation. Within these activities, subjects may undergo several **procedures**, e.g., a pre-treatment imaging study on a RT patient, measurement of daily radiation dose on a hospital staff member, search of mutations on PBL from a SMN patient, etc. Multiple **resources** are generated from those procedures, and these resources can be DICOM (imaging series, RT structure sets, RT plans, etc.) and non-DICOM (spreadsheets, PDF documents, conventional images, files with proprietary formats). Such kinds of resources might have to be linked to each other if needed. All these information entities can be seen in light blue boxes in the middle section of the information model diagram from Figure 3.

Data entities come from specific partners within the project that might have their own access policies. Thus, the partners should have the possibility of managing which repository users may handle their resources and up to what extent. Such a control is based on **users** from **organizations** (partners) being given different **roles** and granting those roles with different **resource handling permissions**, which is a commonplace in the design of data repositories. This is the so-called “user management”. These components are represented in purple boxes at the top of the information model diagram from Figure 3.

As an important number of users are going to manage DICOM files, we are also outlining a mapping among our information model and a simplified version [19] of the **DICOM Model of the Real World** [20] implemented by PACS like Orthanc: both models have **patients**, RT procedures may be **studies** that generate DICOM resources, and these resources may be either **series** *composed* of multiple **instances**. These components are represented in green boxes at the bottom of the information model diagram from Figure 3.

Given that the repository must store both **DICOM resources** that are usually **series of instances**, and **non-DICOM resources** that are usually **single conventional files**, we propose a *composite* approach to reflect that dual nature of resources that may be *single* and *composed* at the same time. In this way, for instance, a single instance of a CT series will be a *single resource* in our system, but at the same time the whole series will be considered as a *composed resource* too.

4.2 Y1 prototype architecture

Once the needs and requirements that the SINFONIA repository is expected to cover were identified and the information flow is modelled, a basic architecture for the Y1 prototype is defined. This basic architecture is represented in the block diagram depicted in Figure 4. As an architecture for a prototype, it intends to set the foundations for a functional first version of the repository, rather than for a complex piece of software integrating each specific need stated by each partner in the user needs analysis phase.

The entry point for users into the repository will be either the web portal or the REST API. Both are web services, so they need to be supported by a type of web application framework able to build and serve web pages and process user requests. These requests need to be directed to a resource management middleware, this is, to the underlying instance of a software solution able to index (usually by means of a database), store and serve the different types of files and metadata that SINFONIA users expect to manage through the repository. All these components should be behind an HTTP server, through which they are exposed to the Internet. Finally, all the repository infrastructure would rely into a server connected to either a local or a network-attached physical storage on which the data resources indexed by the middleware are going to be effectively kept.

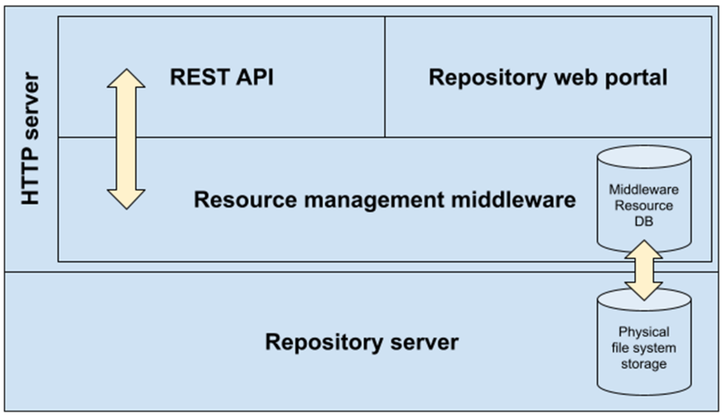


Figure 4: Block diagram of the Y1 prototype architecture

4.3 Mapping needs and design to the available software

The key point for implementing an actual prototype after this architecture is to choose a technology capable of supporting a functional repository (i.e., a tool that allows users to start sharing data and to identify new needs and improvements for next versions), but at the same time not too complex in order to avoiding complicated hardware and software deployments or extensive customizations that may derive in an undesired prolongation of the prototyping process beyond the Y1 planned deadline. Most of the technologies described in Section 3.2, either alone or somehow combined, fit into the generic architecture defined for the prototype. However, a significant proportion of them were virtually discarded after a first sight due to their large scale, their specificity and/or complexity of their features, their extensive hardware and software requirements or the unavailability of full freeware and/or open-source versions.

Both the medical imaging and scientific data repositories are tools extensively used by universities, research centers and medical departments all over the word. Most of these tools seem to cover to different extents some of the needs expressed by the SINFONIA partners. However, most of them are also designed for very specific research fields (such as neuroimaging) or tasks (full management of clinical trials and research studies), and this may lead to a “golden hammer” problem if used: the repository would be based on one of them just because of its extensive features close to the partners’ needs, but probably partners should also have to adapt their workflows and organization of the information to the tool’s preexisting design. Moreover, some of these tools are so complex that a prototype produced after them would have a very steep learning curve, which would ruin the dynamic nature of the prototyping development model. Additionally, some of these tools are also composed of multiple software components that need to run in demanding hardware platforms with very specific requirements. We must concede that a number of them could have been tested in order to confirm all these drawbacks concluded after our features review, but such a study would have also notably extended the technology selection step and consequently affect the timely implementation of the Y1 prototype.

File sharing clouds like Google Workspace and Dropbox Enterprise are very well-known solutions provided by very prominent actors in their sector. Both services are claimed to comply extensively the GDPR and PHI standards such as the US HIPAA. However, as cloud-only services, data will be stored in external servers not directly controlled by a project partner if they were used. Additionally, major features for enterprise users are only available on high subscription fees. File sharing servers like Nextcloud and ownCloud are claimed also to be implemented in order to provide services according to GDPR and other PHI standards, with the advantage that they can be indeed installed in a private server. However, their deployment is complex and quite resource-demanding, and as for the file sharing clouds, the versions including enterprise features are only available for local installation under subscription.

DICOM servers would be an interesting solution for WP2 researchers, since they manage and manage lots of DICOM and DICOM-RT objects from many modalities and IA algorithms. However, DICOM servers are not the best option to index, organize, and store non-DICOM files, which could lead to an insufficient support of the workflows of the rest of SINFONIA WPs.

After this review of the main advantages and drawbacks found on the different tools reviewed in section 3.2, we found in Girder the middle-ground option. It is meant to enable quick and easy construction of web applications that need to manage data that are dynamically provided by users of the system or exposed through external data services.

The features it provides are very close to those of file clouds, which allows users to store files of any type and to freely organize them in custom folder hierarchies. Support for indexing, previewing and tag-based search of DICOM files is also provided thanks to the DICOM plugin. Despite having included an explicit reference to a simplified version of the DICOM Model of the Real World in our information model (see Figure 3), for the moment we prefer to stick to a simplified architecture with just a component for implementing the resource management middleware, instead of combining Girder for with a DICOM server like Orthanc to provide an extended support for DICOM files management. We keep this idea in mind for further iterations of the repository. The installation of Girder is quite simple for an average system administrator, as it based on an IT tasks' automation script written in Ansible[54] that composes a proper Python environment, installs the MongoDB database instance that Girder uses to index data, generates the JavaScript web application, runs the Python server code, and launches the bundled server to start working with the repository from a browser. This server also exposes the Girder API. Finally, let us also remind that Girder is a full open-source project actively maintained by their developers from Kitware.

4.3.1 Main features of Girder

Let us describe in this subsection the main characteristics and features of Girder, as they will be the basis for implementing the main features provided by our repository.

The core of Girder is able to store, serve, and proxy data through a single RESTful web API that minimizes coupling between the backend services and the frontend clients. This API is mostly used to interact with resources that are represented by models in the system. Models internally interact with a MongoDB database to store and retrieve persistent records. The models contain methods for creating, changing, retrieving, and deleting those records. The core Girder model types are Users, Groups, Collections, Folders, Items, Files and Assetstores.

Users are gathered in Groups, and they can create and share Collections to store their data. Collections are the top-level objects in the data organization hierarchy. Within each Collection, there can be many Folders, and the Collection itself is also an access-controlled resource. Typically, Collections are used to group data that share something in common, such as what project the data are used for, or what institution they belong to. Data inside a collection can be organized in a tree-like structure of folders. Folders contain other Folders, Items, or a combination of Folders and Items. Items are the basic units of data in the system, and they may contain from 0 to many Files. Let us advance that we opted for renaming the “Item” entity as “Fileset” when using Girder to implement the repository. In our opinion, the word “fileset” express much better the idea of this entity being “a set of files”.

Files represent raw data objects, and they are stored within an Assetstore. Assetstores are an abstraction representing the location where the raw bytes of Files are actually kept. To use the local server-side storage a folder of the local filesystem of the server must be set as the root of the “assetstore”, but a remote location like a NFS directory can be used also as storage by setting its “mounting point” (a local folder which is indeed a link to the remote location) as the root.

Users can be granted permissions on resources in the system directly and can belong to any number of Groups. One of the main purposes of Groups is to allow role-based access control; resources can grant access to Groups rather than just individual users, such that changing access to sets of resources can be managed simply by changing Group membership. There are four levels of permission a User can hold for a resource. These levels are in a strict hierarchy with a higher permission level including all of the permissions below it. These levels are “no permission” (cannot view, edit, or delete a resource), “READ” (can view and download resources), “WRITE” (includes READ permission, can edit the properties of a resource), and “ADMIN” (also known as own permission, it includes READ and WRITE permission and allows deletion of the resource and control of access to it). Permissions are always additive, that is, given a User with a certain permission on a resource, that permission cannot be taken away from the User by addition of other permissions to the system, but only through removing existing permissions to that User or removing that User from a Group. A site admin always has permission to take any action.

Collections and Folders can have permissions set for both Users and Groups, meaning that a given User or Group can have READ, WRITE, or ADMIN permissions set on either the Collection or the Folder. Moreover,

[54] https://www.ansible.com/

Folders inherit permissions from their parent Folder. Items always inherit their permissions from their parent Folder. Each access-controlled resource (e.g., Folder, Collection) has a list of permissions granted on it, and each item in that list is a mapping of either Users to permission level or Groups to permission level.

Groups can be given any level of access to a resource that an individual User can, and this is managed at the level of the resource in question. For permissions on Groups themselves, Public Groups are viewable (READ permission) to anyone, even anonymous users. Private Groups are not viewable or even listable to any Users except those that are members of the Group, or those that have been invited to the Group. Users can have three levels of roles within the Group. They can be Members, Moderators (also indicates that they are Members), and Administrators (also indicates that they are Members). Moderators can also invite Users to become Members, can accept or reject a request by a User to become a Member, can remove Members or Moderators from the Group, and can edit the Group which includes changing the name and description and changing the Public/Private status of the Group. Administrators of a Group have all of the abilities of Group Moderators and also can delete the Group, promote a Member to Moderator or Administrator, demote an Administrator or Moderator to Member, and remove any Member, Moderator, or Administrator from the Group. The creator of a Group is an Administrator of a group. Any logged in User can create a Group.

The primary method of customizing and extending Girder is via the development of plugins, the process of which is described in the Plugin Development section of this documentation. Plugins can, for example, add new REST routes, modify or remove existing ones, serve up a different web application from the server root, hook into model lifecycle events or specific API calls, override authentication behavior to support new authentication services or protocols, add a new backend storage engine for file storage, or even interact with a completely different DBMS to persist system records – the extent to which plugins are allowed to modify and extend the core system behavior is nearly limitless. Plugins are self-contained in their own directory within the Girder source tree.

Regarding the DICOM plugin, it adds support for previewing DICOM files when viewing an item in Girder. If multiple DICOM files are present in a single item, they are presented as multiple slices. The DICOM image is shown as well as a table of DICOM tags. The window center and width can be changed by the user. Controls allow the user to step through slices, auto-level the window, auto-zoom, or playback the slices at different speeds. This plugin parses the DICOM tags when files are uploaded and stores them in the MongoDB database for quick retrieval. This is mostly used to sort multiple images by series and instance.

5 Y1 prototype implementation

This section details the final set of software tools used to effectively implement the architecture described in section 4, the capabilities of the platform on which the software is running, and a review of the main features offered by the Y1 prototype of the repository.

5.1 Software stack and hardware platform

The core of the architecture is the resource management middleware, whose role is played by Girder, as the correspondence between left and right block diagrams in Figure 5 shows. The core of Girder is a Python application that works as a server, exposing only the API. It relies on a MongoDB database engine on which a Girder-managed database is created for indexing the resources uploaded to the repository. This relation is depicted in Figure 5 by including Girder and MongoDB together in the box of the right block diagram that represents the resource management middleware. Girder stores the resources into an "assetstore" that is mapped to a file system location available for the machine on which the Girder instance is running. That location is the root of a content-addressed folder hierarchy on which the files are stored on chunks of bytes with a fixed size, rather than as the physical files the users upload. In our platform, this location is the mount point of a NFS location corresponding to a region in the CESGA's storage cabin. The connection among the indexing database and the actual file storage location is represented with the double arrows in both block diagrams of Figure 5.

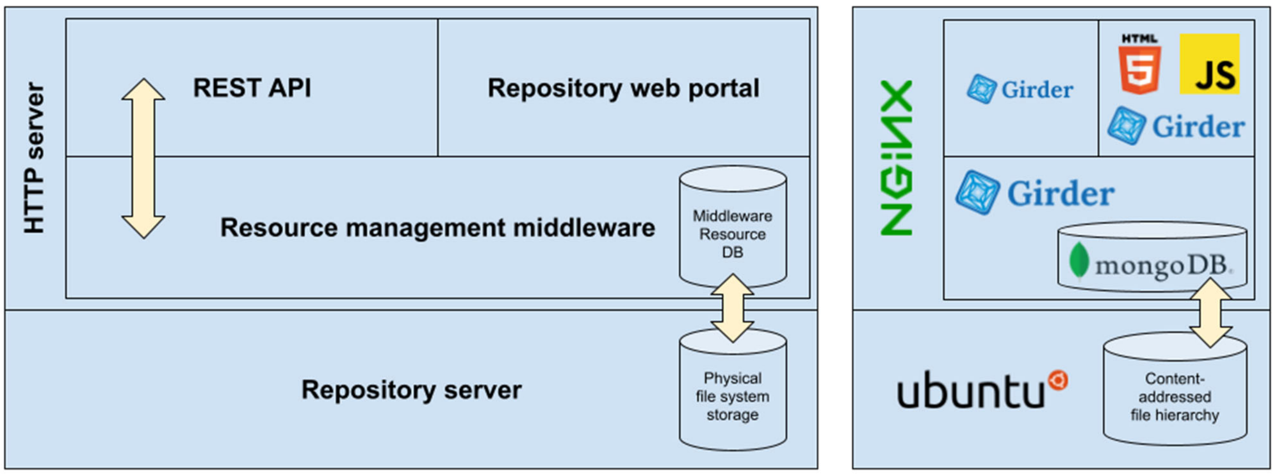


Figure 5: Mapping of the software stack implemented to the Y1 prototype architecture

When the instance of Girder is deployed, it launches an internal HTTP server. This server is in charge of sending the whole HTML5/JavaScript application when a user enters the repository from a web browser. This application is run locally in the user's web browser, using the Girder API to send requests to the Girder internal server listening in our platform. These requests are indeed listened by the Girder internal server and then redirected to the proper component of the Python server-side code. The JavaScript browser-side application makes an extensive use of Backbone.js[55] framework, that works as a local proxy for CRUD operations (create, retrieve, update, delete) with all the Girder model types (Users, Groups, Collections, Folders, Items, Files) maintained in the MongoDB database. This remarkably reduces by design the number of requests among the browser and the server. The DICOM plugin uses Kitware's VTK.js[56] (a library for browser-side scientific visualization) to decode the image data and to render the frames of DICOM files. The web application and the API are depicted in in Figure 5 in the right block diagram on top of the Girder-MongoDB pair, as both rely on the internal server of Girder.

[55] https://backbonejs.org/
[56] https://kitware.github.io/vtk-js/index.html

An instance of NGINX[57] is deployed on the platform to act as a reverse-proxy server. A reverse proxy is a server put in front of another one in order to retrieve resources or attend to requests on behalf of the hidden one. In our case, the reverse-proxy redirects the requests to the aforementioned Girder internal HTTP server. That is the reason because the NGINX block encloses the whole Girder stack (web services -portal, API- and the Girder instance). The insertion of a reverse-proxy improves the security of the whole repository infrastructure by encrypting the connections under the HTTPS protocol (that is the reason of the “https” prefix in the repository URL, https://sinfonia.cesga.es) and dismissing requests from allegedly malicious IP addresses previously included in an access control list.

The server on which this version of the repository is deployed is a virtual machine (VM) created for this purpose in the CESGA’s private cloud infrastructure. The machine has 4 virtual CPUs offloaded onto a node composed of Intel Xeon E5-2650 v3 @ 2.30 GHz 10-core CPUs. It also mounts a virtual hard drive of 20 GB and 16 GB of virtual RAM. The operating system is Ubuntu Linux Server 18.04.6 LTS Bionic Beaver. Instances of Girder, MongoDB and NGINX are installed in the virtual hard drive of the VM, whereas the MongoDB database and the data resources uploaded to the repository are physically stored in a 100 GB partition in the CESGA storage cabin accessible by means of a NFS mountpoint. Daily incremental and weekly full backups are performed for the data stored in that partition, along with manual backups of the virtual hard drive of the VM.

5.2 Fulfillment of regulatory constraints

In section 3.1.3 we identified three main regulation-derived constraints to be fulfilled by the SINFONIA data repository. First, regarding the control and location of the underlying software and hardware infrastructure, both the server on which the virtual machine that supports the repository is hosted, and the storage equipment on which data is saved and backed up are physically installed in CESGA’s premises in the EU (Santiago de Compostela, Spain). This ensures that data collected, generated and/or related to the SINFONIA project will not be processed out of the GDPR territorial scope when interacting through the repository.

Secondly, the implementation of a system of user accounts to avoid undesired and/or unauthorized operations implies the collection of some basic personal information such as the full name of the user, their e-mail address, or their institution. A Privacy Policy informing about the processors of such data, the legal basis for processing them, the duration and location of the storage, the third-party sharing terms and the contact information for any issue related to the policy, is posted in the repository web portal[58]. As data is being stored in CESGA servers, privacy and security is guaranteed by the general CESGA Privacy Policy. Users must accept the SINFONIA Data Repository Privacy Policy when requesting for their account.

With respect to the proper anonymization of the data resources managed through the repository, the Grant Agreement states that it is users’ responsibility rather than of repository administrators to ensure that no personal information is uploaded. Nevertheless, a document with some technical guidelines about de-identification of files and images have been prepared by WP5 to help the consortium to minimize the chance of any personal information leak when sharing data through the repository. These guidelines are posted in the repository web portal in the “SINFONIA Anonymization Guidelines” document[59]. This document introduces the motivation of files anonymization previous to their upload, explains its regulatory basis, and describes a handful of tools for cleaning personal information from DICOM and non-DICOM files along with some quick usage indications for each.

5.3 Customization and extension of original Girder features

The development of the prototype started on top of a plain Girder instance with the DICOM plugin installed, a very basic customization of the layout of pages (welcome messages, SINFONIA logo and name, explicit

[57] https://www.nginx.com/
[58] https://sinfonia.cesga.es/SINFONIA-Users-Privacy-Policy.pdf
[59] https://sinfonia.cesga.es/SINFONIA-Anonymization-Guidelines.pdf

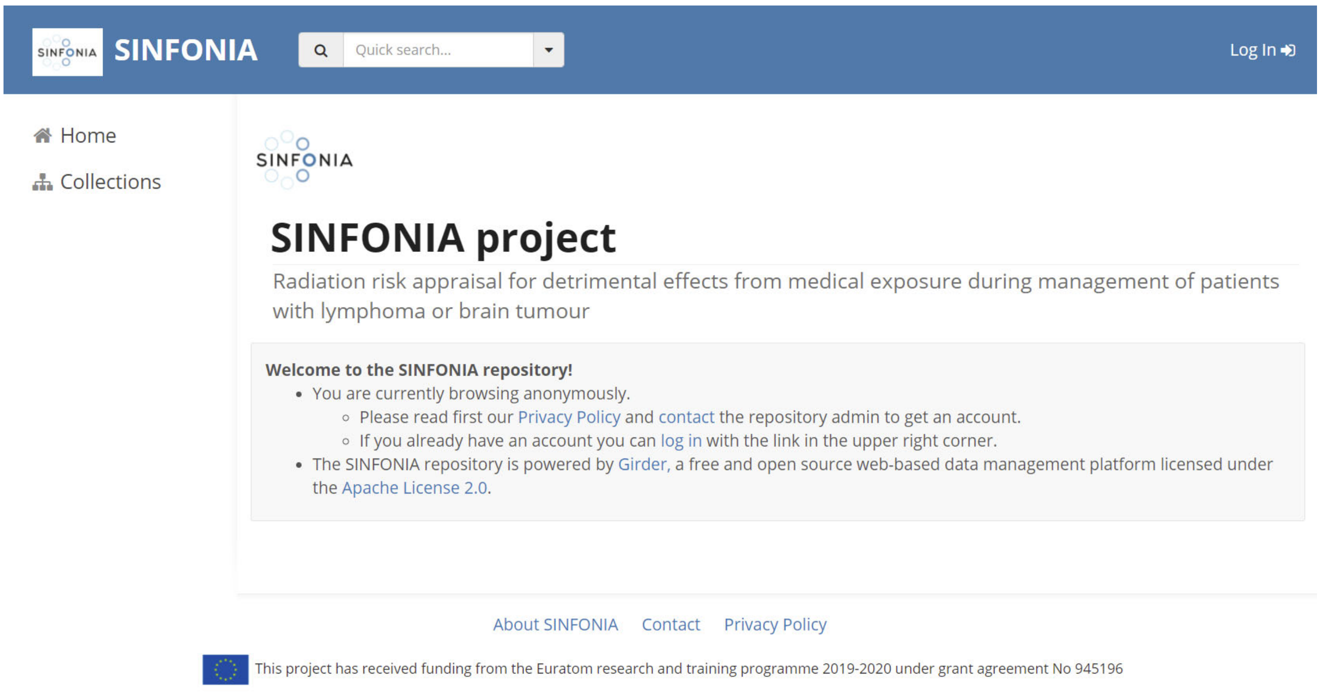
reference to the research program and project grant agreement), and some adjustments on the configuration dashboard to disallow open registration and set the credentials of mail server through which Girder could send password reset messages if requested by users. However, an important part of the features described further in subsection 5.4 relies on several tweaks and improvements we introduced on top of the original Girder implementation:

- **Waiting page during web loading**: Girder web pages are built by a JavaScript web application that is fully downloaded to the browser the first time a visitor reaches the portal. As the loading of that application may be slow sometimes, we have introduced a waiting page to inform the user of this situation, instead of the original blank page that Girder showed by default.
- **Enforcement of closed registration policy**: Despite having set a closed registration policy in the Girder configuration dashboard, a "Register" link was still being shown to anonymous users in some places of the common layout of the pages. We modified the templates that generate the layout to render those links or not depending on the type of registration policy set.
- **Navigation restricted to registered users**: By default, Girder does not limit the general access to the repository to anonymous users, and they are able to inspect resources marked as "Public". In order to avoid the navigation of non-registered users through the repository, we derive them to a warning page if they click on the "Collection" button of the main menu, and we set all the collections as "Private" with access granted to their respective WP groups. Data can be shared among WPs by means of setting explicit permissions on resources' access control lists.
- **Renaming of "Item" entity as "Fileset"**: As described in 4.3.1, the original name that Girder gives to the entities on which one or more physical files are stored is "Item". That name revealed as quite confusing during at a very early stage of the internal development process of the prototype. We decided to rename all the occurrences of that entity in the web portal as "Fileset", which is a denomination much closer to the underlying concept of "set of files".
- **Additional features and information in Fileset pages**: Automatic refresh of total size and last modification date in fileset pages after adding or removing files, deletion of multiple files by means of checkboxes and a confirmation button (replacing the original Girder file-by-file deletion clicking on a "Delete" button in each file entry), and rendering of the creation/modification date of files in their entries.
- **Additional features for the DICOM plugin**: We have extended the behavior of the DICOM plugin with modifications in both its server-side Python code and its client-side JavaScript web application in order to implement these improvements:
    - We have modified the behavior of the JavaScript code of the DICOM web previewer in order to render correctly DICOM files with RGB image data, as the original version was only able to render greyscale images.
    - The original DICOM previewer does not support RT-DOSE, RT-PLAN and RT-STRUCT files, making the page to crash if such files are found in a fileset. We have implemented a workaround to make the previewer able to show the list of DICOM tags of those files without crashing and informing the user that the image data is not available for rendering. We expect to overcome this by including a full web-based research DICOM viewer in the Y2 version of the repository.
    - We have included a dropdown list in the controls of the DICOM previewer. This allows the users to preview a specific instance just by picking it in this new control. The original previewer does not include such a feature, which made users to advance file-by-file manually using the reproduction scroll bar or the Forward button, which is not optimal specially for large filesets.
    - The original DICOM previewer was not being reloaded after adding new files to a DICOM fileset, what led to the issue that those new files cannot be previewed until the next time the web application asked the server to reindex the DICOM data of the modified fileset. We have performed the modifications needed in both the server-side Python code and the JavaScript web application code to trigger that DICOM data reloading.

5.4 Prototype features

In this section the reader will be guided across the main features of the Y1 prototype. Some brief descriptions of those features are given, along with some screenshots of the web portal for illustration purposes.

The starting point is the homepage of the web portal of the prototype, shown in Figure 6. In this page the visitor is welcomed with a message informing them that they are browsing anonymously, so they must either log into the web to continue or read the SINFONIA Data Repository Privacy Policy document and ask the administrators for an account. The general layout of the web portal pages also includes a footer with some useful links ("About SINFONIA" towards the official web of the project[60], "Contact" for sending an email to the repository administrators, and "Privacy Policy" towards the document containing the SINFONIA Data Repository Privacy Policy) and the mandatory reference to the EU research program and the Grant Agreement number of the project.



SINFONIA project

Radiation risk appraisal for detrimental effects from medical exposure during management of patients with lymphoma or brain tumour

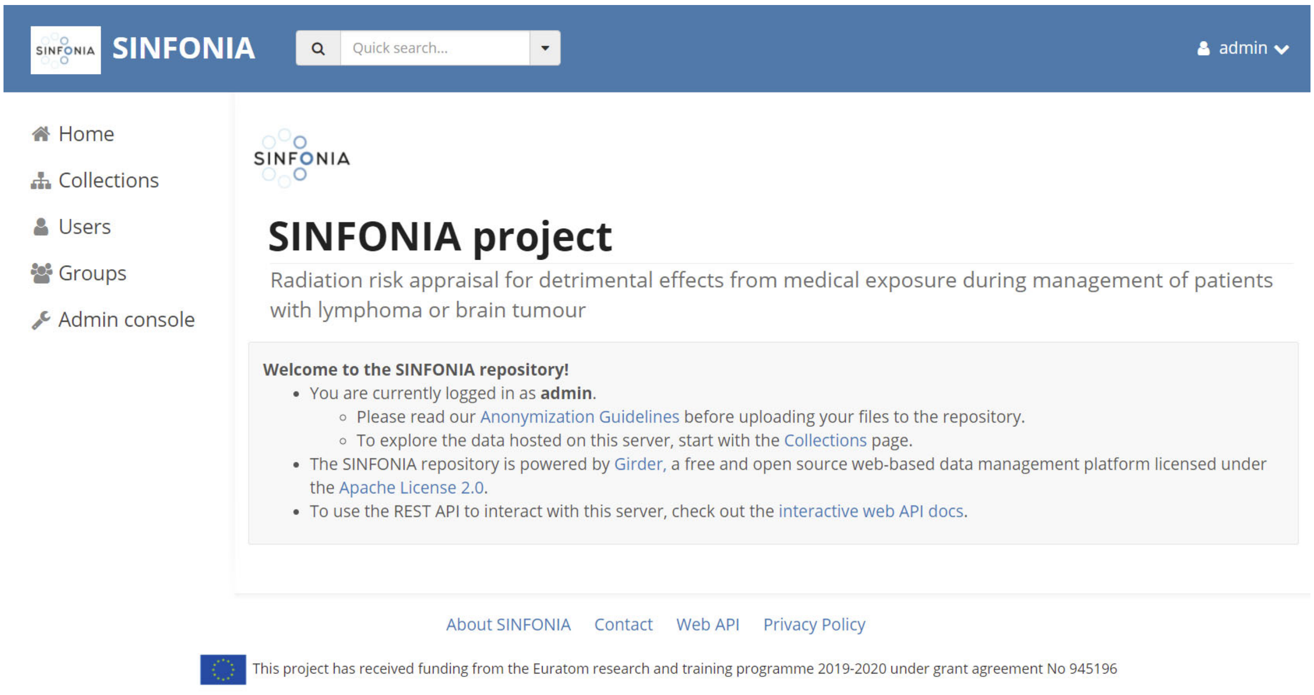
Welcome to the SINFONIA repository!
- You are currently browsing anonymously.
  - Please read first our Privacy Policy and contact the repository admin to get an account.
  - If you already have an account you can log in with the link in the upper right corner.
- The SINFONIA repository is powered by Girder, a free and open source web-based data management platform licensed under the Apache License 2.0.

This project has received funding from the Euratom research and training programme 2019-2020 under grant agreement No 945196

Figure 6: Screenshot of homepage for anonymous users

The "Log In" button in the top right-hand corner allow visitors to login into the repository. If valid credentials are provided, a new landing page (Figure 7) is shown. The landing page informs registered users that they have been successfully logged into the repository and recommends reading the so-called "Anonymization Guidelines". This is a document containing the guidelines mentioned at the end of section 3.1.3 to help users to meet the regulatory constraint about the de-identification of files before uploading them to the repository. The left menu includes new links to sections "Users" and "Groups", which are only shown to registered users. Additionally, the "Admin console" link is only shown for users with administration permissions, which is the case of the "admin" account used to obtain these screenshots.

60 https://sinfonia-appraisal.eu



SINFONIA project

Radiation risk appraisal for detrimental effects from medical exposure during management of patients with lymphoma or brain tumour

Welcome to the SINFONIA repository!

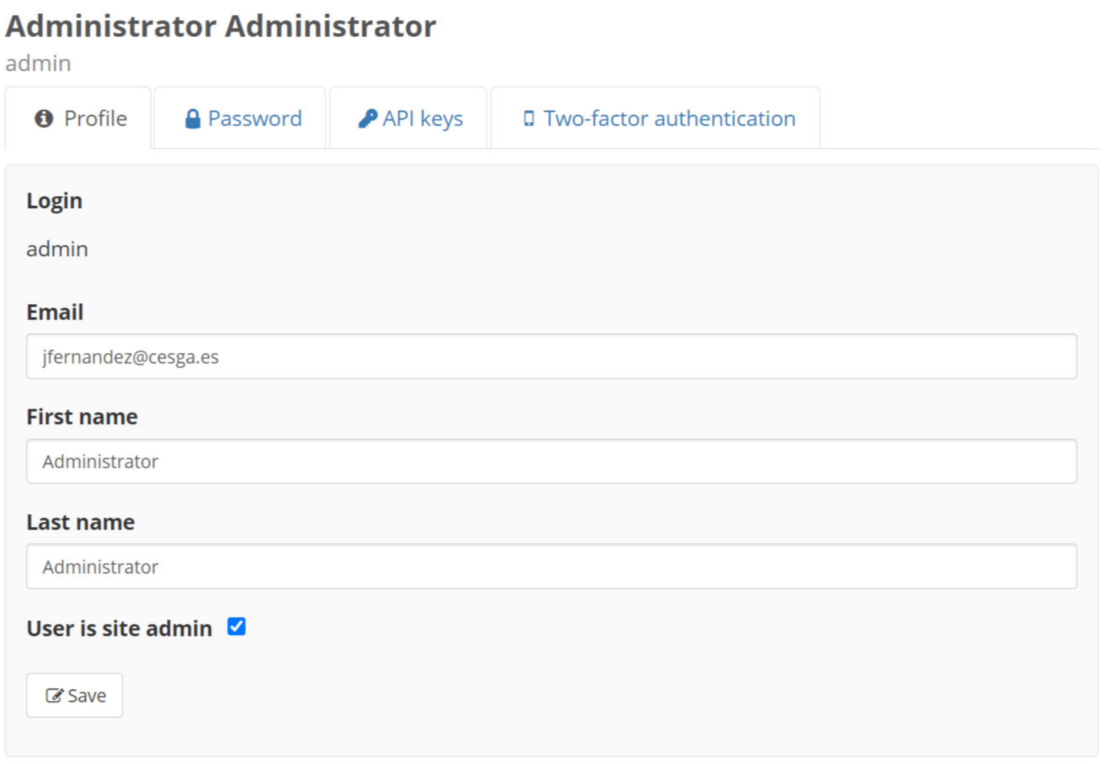
- You are currently logged in as **admin**.
  - Please read our Anonymization Guidelines before uploading your files to the repository.
  - To explore the data hosted on this server, start with the Collections page.
- The SINFONIA repository is powered by Girder, a free and open source web-based data management platform licensed under the Apache License 2.0.
- To use the REST API to interact with this server, check out the interactive web API docs.

About SINFONIA Contact Web API Privacy Policy

This project has received funding from the Euratom research and training programme 2019-2020 under grant agreement No 945196

Figure 7: Screenshot of homepage for authenticated users

Users can see and edit some details of their accounts by clicking on the button with their username in the top right-hand corner and then on “My Account”. In that page (Figure 8) users can edit their profile details, change their password, set keys to grant access through the API on their behalf, and activate the 2-factor authentication by means of mobile apps such as Google Authenticator, Duo Mobile, or FreeOTP.



Administrator Administrator
admin

Login: admin

Email: jfernandez@cesga.es

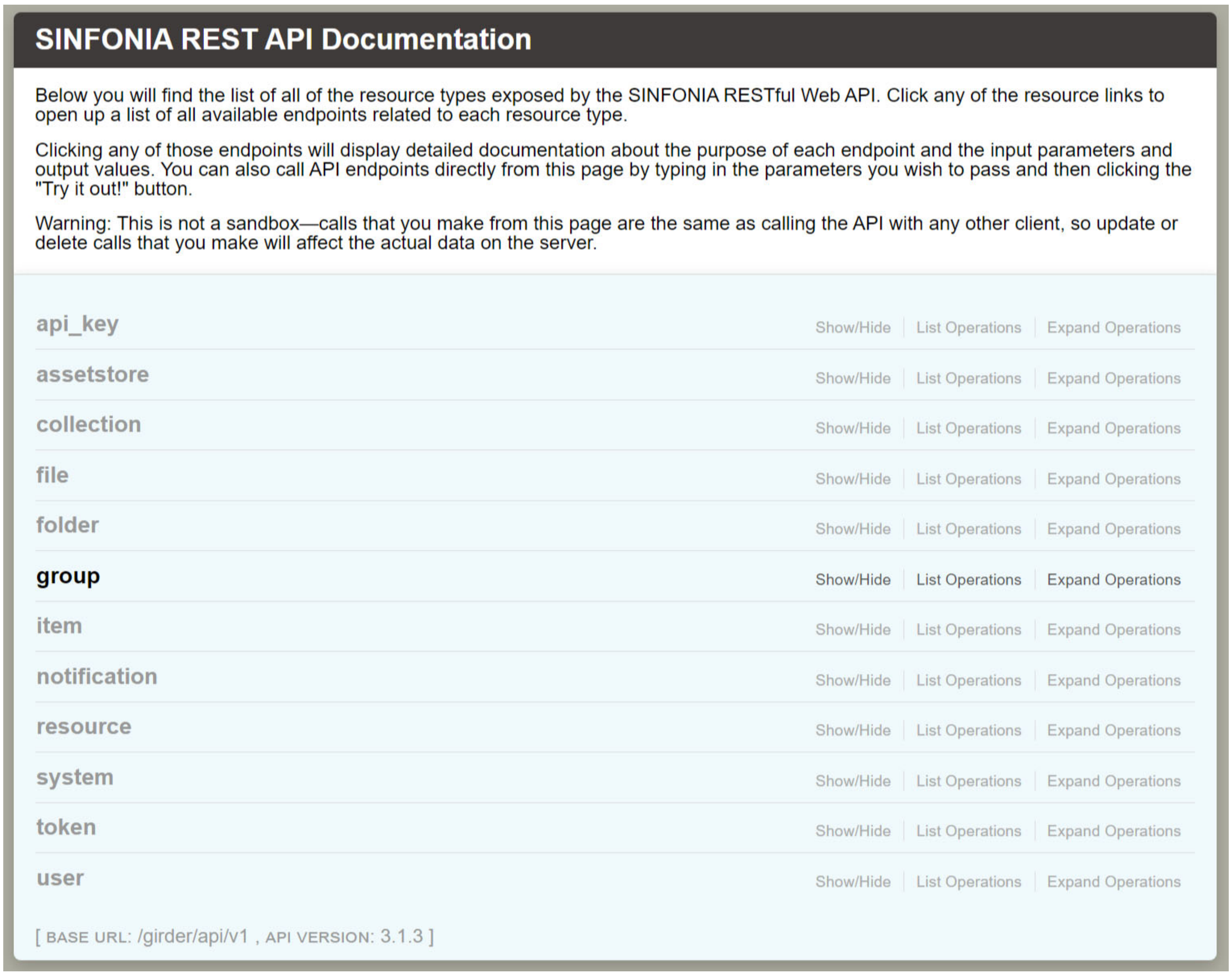
First name: Administrator

Last name: Administrator

User is site admin ✓

Figure 8: Screenshot with detail of "My account" page for "admin" user

Registered users can also see references to the REST API in the last point of the welcome message and in a "Web API" link in the footer. Both links take the user to a page (Figure 9) generated by Girder using Swagger UI[61], which is a tool that builds API documentation automatically from its technical specification. This is indeed the API offered by the underlying Girder instance, but users can leverage it to log into the repository and interact programmatically with it.



SINFONIA REST API Documentation

Below you will find the list of all of the resource types exposed by the SINFONIA RESTful Web API. Click any of the resource links to open up a list of all available endpoints related to each resource type.

Clicking any of those endpoints will display detailed documentation about the purpose of each endpoint and the input parameters and output values. You can also call API endpoints directly from this page by typing in the parameters you wish to pass and then clicking the "Try it out!" button.

Warning: This is not a sandbox—calls that you make from this page are the same as calling the API with any other client, so update or delete calls that you make will affect the actual data on the server.

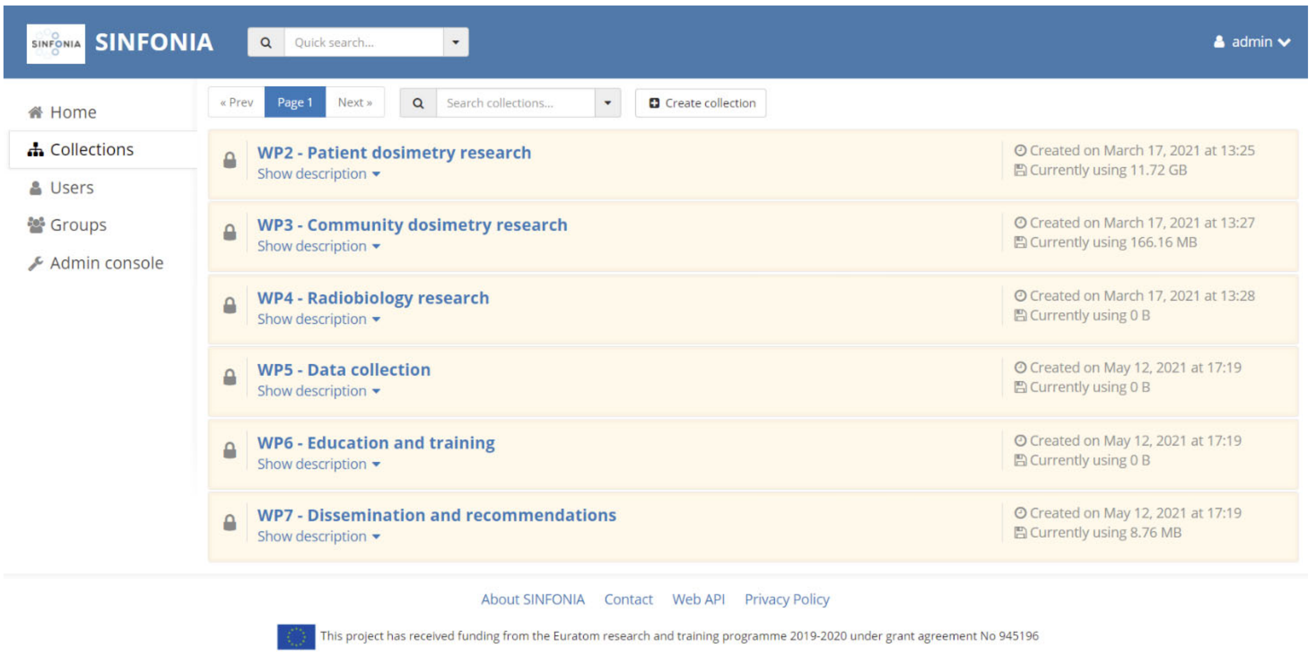
api_key	Show/Hide	List Operations	Expand Operations
assetstore	Show/Hide	List Operations	Expand Operations
collection	Show/Hide	List Operations	Expand Operations
file	Show/Hide	List Operations	Expand Operations
folder	Show/Hide	List Operations	Expand Operations
group	Show/Hide	List Operations	Expand Operations
item	Show/Hide	List Operations	Expand Operations
notification	Show/Hide	List Operations	Expand Operations
resource	Show/Hide	List Operations	Expand Operations
system	Show/Hide	List Operations	Expand Operations
token	Show/Hide	List Operations	Expand Operations
user	Show/Hide	List Operations	Expand Operations

[BASE URL: /girder/api/v1 , API VERSION: 3.1.3]

Figure 9: Screenshot with detail of the REST API Documentation page

The main purpose of this Y1 prototype is to allow SINFONIA researchers to upload both DICOM and non-DICOM datasets in order to keep those files stored in the repository and to share them at their will with other registered individual partners or research groups within the project. Data storage has been organized in separate collections corresponding with each WP of the project, as Figure 10 shows.

[61] https://swagger.io/tools/swagger-ui/


SINFONIA

Home
Collections
Users
Groups
Admin console

« Prev | Page 1 | Next »
Create collection

WP2 - Patient dosimetry research
Show description
Created on March 17, 2021 at 13:25
Currently using 11.72 GB

WP3 - Community dosimetry research
Show description
Created on March 17, 2021 at 13:27
Currently using 166.16 MB

WP4 - Radiobiology research
Show description
Created on March 17, 2021 at 13:28
Currently using 0 B

WP5 - Data collection
Show description
Created on May 12, 2021 at 17:19
Currently using 0 B

WP6 - Education and training
Show description
Created on May 12, 2021 at 17:19
Currently using 0 B

WP7 - Dissemination and recommendations
Show description
Created on May 12, 2021 at 17:19
Currently using 8.76 MB

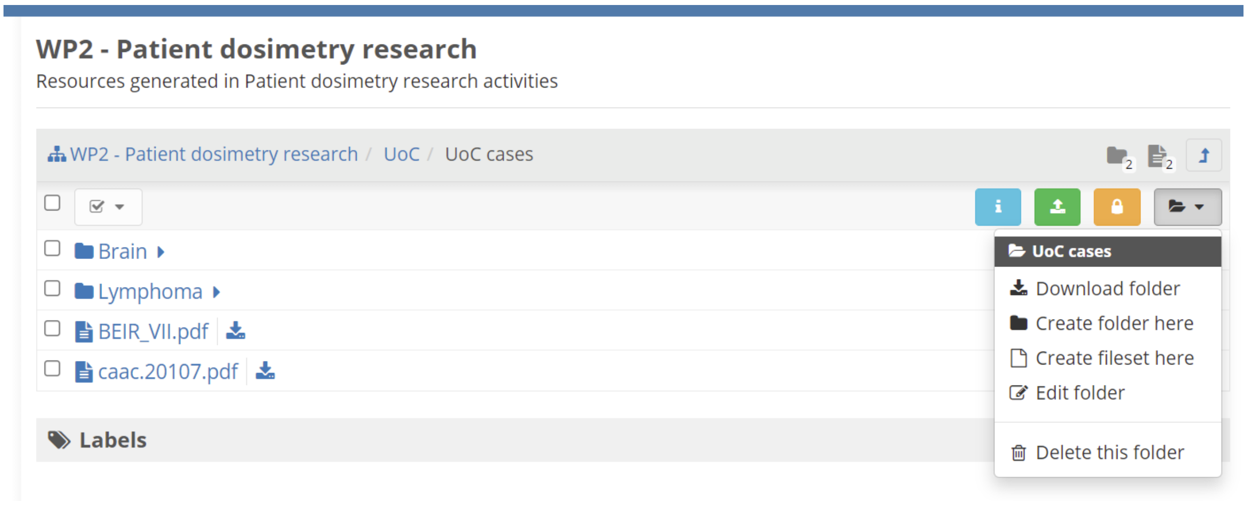
About SINFONIA | Contact | Web API | Privacy Policy

This project has received funding from the Euratom research and training programme 2019-2020 under grant agreement No 945196

Figure 10: Screenshot of Collections page for authenticated users

Inside the collection of a WP, users can create their own custom folder hierarchies and use those folders to save files. For instance, in Figure 11 we can see an example on which UoC opted for creating within the WP2 collection ("WP2 - Patient dosimetry research") their own folder ("UoC") and then creating a subfolder for clinical cases ("UoC cases"). The screenshots also show how they separated the cases in "Brain" and "Lymphoma" folders.

Figure 11 and Figure 12 shows the different operations that can be performed within collections or folders. The menu on the right of the page shown in Figure 11 includes options for downloading the current folder, creating a new subfolder or a new fileset, and editing their properties (name, description). The blue button opens a dialog with basic information of the folder or collection. Files can be uploaded into new filesets created either by means of the aforementioned option or by clicking on the green button near the menu. In the latter case, each file will be uploaded in its own separate fileset. The checkboxes and the menu on the left of the page shown in Figure 12 are used to select and manage the resources (either filesets and other folders) stored in a given collection or folder. Namely, the selection can be downloaded, picked to be copied into or moved to another collection or folder, or deleted from its current location. The same menu on the left will appear enabled in the desired destination location to move or copy there the resources previously selected.


WP2 - Patient dosimetry research

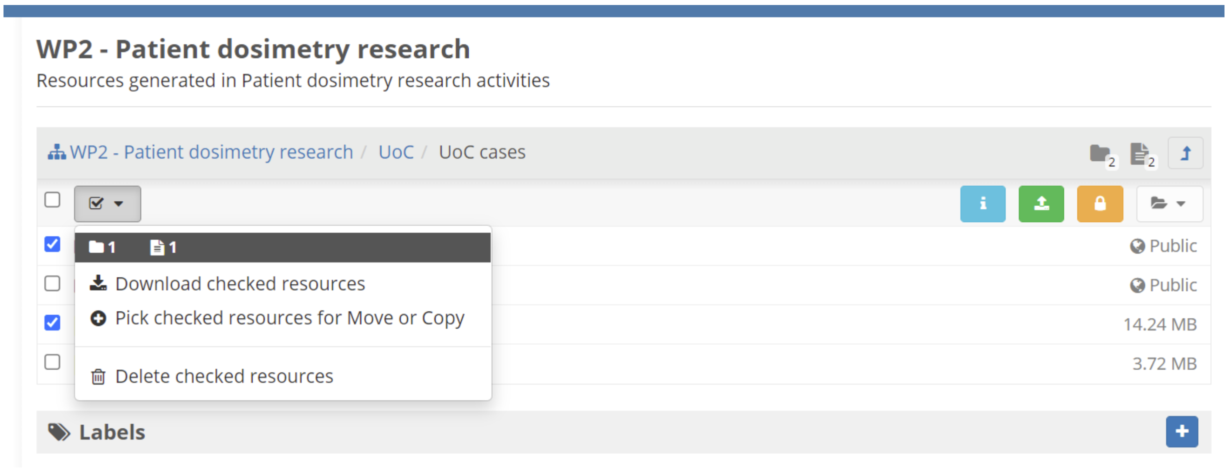
Resources generated in Patient dosimetry research activities

WP2 - Patient dosimetry research / UoC / UoC cases

Brain
Lymphoma
BEIR_VII.pdf
caac.20107.pdf

Labels

Figure 11: Screenshot of Folder page with Folder Operations menu displayed on the right



WP2 - Patient dosimetry research

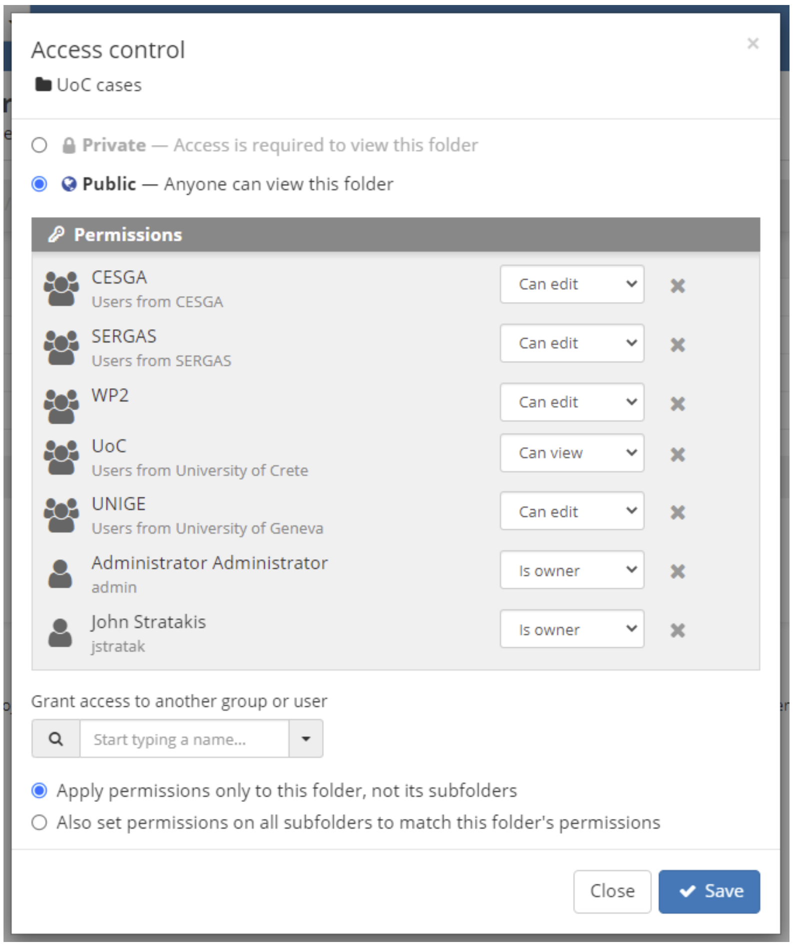
Resources generated in Patient dosimetry research activities

WP2 - Patient dosimetry research / UoC / UoC cases

1 1
Download checked resources
Pick checked resources for Move or Copy
Delete checked resources
Public
Public
14.24 MB
3.72 MB
Labels

Figure 12: Screenshot of Folder page with Selection Options on the left

Users will be able to perform the operations described until now just if they have been previously granted with the proper permissions. The permissions for each collection and folder of the repository are set in an access control list like the one shown in Figure 13. Owners of a given collection or folder can set it as “Public” (anyone with an account can view the folder) or “Private” (explicit permissions must be set). Those explicit permissions can be set for both individual users and groups, which can be allowed to read (“Can view”), write (“Can edit”), or manage (“Is owner”) that given resource. Finally, any modification regarding the access control can be applied just for the current resource or also recursively to all the subfolders within it. Let us also add that, as the reader may have already supposed, only users with “Is owner” permissions for a given resource can invoke this feature.



Access control

UoC cases

[ ] Private — Access is required to view this folder

[✓] Public — Anyone can view this folder

Permissions

CESGA Users from CESGA	Can edit
SERGAS Users from SERGAS	Can edit
WP2	Can edit
UoC Users from University of Crete	Can view
UNIGE Users from University of Geneva	Can edit
Administrator Administrator admin	Is owner
John Stratakis jstratak	Is owner

Grant access to another group or user

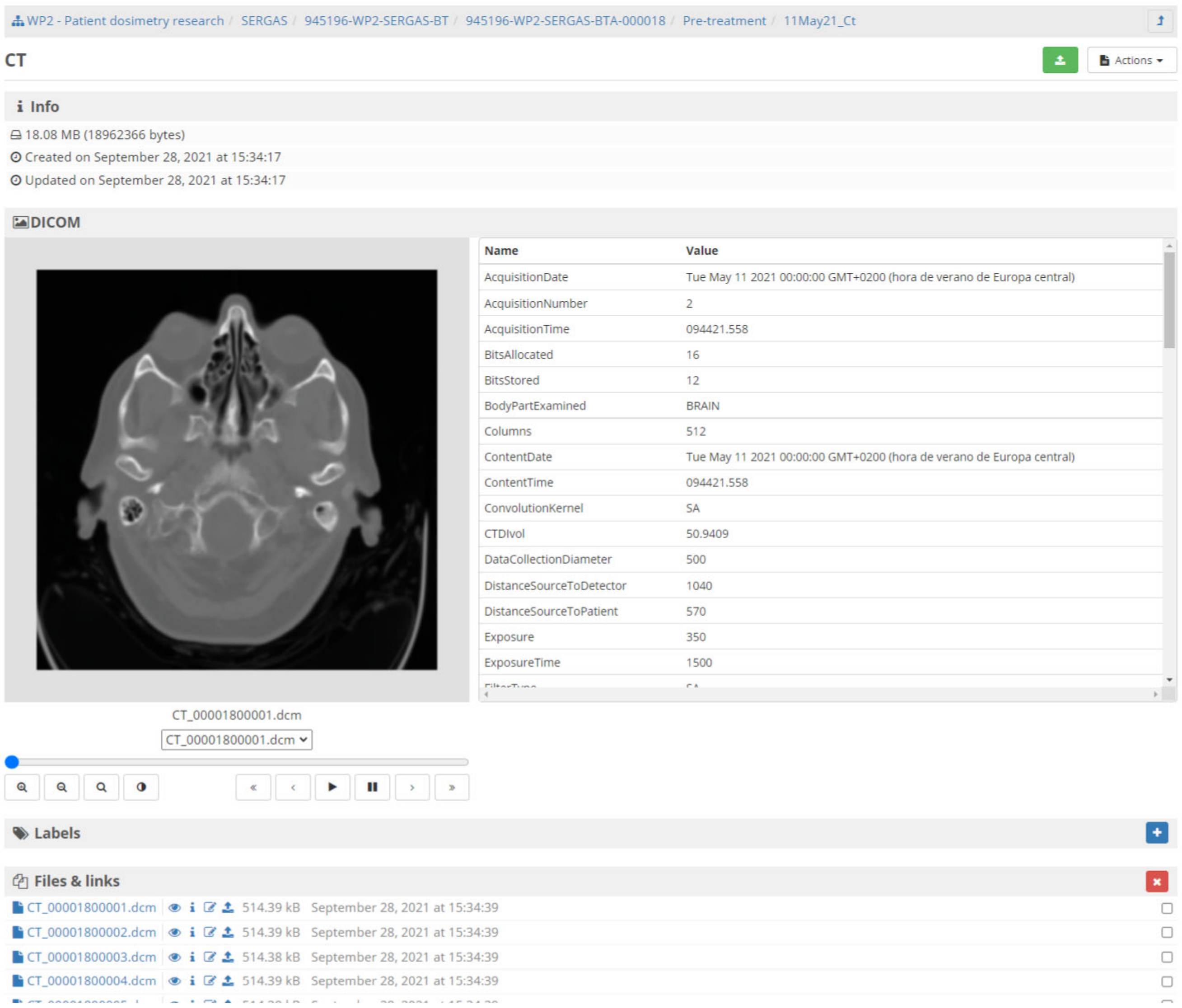
Start typing a name...

[✓] Apply permissions only to this folder, not its subfolders

[ ] Also set permissions on all subfolders to match this folder's permissions

Figure 13: Screenshot of Access Control dialog

By means of the DICOM plugin of Girder, the repository detects at upload time if a fileset is indeed a series of DICOM files. If so, it inspects the files looking for DICOM tags, which are parsed and stored in the internal database to make them available for quick retrieval and indexing purposes. This plugin also provides the web portal with a DICOM previewer. Thus, if a fileset contains several DICOM files, the user can navigate through the images and read the DICOM tags of each file. An example of how this previewer is integrated in the platform is shown in the screenshot from Figure 14.



WP2 - Patient dosimetry research / SERGAS / 945196-WP2-SERGAS-BT / 945196-WP2-SERGAS-BTA-000018 / Pre-treatment / 11May21_Ct

CT

Actions

Info

18.08 MB (18962366 bytes)
Created on September 28, 2021 at 15:34:17
Updated on September 28, 2021 at 15:34:17

DICOM

Name	Value
AcquisitionDate	Tue May 11 2021 00:00:00 GMT+0200 (hora de verano de Europa central)
AcquisitionNumber	2
AcquisitionTime	094421.558
BitsAllocated	16
BitsStored	12
BodyPartExamined	BRAIN
Columns	512
ContentDate	Tue May 11 2021 00:00:00 GMT+0200 (hora de verano de Europa central)
ContentTime	094421.558
ConvolutionKernel	SA
CTDIvol	50.9409
DataCollectionDiameter	500
DistanceSourceToDetector	1040
DistanceSourceToPatient	570
Exposure	350
ExposureTime	1500

CT_00001800001.dcm

CT_00001800001.dcm

Labels

Files & links

CT_00001800001.dcm	514.39 kB	September 28, 2021 at 15:34:39	☐
CT_00001800002.dcm	514.39 kB	September 28, 2021 at 15:34:39	☐
CT_00001800003.dcm	514.38 kB	September 28, 2021 at 15:34:39	☐
CT_00001800004.dcm	514.39 kB	September 28, 2021 at 15:34:39	☐

Figure 14: Screenshot of DICOM fileset page showing the image data and tags previewer

Files of the fileset are shown in the list at the bottom of the page. Each file has an entry with its name (which is a link for downloading the file), some buttons for additional actions (open on a new browser tab, see basic information, download, replace) and a checkbox at the rightmost end of the entry. These checkboxes were added to support the deletion of several files at a time, as stated in subsection 5.3.

The resources stored in the repository can be inspected by navigating through the collections, folders and filesets, or through the "Quick search" box available in the top left-hand corner of the any page, next to the project logo and name. As Figure 15 shows, three different search methods are supported: full text, that looks for exact occurrences of the string in resource names and descriptions; prefix search, that looks for resources names and descriptions that start with the given prefix; and DICOM metadata search, an extension added by the DICOM plugin that is able to search the given text in the values of the tags of the DICOM files stored in the repository.

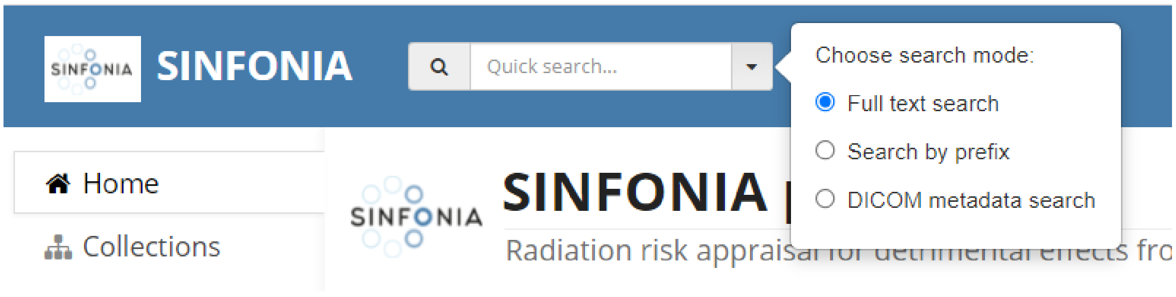


Figure 15: Screenshot with detail of "Quick search" box

Let us come back to the rest of the options shown in the leftmost menu of the portal. The “Users” button of that menu takes the user to a paginated list with the user accounts active in the portal is shown (Figure 16). Each username is indeed a link to the “personal collection” of the logged-in user. On their personal collection, like the one shown in

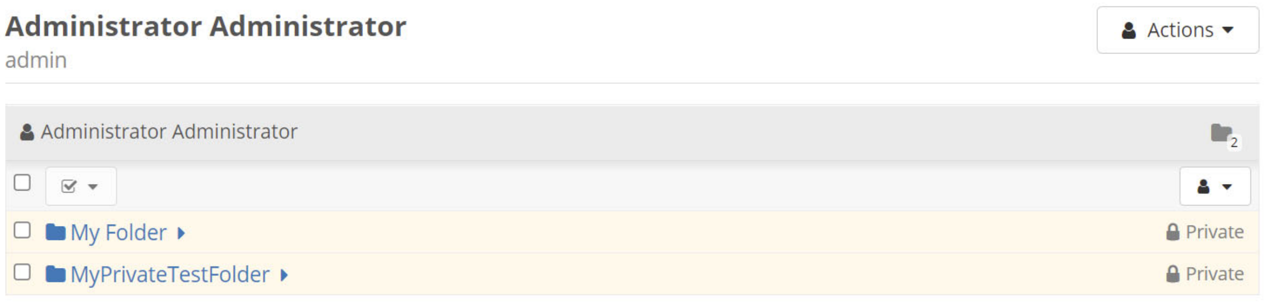
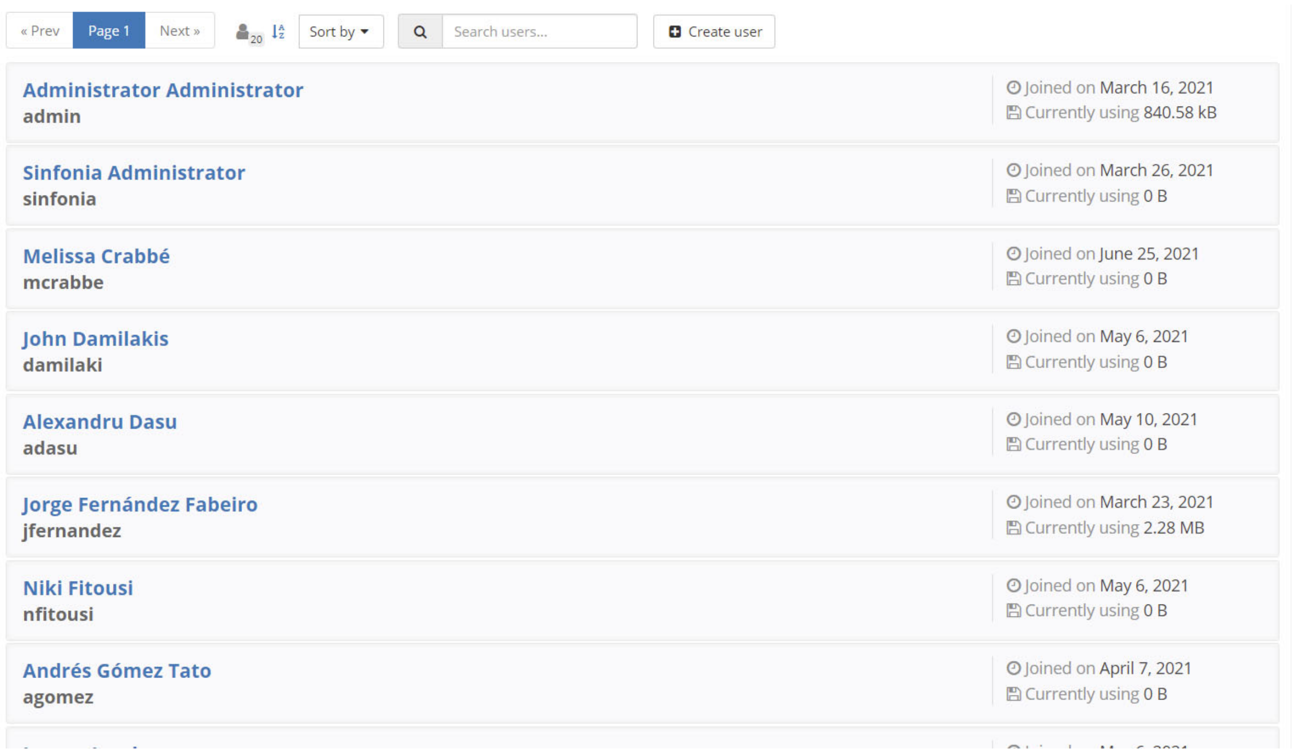


Figure 17 for “admin” user, the owner can create folders, set access permissions for them, and upload resources into filesets, as in the regular ones. As the folders stored in personal collections are part of the repository too, data on them can be eventually moved to a WP collection or folder if desired. This feature is useful for sharing datasets privately with other users and/or groups in situations like when attaching data to an email is not possible (e.g.: size of dataset exceeds the storage limit of the sender and/or recipient mailboxes), when it is not clear on which regular collection/folder the dataset must be stored, or how the files must be distributed in folders and filesets, or when a second opinion from a project partner is desired before sharing the dataset.



Administrator Administrator
admin
Joined on March 16, 2021
Currently using 840.58 kB

Sinfonia Administrator
sinfonia
Joined on March 26, 2021
Currently using 0 B

Melissa Crabbé
mcrabbe
Joined on June 25, 2021
Currently using 0 B

John Damilakis
damilaki
Joined on May 6, 2021
Currently using 0 B

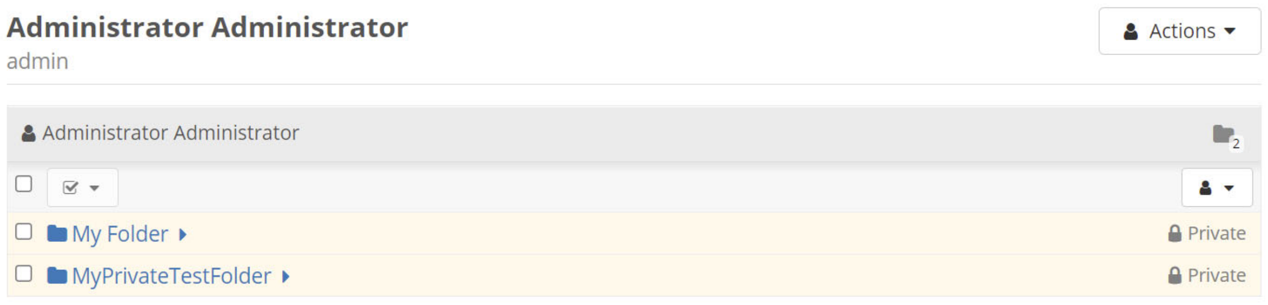
Alexandru Dasu
adasu
Joined on May 10, 2021
Currently using 0 B

Jorge Fernández Fabeiro
jfernandez
Joined on March 23, 2021
Currently using 2.28 MB

Niki Fitousi
nfitousi
Joined on May 6, 2021
Currently using 0 B

Andrés Gómez Tato
agomez
Joined on April 7, 2021
Currently using 0 B

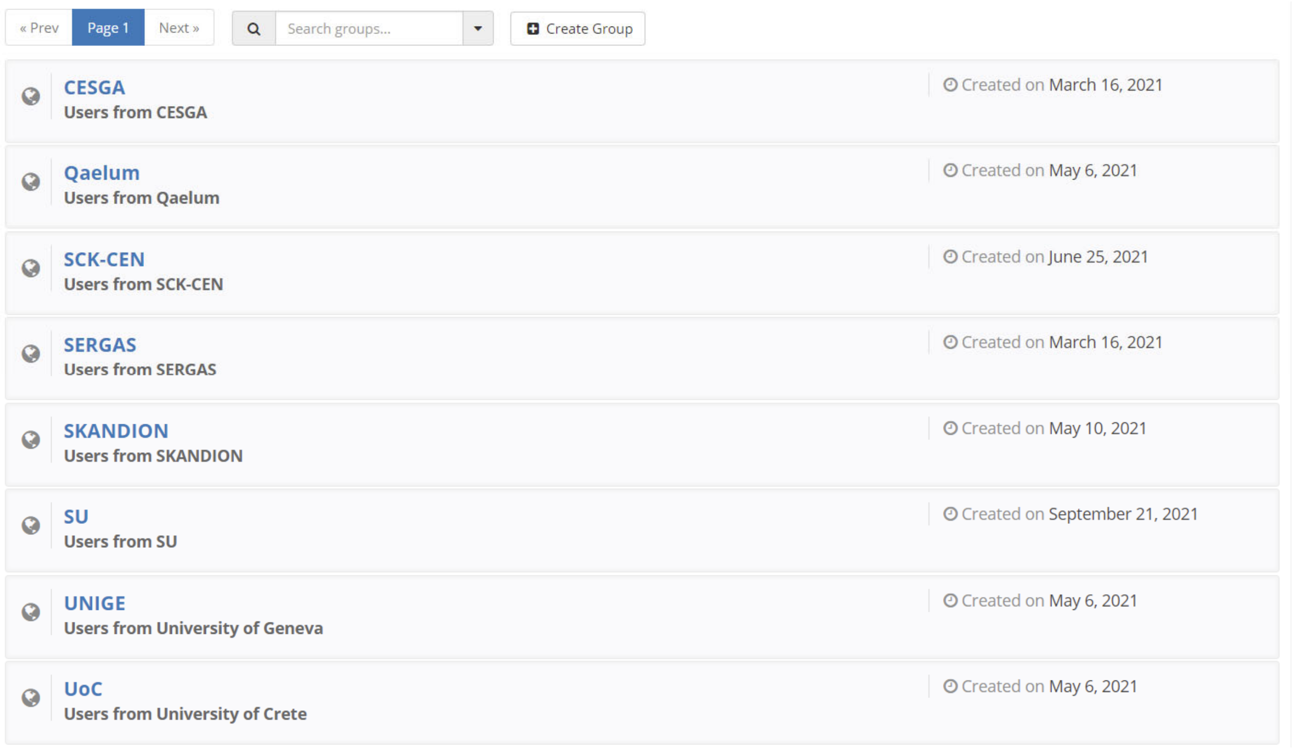
Figure 16: Screenshot of the list of active users of the repository


Administrator Administrator
admin

My Folder — Private
MyPrivateTestFolder — Private

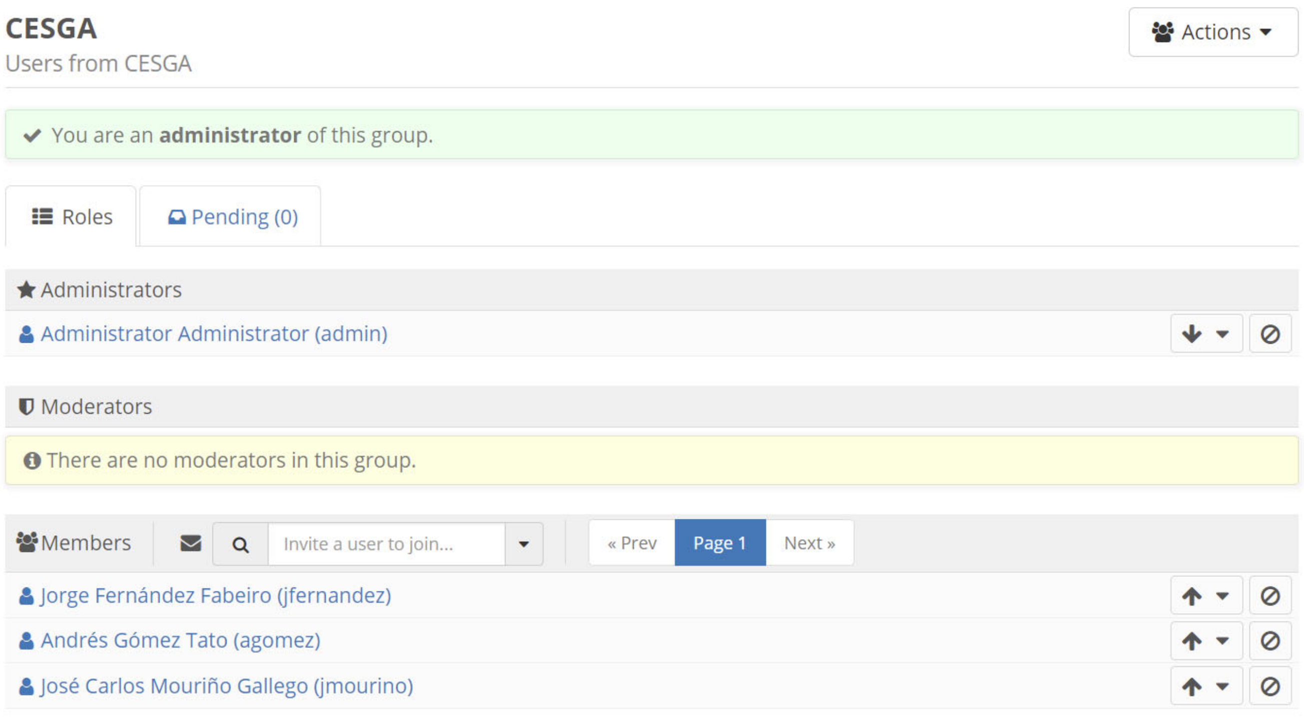
Figure 17: Screenshot of the personal folder of a user

In a similar vein, the “Groups” button takes the user to a paginated list with the groups defined in the portal (Figure 18). Each group in the list is indeed a link to a details page like the one for CESGA group shown in Figure 19. Group details pages include a list of group members classified as plain members, moderators (they can invite and approve members), and administrators (they can edit group properties and can add, invite, and remove members, moderators, and administrators). The operations mentioned for moderators and administrators can be performed from these group detail pages. Finally, the “Admin console” button, which is shown only to the portal admins, gives access to the control panel of the underlying Girder instance and the plugins installed.


« Prev | Page 1 | Next »
Search groups...
Create Group

Group	Created
CESGA Users from CESGA	Created on March 16, 2021
Qaelum Users from Qaelum	Created on May 6, 2021
SCK-CEN Users from SCK-CEN	Created on June 25, 2021
SERGAS Users from SERGAS	Created on March 16, 2021
SKANDION Users from SKANDION	Created on May 10, 2021
SU Users from SU	Created on September 21, 2021
UNIGE Users from University of Geneva	Created on May 6, 2021
UoC Users from University of Crete	Created on May 6, 2021

Figure 18: Screenshot of the Groups list of the repository


CESGA
Users from CESGA
Actions

You are an **administrator** of this group.

Roles | Pending (0)

Administrators
Administrator Administrator (admin)

Moderators
There are no moderators in this group.

Members
Invite a user to join...
« Prev | Page 1 | Next »
Jorge Fernández Fabeiro (jfernandez)
Andrés Gómez Tato (agomez)
José Carlos Mouriño Gallego (jmourino)

Figure 19: Screenshot with details and users list for CESGA group

6 Y1 prototype validation

6.1 Internal validation

The version of the prototype described in section 5 is composed of a plain Girder instance with the DICOM plugin installed, a basic customization of the layout of pages (welcome messages, SINFONIA logo and name, explicit reference to the research program and project grant agreement), and all the extensions implemented on top of the original Girder source, as detailed in subsections 5.3 and 5.4. It was demonstrated to the WP5 members in M9 (May 2021). The WP5 members were asked to make their experiments with this very first version of the prototype, in order both to get familiar with the Girder interface and to think about existing feature to adapt, additional features to include, or eventual issues to solve. Namely, SERGAS, UoC and QAELUM participated actively in these tasks.

SERGAS invested an important amount of time on preparing the data of three patients with the RT relevant information (CBCT, CT, Dose distribution, RT plan, RT structures, PET/CT). The data was then uploaded as a way to get familiar with the Y1 prototype interface. UoC team have uploaded a sample test set of DICOM and non-DICOM data. DICOM data comprised of an anonymized CT scan from a lymphoma patient and non-DICOM included calculated 3D dose distributions for various CT kVp (kilovoltage peak) potentials. Non-DICOM data also contained ROIs for thorax organs in an ImageJ[62] ZIP package and overlayed GIF format. All the file formats required by UoC (DCM, ZIP, PDF, ROI, GIF) were uploaded successfully and handled correctly from the repository interface. QAELUM focused specially on experiments about users and permissions management, data upload and access (by means of virtual patients' data) and friendliness of the web interface.

As a result of their experiments, some bugs and confusing features were also addressed after being detected by SERGAS, UoC, and QAELUM. An abnormal generation of the links to the personal collections of users invited to join groups, shown in the "Pending Invitations" section of group details page, was fixed. The "Eye" buttons in entries of the resource list of Folder pages have been removed, as it behaved like the "Download" one except for the very specific case of trying to view the content of a fileset with a single file inside. Content of individual files in a Fileset can be viewed always by entering first in the Fileset page. The initial resource detail dialogs included a Girder-generated "Unique ID". This item of information is not showed anymore, as it was being confused with an eventual project-related identifier (which is not implemented, as identification scheme is still on discussion). A more expressive warning message was set before confirming the deletion of a user account. The session timeout had to be expressed in full days and it was set for the Girder default of 180 days. Both the JavaScript web application and the server-side Python code of Girder were modified to accept floating values and then setting the timeout as 0.125 days (3 hours). The Search Results page just showed the resource name for each occurrence found. This was confusing as there was no way to distinguish two resources with the same name uploaded to different places. The template that generates this page was edited to show the full path to each resource found.

WP5 partners also proposed several improvements and possible additional features that will be considered when developing the next versions of the repository. UoC and SERGAS collaborated in the definition of a resource identification scheme by circulating a proposal made of the concatenation of several items: the SINFONIA project ID, an internal ID somehow connected to WP tasks and subtasks, the partner ID, an order number of the patient, and additional partner-level notation for particular characteristics of the data uploaded (pediatric patients, specific anatomical regions, etc.). This proposal sets the foundations for implementing an identification mechanism for data uploads in further versions. Other are related to a better support of massive data uploads and downloads, such as the activation of an FTP service, or to reinforcing the security of such processes by limiting access to specific known IP addresses from the SINFONIA partners. There were also proposals for improving the design and the features of the web interface, as the extensive usage of tooltips for guiding users through the application, the inclusion of a contact form for reporting issues or making suggestions instead of a plain link to an e-mail address, or enrich the DICOM viewer with proper auto-levels for any kind of image or a measurement function.

[62] https://imagej.nih.gov/ij/

A full research DICOM viewer is expected to be included in the Y2 version. Some ideas about the integration of IA algorithms have arisen too, like providing a jupyterHub as a service for running scripts remotely on uploaded data. Such features are to be studied and evaluated for implementation gradually from Y3 version and ahead.

6.2 Consortium validation

The version resulted from the internal WP5 validation process was introduced to the consortium in a live demonstration performed in a webinar held at the end of M10 (June 2021). The consortium was also informed that the prototype was online[63], encouraging all the project partners to make their experiments in order get familiar with the web portal, to learn more about the features provided and to think about eventual suggestions or bug reports. CESGA as leader of the development of the repository remained attentive to any suggestion or bug report that the consortium could send to sinfonia@cesga.es, a dedicated mailbox open for these questions. The workflow for any eventual proposal is to assess it first in order to distinguish minor patches whose application on the fly to this Y1 prototype would be manageable from major changes that would fit better as new features in further versions of the repository. As this document is being written, some proposals have been received, such as the inclusion of an embedded online editor for text documents, spreadsheets, and presentations.

6.3 Review of the delivered prototype

Let us finish the validation section with a self-assessment about how, and up to which grade of fulfilment, the requirements collected from the users and the regulatory constraints have been effectively addressed in the Y1 prototype delivered in M11. A chart summarizing this review is shown in Figure 20, with each requirement being marked in green, yellow, or not marked depending on being totally, partially, or not fulfilled, respectively.

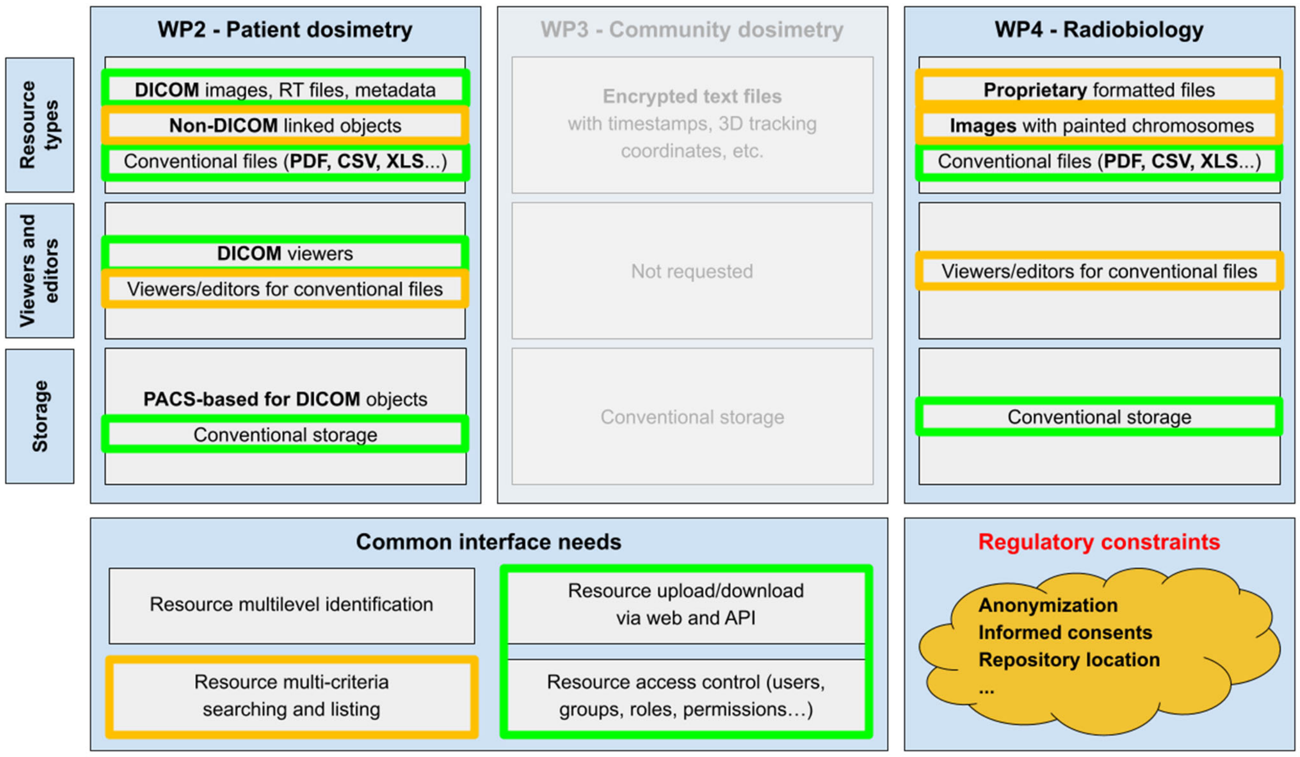

WP2 - Patient dosimetry
WP3 - Community dosimetry
WP4 - Radiobiology
Resource types
Viewers and editors
Storage
DICOM images, RT files, metadata
Non-DICOM linked objects
Conventional files (PDF, CSV, XLS...)
Encrypted text files with timestamps, 3D tracking coordinates, etc.
Proprietary formatted files
Images with painted chromosomes
Conventional files (PDF, CSV, XLS...)
DICOM viewers
Viewers/editors for conventional files
Not requested
Viewers/editors for conventional files
PACS-based for DICOM objects
Conventional storage
Conventional storage
Conventional storage
Common interface needs
Resource multilevel identification
Resource upload/download via web and API
Resource multi-criteria searching and listing
Resource access control (users, groups, roles, permissions…)
Regulatory constraints
Anonymization
Informed consents
Repository location
...

Figure 20: Summary chart of needs and constrains fulfilment

Regarding the needs stated by WP2 users, both DICOM and conventional files can be managed through the prototype, as the underlying customized Girder instance is able to store any kind of files. Non-DICOM objects

[63] https://sinfonia.cesga.es

linked to DICOM ones can be uploaded too, but the way to effectively represent that link is not implemented as the resource multi-level identification system has not been defined yet. The DICOM previewer allows users to inspect the DICOM tags of any DICOM object, and both greyscale and RGB pixel data of images coming from several DICOM modalities. WP3 partners declined to use the repository for their daily research work, as they have very specific requirements including real-time upload of massive amounts of tracking data. Supporting those needs would lead to an oversized system in terms of network bandwidth and physical storage. Nevertheless, a separate collection and several user accounts were created for WP3 researchers just in case they want to share conventional files like data spreadsheets, text reports or processed images. With relation to the fulfillment of specific needs of WP4 researchers, there are no special features for manipulating proprietary-formatted files or images with painted chromosomes, but they may be uploaded to and shared through the prototype. In general, any registered user is able to upload conventional files like text reports or data spreadsheets, but for the moment visualization and edition capabilities are limited to the support of the local browser for those formats.

Most of the common interface features are provided through the customized Girder web application: upload and download features are offered through the web portal and the API, access control lists can be set for Girder collections and folders, and a limited search engine able to find resources by names, descriptions, and the content of DICOM tags, is available. The resource multi-level identification was not addressed since the debates about the most proper way to define an identification scheme valid for all the research activities in the project are still open.

With relation to the regulatory constraints, a first approach to the anonymization guidelines mentioned in the project proposal has been composed. Privacy and security of research data is enforced by restricting both the registration to SINFONIA members and the navigation through the repository to logged-in users. In order of their account to be created, project members must explicitly accept first the SINFONIA Data Repository Users Privacy Policy. Finally, the physical infrastructure on which the repository server is deployed and the data is stored and backed up is hosted on CESGA’s premises in the EU (Santiago de Compostela, Spain)

7 Conclusion

In this deliverable, the software development method of the SINFONIA data repository has been presented, focusing on its application during Y1 in order to generate the first version of the repository released in M11 (July 2021).

A brief insight of the purpose of the SINFONIA project has been provided as an introduction for the reader, along with the key objectives that the implementation of a data repository intends to fulfill and correlating them with the relevant work packages of the project and the tasks within those work packages. A software engineering process for driving the development of the data repository and consistent with the general SINFONIA methodology has been defined. The steps of that process and the relationships among them has also been described.

The analysis step of the development process was composed of two parallel tasks: the capture of the users' needs and requirements and an analysis of those needs and the underlying regulatory constraints; and a survey about existing technologies able to provide features similar to those expected for the SINFONIA data repository, either as finished software products or as development frameworks to compose it.

The design of the prototype architecture was started with a review of the needs, requirements, and constraints to compose a model of how the information generated within the SINFONIA project could be organized, and to define a basic architecture for the data repository. Then, the main features and characteristics of the different software solutions reviewed were mapped to all the needs, requirements and constraints identified, and to the design foundations outlined. As a result, Girder, one of the file storage servers studied, was the software solution selected to implement the core of the repository. An extensive description of Girder's main characteristics has been provided too for completion's sake.

The prototype implementation has been described thoroughly. That description included an enumeration of the software tools deployed and some details about the underlying hardware platform, linking all these components to their counterparts from the basic architecture. Some details about how the regulatory constraints identified during the analysis step are enforced in the implemented repository have been provided too. The look-and-feel customizations and features extensions applied on the Girder instance deployed have been also introduced, along with an extended description enriched with screenshots of the main features that Y1 prototype offers.

Finally, a two-phase repository validation process has been narrated. In the first phase WP5 partners made their own experiments to get familiar with the prototype and to think about new features, improvements, or eventual issues to solve. Some bugs and confusing features identified by the partners, as well as improvements and additional features proposed for further iterations, have been enumerated. A similar validation process launched in the second phase for the whole consortium has been described too. The validation process was concluded with a self-assessment about up to which extent the prototype implemented addresses the users' needs and requirements, and the regulatory constraints identified.

References

[1] J. Crinnion, Evolutionary Systems Development: A Practical Guide to the Use of Prototyping Within a Structured Systems Methodology, Pitman, 1991.

[2] R. S. Pressman, Software Engineering: A Practitioner’s Approach, 7 ed., New York: McGraw-Hill, 2010.

[3] R. S. Pressman, "Companion slides for Software Engineering: A Practitioner’s Approach, Seventh Edition," McGraw-Hill, 2009.

[4] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons, "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific Data,* vol. 3, March 2016.

[5] DICOM RT Working Group, *Leaflet with DICOM RT information for AAPM meeting,* 1998.

[6] L. Wills, “A Very Brief Introduction to the GDPR Recitals,” American Bar Association, 01 05 2019. [Online]. Available: https://www.americanbar.org/groups/litigation/committees/minority-trial-lawyer/practice/2019/a-very-brief-introduction-to-the-gdpr-recitals/. [Accessed 27 10 2021].

[7] S. Schnyder, “Revised to Match the EU General Data Protection Regulation — or Almost,” Sidley Austin LLP, [Online]. Available: https://www.sidley.com/en/insights/publications/2021/02/part-2-revised-to-match-the-eu-general-data-protection-regulation-or-almost. [Accessed 27 10 2021].

[8] A. Job, “Swiss nDPA - a clumsy copy of EU requirements?,” KPMG Switzerland, 15 07 2021. [Online]. Available: https://home.kpmg/ch/en/blogs/home/posts/2021/07/swiss-ndpa-gdpr.html. [Accessed 27 10 2021].

[9] NRG @ Washington University School of Medicine, "XNAT Documentation: Case Studies," [Online]. Available: https://wiki.xnat.org/documentation/case-studies/. [Accessed 05 11 2021].

[10] A. Scott, W. Courtney, D. Wood, R. de la Garza, S. Lane, M. King, R. Wang, J. Roberts, J. A. Turner and V. D. Calhoun, "COINS: An Innovative Informatics and Neuroimaging Tool Suite Built for Large Heterogeneous Datasets," *Frontiers in Neuroinformatics,* vol. 5, pp. 33.1-33.15, 2011.

[11] D. Wood, M. King, D. Landis, W. Courtney, R. Wang, R. Kelly, J. A. Turner and V. D. Calhoun, "Harnessing modern web application technology to create intuitive and efficient data visualization and sharing tools," *Frontiers in Neuroinformatics,* vol. 8, pp. 71.1-71.7, August 2014.

[12] S. Das, A. P. Zijdenbos, J. Harlap, D. Vins and A. C. Evans, "LORIS: a web-based data management system for multi-center studies," *Frontiers in Neuroinformatics,* vol. 5, p. 37.1–37.11, 2012.

[13] T. Skripcak, U. Just, M. Simon, D. Buttner, A. Luhr, M. Baumann and M. Krause, "Toward Distributed Conduction of Large-Scale Studies in Radiation Therapy and Oncology: Open-Source System Integration Approach," *IEEE Journal of Biomedical and Health Informatics,* vol. 20, no. 5, p. 1397–1403, September 2016.

[14] J. Poortvliet, “Digital Imaging for Medicine in Nextcloud,” Nextcloud, 19 02 2018. [Online]. Available: https://nextcloud.com/blog/digital-imaging-for-medicine-in-nextcloud/. [Accessed 27 10 2021].

[15] R. H. Choplin, J. M. Boehme and C. D. Maynard, "Picture archiving and communication systems: an overview.," *RadioGraphics,* vol. 12, no. 1, p. 127–129, January 1992.

[16] S. Jodogne, "The Orthanc Ecosystem for Medical Imaging," *Journal of Digital Imaging,* vol. 31, p. 341–352, May 2018.

[17] S. Jodogne, “References. Who uses Orthanc. What has been written about Orthanc.,” Osimis, 2021. [Online]. Available: https://www.orthanc-server.com/static.php?page=references. [Accessed 28 10 2021].

[18] H. J. Hazarika, A. Handique and S. R. S. Ravikumar, "DICOM-based medical image repository using DSpace," *Collection and Curation,* vol. 39, no. 4, p. 105–115, February 2020.

[19] DICOM Standards Committee, “DICOM PS3.4 2021d - Service Class Specifications: C6.1.1 Patient Root Query/Retrieve Information Model,” NEMA, 2021. [Online]. Available: http://dicom.nema.org/medical/dicom/current/output/html/part04.html#figure_C.6-1. [Accessed 28 10 2021].

[20] DICOM Standards Committee, “DICOM PS3.3 2021d - Information Object Definitions: 7. DICOM Model of the Real World,” NEMA, 2021. [Online]. Available: http://dicom.nema.org/medical/dicom/current/output/html/part03.html#chapter_7. [Accessed 28 10 2021].

Annex A Questionnaire for partners

Task 5.1 – Questionnaire for partners: Definition of platform architecture

In order to design an initial architecture for the computing platform that will host SINFONIA's repository (that will be created and improved during the project), we need your feedback concerning data from your institution (to be generated and uploaded) and possible algorithms (to be developed). Please, help WP5 Task 5.1 by filling this questionnaire with the maximum detail you can and sending it back by February 5th, 2021, EOB at the latest. Please note that subsequent information may be needed following your feedback, to specifically tailor your repository requirements.

1. Data Summary. Provide a summary of the data (DICOM and non-DICOM) addressing the following issues:
1.1 Specify the file types and data formats you will upload to the repository. Please describe what kind of information you will need to store and access on the platform. (For example, if you want to store radiotherapy treatments, let us know if you want to use DICOM-RT along with DICOM CT/MRI, or any other file format such as FASTA, etc.)
1.2 Do you need to store structured files such as DICOM objects? Do you want the repository to automatically extract and store the metadata associated with those files? (for example, from DICOM-RT do you want to extract info such as the planned doses?)
1.3 Do you need to store additional files? Are these files independent from and/or related to the structured files referred in 1.2? Do these additional files need to be stored in a structured format (e.g., because they contain format-specified metadata) or only as plain files (free text, references to external files, etc)? How these external data should be connected with the corresponding DICOM files? (for example: if you want to store DICOM datasets, but also want to store files like a clinical report in PDF or additional patient-/treatment-related information as a structure -e.g., via HL7 standards-)
1.4 State the expected size of the data to be uploaded. (How large is a typical size of your information per examination and/or per patient?) You can use this calculator to make a first guess)
1.5 How many (patient) examinations do you plan to upload to the platform? (this should be directly connected to information provided in each partner's Data Management Plan)
1.6 Do you need to keep persistent and unique identifiers (such as Digital Object Identifiers) for your data objects?
1.7 Please propose or outline any naming conventions that will be used

1.8 Indicate approximately, in project month, when do you plan to start uploading your information. Also provide an estimation of how long you expect the uploading process to last.

2. Accessing Data Upload, download, searching, and visualization needs.
2.1 **Uploading information to the repository** How do you expect to upload the data to the repository? (Web, API, Software, Other) Please specify if you need multiple uploading methods depending on the data object type (i.e., you may want to upload DICOM files directly from an imaging equipment or PACS and to upload PDF clinical reports through the repository web)
2.2 **Downloading information from the repository** How do you plan to retrieve your data from the repository? (Web, API, Software, Other) Please specify if you need multiple downloading methods depending on the data object type (i.e., you may want to download DICOM-RT files directly to a treatment equipment and retrieve a PDF clinical report from the repository web)
2.3 Specify what methods or software tools will be needed to handle the data. Will you provide documentation about this software or the software itself? (e.g., open-source software)?
2.4 **Searching for data objects** Indicate all the criteria you expect to search for your uploaded data in the repository after uploading (name, DOI, keywords, etc.)
2.5 **Data remote preview before/after upload and/or download** Do you need to preview your data? What kind of files (DICOM imaging files, reports, patient information, etc.) do you need to preview? Please specify how and when you expect to preview those files (for example, if you upload a DICOM-CT imaging file, you may want to preview it remotely after upload using your browser without downloading it to your PC, or you may want to preview a full dataset before upload and just pick the files you are really interested on being stored in the repository)
2.6 Please indicate or suggest any way that your data should be listed or organized in the repository web portal (paginated lists, organized in folders per contributor/patient/treatment, etc.)

2.7 Please indicate how you would like to have your search results displayed in the repository web portal (per contributor, per patient, per treatment, etc.)

3. Data Security Data anonymization and access rights.
3.1 **Anonymization** EU and national privacy regulations may oblige partners to have any privacy-sensitive information purged (anonymized) before uploading to the repository. Do you need any kind of assessment about the anonymization performed on your data before uploading?
3.2 **Pseudonymization** Do you need to keep a track of the anonymized information after uploading? If so, please specify how you expect to identify it.
3.3 **Management of access rights** Who will be authorized to access your data? (for example, only you, your group/hospital, project partners, general public, etc.)
3.4 What are the expected lifetimes of those access rights? (specific stages of the project, the whole project, indefinitely, patient treatment duration, etc.)
3.5 Up to what extent do you need to adjust the aforementioned authorization rights? (for example, you may want to access indefinitely to the whole dataset of a clinical case you manage, but allow to access to general public during a period of time)

4. AI Algorithms First survey about your needs regarding the integration of your eventual AI algorithms in the repository. You will receive a technical questionnaire if interested at a further stage of the project
4.1 Do you plan to develop AI algorithms? Do you expect to integrate your algorithms in the platform to be developed?
4.2 If you plan to develop AI algorithms, will you need any kind of remote processing (training and/or inference) within the platform? What kind of images will you be dealing with?
4.3 If you plan to develop AI algorithms, will your algorithms (code, etc.) be freely accessible to everyone or only to authorized users? (as in question 3.3)

5. Other Please provide any other initial requirements for the repository shared infrastructure and add a level to the included ones
Below you will find some typical functional and non-functional requirements for research repositories. Your suggestion will aid WP5 group and repository developers to identify your data storage needs and subsequently aid the developing process. Please provide information about which of the proposed requirements must/should/may be included in the platform and add additional ones (if not yet covered by previous questions). The level terminology is included at the end of this document.

Functional Requirements

Functional requirements are product features or functions that the developers must implement to enable users to accomplish their tasks.

Table A.1: Functional Requirements

No.	**Description**	**Comments/ Examples**	**Level** (see terminology below)
1.	Identifiers	e.g., support of PID	
2.	Authorization and authentication	Single sign-in or multiple authentication method	
3.	Metadata	Support different metadata schemes, extended metadata, labelling by data owner, metrics for metadata quality assessment, support XML storing, search engine	

4.	Data access	Support data download, download API, manual and automatic search, multiple data versions	
5.	Data organization	Logical naming	
6.	Integration / Interoperation	Support for different files and OS, support for programming language (Python, R), support for analysis and visualization tools, email support for submission and verification, Interoperation among individual data	
7.	User interface	Provide a seamless and consistent interface. Allow file uploading for each metadata field, when needed, allow search by community, organization, allow enabling/disabling all internal and external data services;	
8.	User interface	Provide a sandbox where users can test their data before submission, allow specific users (project administrator, etc.) download specific data	
9.	Preview		
10.	Integration of the different tools developed by other partners		
11.	Availability of fully annotated data	Where and when the data were collected	
12.	Ranking based on metadata	Connection of data with type of patients / institution / …	
13.	Cross-referencing of data	No duplication of data	
14.	Sharing data in a collaborative environment	Easy access to other partners’ data (anonymized)	

Non-Functional Requirements

Include requirements about storage needs, security, data access control or functionalities, auditing, usability, robustness (such as Mean Time Between Failures), performance, availability, ease of maintenance, interfaces, supported hardware, software interfaces, user interfaces, legal and regulatory issues, exploitation and licenses, applicable standards, etc.

Table A.2: Non-Functional Requirements

No.	Description	Comments/ Examples	Level (see terminology below)
1.	Security	React to unauthorised, accidental, or unintended usage and provide access only to legitimate users	
2.	Capacity / scalability	Meet the agreed capacity requirements and designed to accommodate increased data traffic, storage volumes, workloads, and users	
3.	Volumetric / performance	Meet the agreed capacity requirements	
4.	Location	HPC system	
5.	Availability/failover	Uptime 24/7/365, minimizing downtime	
6.	Recoverability	Regular backups/snapshots	
7.	Compliance with privacy and protection standards	EU GDPR, other legal requirements in users' countries Partner uploading data is responsible for the anonymization process	
8.	Deployment	Release a new minor or patch version, replacing the previous minor or patch version	
9.	Serviceability	Be designed so that technical support personnel are able to monitor and manage it in operation	
10.	Test environment	An alternative (staging) testing platform may be designed for testing and assurance before main platform is launched	
11.	Number of simultaneous accesses greater than number of partners		

12.	Uploading time lesser than 2 minutes/500MB		
14.	Human-friendly interfaces		
15.	Storage needs	Sufficient storage must be foreseen (avoid slow use of the data repository due to 'overloaded' or inefficient data storage)	
16.	Data security	Protection from malware attacks	
17.	Data preservation	History of data versions	

Terminology

Keywords to indicate requirement levels

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 to indicate requirement levels.

MUST

This word, or the terms "REQUIRED" (abbreviation "REQUIRE") or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT

This phrase, or the phrase "SHALL NOT" (abbreviation "SH-NOT"), means that the definition is an absolute prohibition of the specification.

SHOULD

This word, or the adjective "RECOMMENDED" (abbreviation "RECOM"), means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT

This phrase, or the phrase "NOT RECOMMENDED" (abbreviation "NOT-REC") mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood, and the case carefully weighed before implementing any behaviour described with this label.

MAY

This word, or the adjective "OPTIONAL" (abbreviation "OPTION"), means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

Annex B Partners' answers to the requirements questionnaire

Task 5.1 – Questionnaire for partners: **Definition of platform architecture**

In order to design an initial architecture for the computing platform that will host SINFONIA's repository (that will be created and improved during the project), we need your feedback concerning data from your institution (to be generated and uploaded) and possible algorithms (to be developed). Please, help WP5 Task 5.1 by filling this questionnaire with the maximum detail you can and sending it back by February 5th, 2021, EOB at the latest. Please note that subsequent information may be needed following your feedback, to specifically tailor your repository requirements.

1. Data Summary.
Provide a summary of the data (DICOM and non-DICOM) addressing the following issues:

1.1 Specify the file types and data formats you will upload to the repository. Please describe what kind of information you will need to store and access on the platform. (For example, if you want to store radiotherapy treatments, let us know if you want to use DICOM-RT along with DICOM CT/MRI, or any other file format such as FASTA, etc.)

UoC: Our uploaded files will include DICOM images, DICOM-RT structures (for contouring) and potentially DICOM RT Plans. We will also upload files that contain volumetric dose data (dose images) from Monte Carlo simulations. These files are saved in DICOM format (Part 10 of DICOM standard) but do not strictly adhere to DICOM standardized application profiles (Part 11 of DICOM standard) as they are not generated by a specific clinical procedure. We suggest storing them separately as non-DICOM files (although they will be saved in DICOM format and will be associated with CT DICOM images through common UID).

SERGAS: We will include DICOM image objects and DICOM-RT objects and modules: RT Image (CT, CBCT, PET/CT, DR), RT Structure Set (contours and volumes), RT Plan (treatment beams, prescription, patient setup...), RT Dose (dose distributions).

SCO: Data consisting of pre-treatment diagnostic PET and CT, treatment planning CT, treatment plans, pre-treatment CBCT, IGRT kV-planar DR and follow-up CT and PET scans will be collected to be used as input for the development of a system for personalised dosimetry to account for in-field and out-of-field organ exposure from scatter dose during photon-based therapy for lymphoma and brain tumour patients. We will include DICOM image objects and DICOM-RT objects and modules: RT Image (CT, CBCT, PET/CT, DR), RT Structure Set (contours and volumes), RT Plan (treatment beams, prescription, patient setup...), RT Dose (dose distributions).

SKANDION: We will include DICOM objects (CT scans, CBCT scans and 2D images), as well as DICOM-RT objects (structure sets, treatment plans, 3D dose matrices.

UNIGE: The dataset contains three parts. The 1st part is chest CT images in DICOM standard (.dcm) format for 20 paediatric and 50 adult patients. The second part is Abdomen/Pelvic CT images for 20 paediatric and 50 adult patients. In addition, the 3 D dose distribution generated by Monte-Carlo simulations is provided for these two parts. The organ-level dose, 3D dose distribution and organ label map are available. The third part consists of whole-body PET/CT images at various time points, i.e., dynamic scans, utilized for evaluation of patient-specific absorbed dose according to the real pharmacokinetic data. Besides the derived input functions are available in excel comma separated values (.csv) format for 12 adult patients.

UGENT-WP2: Upload in repository ONLY if approved by hospital/ethical committee. If yes, DICOM formats will be used for nuclear medicine / radiological / radiotherapy data.

SCK-CEN: WP3 could make use of the repository within task 3.1 and task 3.2. In both cases, the type of data uploaded to the repo will consist of encrypted texts including timestamps, 3D coordinates from object and person tracking, and calculated organ doses. Whether this will be done by directly transferring files (e.g., In JSON format) or via messages such as through TCP/IP, it is still to be established.

SU: We will upload METASYSTEMS files with images of cells with painted chromosomes.

UJK: We will upload pdf files with images of FACS gates and numerical tables.

UGENT-WP4: For micronuclei no images will be uploaded.

QAELUM: In WP 5 Qaelum will help define the platform architecture that will be used to collect data and help integrate the AI algorithms. Qaelum will validate the integrated AI algorithms in a commercial radiation dose management solution as a proof-of-concept. An analysis will be made on the possibilities of collecting data automatically and how results can be presented. No data will be uploaded to the data repository. Further, in WP 6 an interactive multidisciplinary open online course on clinical dosimetry and radiation risk analysis will be organised, followed by the development of an AI algorithm that selects the relevant teaching material for the healthcare professional considering daily-life and patient case – specific conditions. Data generated from this algorithm will encourage the healthcare professional to effectively open/read the relevant teaching material facilitating a lifelong learning climate. The course and the AI algorithm are not intended to be uploaded to the data repository that is the context of this questionnaire.

1.2 Do you need to store structured files such as DICOM objects? Do you want the repository to automatically extract and store the metadata associated with those files? (for example, from DICOM-RT do you want to extract info such as the planned doses?)

UoC: Yes, DICOM files will be stored, and metadata should be accessible to enable search of relevant data in DICOM and non-DICOM files.

SERGAS: Yes, we do need to store DICOM objects and metadata

SCO: Yes, DICOM files will be stored, and metadata will be accessible to enable search of relevant data in DICOM and non-DICOM files.

SKANDION: Yes, we need to store DICOM data as well as metadata included in DICOM header format and keywords concerning radiation type, irradiation/treatment site etc.

UNIGE: Our data are in DICOM format and all necessary DICOM metadata are included

UGENT-WP2: Yes. With respect to metadata depends how data of retrospective study will be used. If in-house (UGent) calculations are performed no further features are needed; if retrospective UGent data will be handled by other groups in WP2, then yes.

SCK-CEN: The repository will not need to extract any metadata from the uploaded data.

SU / UJK / UGENT-WP4: No

QAELUM: No

1.3 Do you need to store additional files? Are these files independent from and/or related to the structured files referred in 1.2? Do these additional files need to be stored in a structured format (e.g., because they contain format-specified metadata) or only as plain files (free text, references to external files, etc)? How these external data should be connected with the

corresponding DICOM files? (for example: if you want to store DICOM datasets, but also want to store files like a clinical report in PDF or additional patient-/treatment-related information as a structure -e.g., via HL7 standards-)

UoC: Non-DICOM data for our studies will be related to specific DICOM image set (patient exams) and will include:

- Specific organ absorbed radiation dose
- CTDIvol - Dose Length Product or other absolute or relative norm method
- Patient characteristics such as age, gender, weight, BMI, WED (water equivalent diameter) etc…
- Dosimetric data may include:
    - Clinical Indication
    - Procedure type
    - Acquisition parameters
    - Method used to generate dosimetric data (e.g., Monte Carlo simulation)
    - Details such as fixed current value versus tube current modulation, X-ray spectrum
    - Simulated equipment model

SERGAS: There will not be necessary to store additional files to ones stated in 1.2.

SCO: Non-DICOM data will be stored according to WP4 requirements

SKANDION: At present there is not necessary to store additional files to ones stated in 1.2.

UNIGE: An additional patient-/treatment-related information in excel comma separated values (.csv) format for dynamic examinations are available named by "UNIGE_Dynamic_info"

UGENT-WP2: Probably yes, but not decided yet.

SCK-CEN: N/A

SU / UJK / UGENT-WP4: Excel file(s) with scoring results.

QAELUM: No

1.4 State the expected size of the data to be uploaded. (How large is a typical size of your information per examination and/or per patient?) You can use this calculator to make a first guess)

UoC: CT DICOM images will normally accommodate approx. 150MB per patient. However, simulations will increase non-DICOM patient data by approx. a factor of 10 (different scanners, kV, mA modulation, etc.). Consequently, average data size per patient would be approx. 1200-1500 MB.

SERGAS: Larger than 500 MB (4 CT + 4 PET-CT + 30 CBCT + dose + contours + plan)/patient

SCO: The same approx. size as UoC: CT DICOM images will normally accommodate approx. 150MB per patient. However, simulations will increase non-DICOM patient data by approx. a factor of 10 (different scanners, kV, mA modulation, etc.). Consequently, average data size per patient would be approx. 1200-1500 MB.

SKANDION: The size of the data varies with length of scanned ROI, number of structures and number of replannings. The current estimate lies in the order of 2000-2500 MB per patient.

UNIGE: The size of the datasets are 15 G, 2*70*40 M, and 2*70*60 MB for PET/CT, chest CT, and abdomen CTs, respectively. Totally: 30 GB

UGENT-WP2: No info yet. Image data alone are at least 400 MB.

SCK-CEN: Once the dosimetry calculation systems will be deployed in their functional iterations (expected during the third year of the project), the data that will be uploaded will have a limited size, i.e., less than 1GB per day. During development, RAW data will also be uploaded for the scope of testing and fine tuning. In this case, the size of the data could reach 10 GB per day.

SU: We expect to collect 300 images per patient. With 100 patients this will be 30 000 images. Each image weighs ca 100 kB.

UJK: We expect to collect 15 pdfs per patient. With 100 patients this will be 1500 pdfs. Each pdf weighs ca 100 kB.

UGENT-WP4: No images will be collected.

QAELUM: N/A

1.5 How many (patient) examinations do you plan to upload to the platform? (this should be directly connected to information provided in each partner's Data Management Plan)

UoC: Approx. 300 patient CT examinations + 180 Planning CT + 160 pre-treatment CBCT examinations

SERGAS: 358 patients (138 lymphomas + 220 brain malignancies)

SCO: We will provide blood samples from patients with brain cancer, Hodgkin lymphoma and second malignant neoplasms (SMN) (30 patients per group). For the same patients DICOM image objects and DICOM-RT objects and modules: RT Image (CT, CBCT, PET/CT, DR), RT Structure Set (contours and volumes), RT Plan (treatment beams, prescription, patient setup...), RT Dose (dose distributions) will be included.

SKANDION: Approximately 250 patients (150 lymphomas and 100 brain tumours)

UNIGE: 140 CT exams, 12*13 (frames) PET/CT images.

UGENT-WP2: If approved: 100 patients

SCK-CEN: N/A

SU: We plan to analyse samples from 100 patients, at 3 time points.

UJK: We plan to analyse samples from 100 patients, at 3 time points, 5 pdfs per patient and time point.

UGENT-WP4: We plan to analyse samples from 100 patients, at 3 time points. Only excel files with scoring results will be stored.

QAELUM: N/A

1.6 Do you need to keep persistent and unique identifiers (such as Digital Object Identifiers) for your data objects?

UoC: No, but since data should be anonymized, and all DICOM objects should have globally unique identifiers (UIDs), the platform may need to have a technique ensuring global uniqueness (e.g., a JAVA tool to generate unique identifiers to construct DICOM compliant UIDs. In addition, DICOM and associated non-DICOM files will be correlated through their UID

SERGAS: The same as UoC

SCO: Yes, because the data should be anonymized, and all DICOM objects should have globally unique identifiers (UIDs), the platform may need to have a technique ensuring global uniqueness (e.g., a JAVA tool to generate unique identifiers to construct DICOM compliant UIDs. In addition, DICOM and associated non-DICOM files will be correlated through their UID

SKANDION: Unique identifiers are needed to mitigate the impact of data anonymisation and to keep track of such anonymised datasets.

UNIGE / UGENT-WP2: No

SCK-CEN: It will not be necessary to have DOI associated to our data

SU / UJK: No, the images/PDFs will be organised in folders, corresponding to the patient numbering system.

UGENT-WP4: No.

QAELUM: N/A

1.7 Please propose or outline any naming conventions that will be used

UoC: A Fake patient name (ID), made of a chain of several items e.g., SINFONIA project ID (i.e., 945196), SINFONIA Clinical research study ID, SINFONIA partner (i.e., site ID) Patient number (i.e., order number assigned by each site): e.g. 945196-stxxx-UOC-xxxxx.

SERGAS: We support UoC's convention. Maybe it would be useful to add the work package (WP) associated to that examination, so in this way they can be grouped by WP if necessary. e.g., 945196-WPx-stxxx-SERGAS-xxxxx.

SCO: We will use whichever naming convention is determined by the processing team, for example UoC or SERGAS.

SKANDION: We support the proposed naming convention including the project number, clinical study number, partner acronym and patient number.

UNIGE: For all datasets, We will follow any naming conventions adopted by the project for consistency. We would like to keep the institution's name in the name "UNIGE_". This is for all files, not only the "CSV" files.

UGENT-WP2: No info yet

SCK-CEN: N/A

SU / UJK: Lymphoma Patient 1 sampling 1

UGENT-WP4: Year - Patient 001 - sampling 01

QAELUM: N/A

1.8 Indicate approximately, in project month, when do you plan to start uploading your information. Also provide an estimation of how long you expect the uploading process to last.

UoC: M12-M36

SERGAS: M7. No more than two hours.

SCO: M12-M40; approx. 2 hours.

SKANDION: After month 12. The duration of the upload may depend on the amount of data to be uploaded.

UNIGE: The data volume is 30 GB, by usual internet speed, data will be uploaded in less than 10 hrs. This could be done approximately on month 10-12 of the project.

UGENT-WP2: Approval probably by summer, next step is patient selection, dataset ready earliest by begin 2022.

SCK-CEN: We should be able to start uploading the first sample data in the next 6 months of the project, i.e., around May/June 2021.

SU / UJK / UGENT-WP4: By month 10

QAELUM: N/A

2. Accessing Data

Upload, download, searching, and visualization needs.

2.1 Uploading information to the repository

How do you expect to upload the data to the repository? (Web, API, Software, Other)
Please specify if you need multiple uploading methods depending on the data object type (i.e., you may want to upload DICOM files directly from an imaging equipment or PACS and to upload PDF clinical reports through the repository web)

UoC: Web (single patient), Web (batch), API (batch)

SERGAS: Web + API

SCO: Web (single patient)

SKANDION: Web + API (the possibility of batch processing of data will allow increased flexibility and speed for data upload)

UNIGE: Web - depending on the provider or host for cloud space.

UGENT-WP2: If approved, web

SCK-CEN: If that is possible to be implemented within the repo, we would prefer to transfer data through an ad-hoc API with specific optimization ensuring stable transfer performance.

SU / UJK: Web

UGENT-WP4: Excel files will be uploaded by the Web

QAELUM: N/A

2.2 Downloading information from the repository

How do you plan to retrieve your data from the repository? (Web, API, Software, Other)
Please specify if you need multiple downloading methods depending on the data object type (i.e., you may want to download DICOM-RT files directly to a treatment equipment and retrieve a PDF clinical report from the repository web)

UoC: Web (single patient), Web (batch)

SERGAS: Web + API

SCO: Web (single patient)

SKANDION: Web + API (the possibility of batch processing of data will allow increased flexibility and speed for data download)

UNIGE: Web

UGENT-WP2: Web

SCK-CEN: Like in the upload, we would prefer to transfer data through an ad-hoc API with specific optimization ensuring stable transfer performance.

SU / UJK: Web

UGENT-WP4: -

QAELUM: Web (single patient), Web (batch): For testing/validation purposes

2.3 Specify what methods or software tools will be needed to handle the data. Will you provide documentation about this software or the software itself? (e.g., open-source software)?

UoC: We will provide open-source software to handle UoC data or propose methods and/or commercial packages. Open-source software can be uploaded to the repository if needed. Otherwise, links to UoC repository or software home sites will be provided.

SERGAS: At SERGAS, patient data are handled with the ARIA software, which is not open access. However, we aim to use open access resources following SINFONIA's aim.

SCO: We will provide open-source software to handle our data or propose methods and/or commercial packages. Open-source software can be uploaded to the repository if needed. Otherwise, links to SCO repository or software home sites will be provided.

SKANDION: Patient data comes from several sources including TPS, OIS, patient positioning software, etc. Independent software for data handling will be sought.

UNIGE: For automatic dose calculation, MATLAB codes will be provided based on a Monte Carlo transport program, here MCNP code. For deep-learning models, MATLAB-based and TensorFlow-based codes will be provided.

UGENT-WP2: No info yet

SCK-CEN: N/A

SU: ISIS software of Metasystems. Not open access.

UJK: Acrobat Reader, open access.

UGENT-WP4: -

QAELUM: N/A

2.4 **Searching for data objects**

Indicate all the criteria you expect to search for your uploaded data in the repository after uploading (name, DOI, keywords, etc.)

UoC: ID, Clinical study, examination type, gender, age, exposure site, scanner model

SERGAS: Name, ID (the *Fake patient* ID defined in 1.7), DOI, examination type, diagnostic, date, dose…

SCO: ID, Clinical study, examination type, gender, age, exposure site, scanner model

SKANDION: Name, ID, diagnostic, examination type, gender, date, age, dose

UNIGE: Name - specific special ID, special ID refers to the patient ID in DICOM header that is provided during anonymization. This is a 6-digit ID given to each patient.

UGENT-WP2: Keywords, fake name

SCK-CEN: Name, ID

SU: Name

UJK: Name

UGENT-WP4: Name, date

QAELUM: N/A

2.5 **Data remote preview before/after upload and/or download**

Do you need to preview your data? What kind of files (DICOM imaging files, reports, patient information, etc.) do you need to preview? Please specify how and when you expect to preview those files (for example, if you upload a DICOM-CT imaging file, you may want to preview it remotely after upload using your browser without downloading it to your PC, or you may want to preview a full dataset before upload and just pick the files you are really interested on being stored in the repository)

UoC: No, but DICOM/file preview would be useful in downloading data.

SERGAS: We agree with UoC, image previsualization would be useful.

SCO: No, but DICOM/file preview would be useful in downloading data.

SKANDION: Image previsualisation would be useful for reviewing data.

UNIGE: DICOM - the selected data will be collected offline. A DICOM data viewer which offers common LUTs (Look up table), and easy management of metadata is preferred. Also, an extension or application for viewing or management of excel sheets is necessary.

UGENT-WP2: Not needed

SCK-CEN: N/A

SU / UJK / UGENT-WP4: No

QAELUM: N/A

2.6 Please indicate or suggest any way that your data should be listed or organized in the repository web portal (paginated lists, organized in folders per contributor/patient/treatment, etc.)

UoC: Paginated lists as in 2.4

SERGAS: Paginated Lists

SCO: Paginated and Listed in accordance with the record format sent in point 2.4

SKANDION: Dedicated folders

UNIGE: Modality, patient's ID

UGENT-WP2: Dedicated UGent folder (retrospective study)

SCK-CEN: N/A

SU / UJK / UGENT-WP4: Organized in folders per patient

QAELUM: N/A

2.7 Please indicate how you would like to have your search results displayed in the repository web portal (per contributor, per patient, per treatment, etc.)

UoC: All ways as mentioned in 2.4

SERGAS: *Fake* patient ID defined in 1.7, contributor (partner), WP (?)

SCO: All ways as mentioned in 2.4

SKANDION: Patient ID, source/contributor, diagnosis, type of data

UNIGE: The patient's ID and the modality.

UGENT-WP2: All

SCK-CEN: N/A

SU / UJK: Per patient

UGENT-WP4: Per patient and biological endpoint (e.g.: MN assay, gammaH2AX…)

QAELUM: N/A

3. Data Security

Data anonymization and access rights.

3.1 **Anonymization**

EU and national privacy regulations may oblige partners to have any privacy-sensitive information purged (anonymized) before uploading to the repository. Do you need any kind of assessment about the anonymization performed on your data before uploading?

UoC: We will provide fully anonymized patient data to be uploaded to the repository

SERGAS: Sure, we believe it would be useful if the platform carries out some kind of assessment or evaluation about the anonymization before uploading, to avoid errors and maximize the standardization of the platform for data coming from different partners.

SCO: We will provide fully anonymized patient data to be uploaded to the repository

SKANDION: Patient ID, source/contributor, diagnosis, type of data

UNIGE: Yes

UGENT-WP2: Anonymized data will be provided

SCK-CEN: There will not be any need of anonymization, as data will not include privacy-sensitive information

SU / UJK / UGENT-WP4: No, patients will be numbered

QAELUM: N/A

3.2 **Pseudonymization**
Do you need to keep a track of the anonymized information after uploading? If so, please specify how you expect to identify it.

UoC: By (fake) patient name (ID)

SERGAS: By the *Fake* patient ID defined in 1.7

SCO: By patient name (ID)

SKANDION: Naming convention with possibility of keeping a key at the local institution for later data retrieval.

UNIGE: We will give each patient a specific ID (similar to 2.4)

UGENT-WP2: Fake name

SCK-CEN: Pseudonymization will be performed by the dosimetry system, so it is not required to be implemented at the level of the repo

SU / UJK: No

UGENT-WP4: Yes, through data link with biobank Radiobiology of UGENT

QAELUM: N/A

3.3 **Management of access rights**
Who will be authorized to access your data? (for example, only you, your group/hospital, project partners, general public, etc.)

UoC: Project partners

SERGAS: Project partners

SCO: Project partners

SKANDION: Project partners

UNIGE: Only project partners

UGENT-WP2: Depending on approval

SCK-CEN: We will be the only users accessing the data

SU / UJK / UGENT-WP4: Project partners

QAELUM: N/A

3.4 What are the expected lifetimes of those access rights? (specific stages of the project, the whole project, indefinitely, patient treatment duration, etc.)

UoC: Project partners – whole project

SERGAS: the whole project

SCO: Project partners – whole project

SKANDION: The whole project

UNIGE: 10 years

UGENT-WP2: Depending on approval, project duration, project partners

SCK-CEN: Indefinitely

SU / UJK / UGENT-WP4: Project duration

QAELUM: N/A

3.5 Up to what extent do you need to adjust the aforementioned authorization rights? (for example, you may want to access indefinitely to the whole dataset of a clinical case you manage, but allow to access to general public during a period of time)

UoC: DICOM and non-DICOM files (project partners), algorithms and results public after project finishes

SERGAS: as it will be established in next meetings

SCO: DICOM and non-DICOM files (project partners), algorithms and results public after project finishes

SKANDION: To be decided later

UNIGE: No change in policy, only project partners

UGENT-WP2: No info yet

SCK-CEN: N/A, data will be accessed only by SCK CEN

SU / UJK / UGENT-WP4: Indefinite access to the whole dataset

QAELUM: N/A

4. AI Algorithms

First survey about your needs regarding the integration of your eventual AI algorithms in the repository. You will receive a technical questionnaire if interested at a further stage of the project

4.1 Do you plan to develop AI algorithms? Do you expect to integrate your algorithms in the platform to be developed?

UoC: We will develop AI and machine learning algorithms for specific dose calculation tasks as described in WP2 task 2.1. UoC plan to upload the algorithms to the platform and possibly integrate algorithms with examples and results if an available collaboration and visualization platform exists (e.g., an online data science notebook with data analysis flow and team collaboration)

SERGAS: -

SCO: No, We will use algorithms developed within the project

SKANDION: To be decided later

UNIGE: Yes

UGENT-WP2: No

SCK-CEN: Yes, both in the stage of data generation (i.e., before upload) and for processing and data analysis. If possible, these models could be integrated on the repo, proven that sufficient computational resources can be provided.

SU / UJK / UGENT-WP4: No

QAELUM: Yes, in WP 6 (but not intended to be integrated in the data repository).

4.2 If you plan to develop AI algorithms, will you need any kind of remote processing (training and/or inference) within the platform? What kind of images will you be dealing with?

UoC: No

SERGAS: -

SCO: No

SKANDION: To be decided later

UNIGE: No

UGENT-WP2: No

SCK-CEN: Yes, we would benefit from performing both the training and the prediction of the AI models within the platform.

SU / UJK / UGENT-WP4: No

QAELUM: N/A

4.3 If you plan to develop AI algorithms, will your algorithms (code, etc.) be freely accessible to everyone or only to authorized users? (as in question 3.3)

UoC: Project partners until end of project, public after project finishes

SERGAS: -

SCO: We do not plan to develop AI algorithms

SKANDION: To be decided later

UNIGE: Only to authorized users

UGENT-WP2: No

SCK-CEN: Only authorized users from our institute.

SU / UJK / UGENT-WP4: No

QAELUM: N/A

5. Other
Please provide any other initial requirements for the repository shared infrastructure and add a level to the included ones

Below you will find some typical functional and non-functional requirements for research repositories. Your suggestion will aid WP5 group and repository developers to identify your data storage needs and subsequently aid the developing process. Please provide information about which of the proposed requirements must/should/may be included in the platform and add additional ones (if not yet covered by previous questions).

The level terminology is included at the end of this document.

Table B.1 and Table B.2 below collect the partners' answers for Table A.1 and Table A.2, respectively. Table B.1 is for functional requirements and Table B.2 is for non-functional requirements. The structure of both tables is the same. Requirements are described each in a separate numbered row. The answers given by each partner are represented in a column titled with the name of the partner. To know the level assigned by a specific partner to a specific requirement, the reader must look for the row corresponding to the requirement and then go to the cell in that row that is also in the column corresponding to the specific partner. A color code has been set for the levels described in the terminology included at the end of the questionnaire (see Annex A): absolute requirements (MUST, MUST NOT) are highlighted in red, recommendations (SHOULD, SHOULD NOT) are highlighted in yellow, and optional features (MAY) are highlighted in green. The purpose of such a color code is to provide the reader with a mechanism for identifying quickly which requirements are the strictest and the stake they represent in the whole analysis of the repository needs.

FUNCTIONAL REQUIREMENTS

No.	Description	Comments/ Examples	UoC	SERGAS	SCO	UNIGE	UGENT-WP2	SKANDION	SCK·CEN	SU	UJK	UGENT-WP4	QAELUM
1	Identifiers	e.g. support of PID	MUST	MUST	MUST	MUST	-	MUST	MAY	MUST	MUST	MUST	MUST
2	Authorization and authentication	Single sign-in or multiple authentication method	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST
3	Metadata	Support different metadata schemes, extended metadata, labelling by data owner, metrics for metadata quality assessment, support XML storing, search engine	MUST	MUST	MUST	MUST	MUST	MUST	MAY	MAY	MAY	MAY	MUST
4	Data access	Support data download, download API, manual and automatic search, multiple data versions	SHOULD	MUST	SHOULD	SHOULD	SHOULD	MUST	SHOULD	MUST	MUST	MUST	MUST
5	Data organization	Logical naming	SHOULD	SHOULD	MUST	SHOULD	SHOULD	SHOULD	MAY	SHOULD	SHOULD	SHOULD	SHOULD
6	Integration / Interoperation	Support for different files and OS, support for programming language (Python, R), support for analysis and visualization tools, email support for submission and verification, Interoperation among individual data	SHOULD	SHOULD	MUST	MUST	SHOULD	MUST	SHOULD	SHOULD	SHOULD	SHOULD	MAY
7	User interface	Provide seamless and consistent interface. Allow file uploading for each metadata field, when needed, Allow search by community, organization, Allow enabling/disabling all internal and external data services;	SHOULD	SHOULD	SHOULD	MUST	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD
8	User interface	Provide a sandbox where users can test their data before submission, allow specific users (project administrator, etc.) download specific data	MAY	SHOULD	SHOULD	MAY	MAY	MAY	MAY	SHOULD	SHOULD	SHOULD	MAY
9	Preview	Ability to visualize data before uploading to the platform	MAY	MAY	MAY	MAY	MAY	SHOULD	MAY	MAY	MAY	MAY	MAY
10	Integration of the different tools developed by other partners	Ensure the correct compatibility between the different formats of the tools developed by all partners.	MAY	-	SHOULD	MUST	SHOULD	MAY	SHOULD	MAY	MAY	MAY	MAY
11	Availability of fully annotated data	Where and when the data were collected	MAY	SHOULD	SHOULD	SHOULD	MAY	MAY	MAY	SHOULD	SHOULD	SHOULD	SHOULD
12	Ranking based on metadata	Connection of data with type of patients / institution / …	MAY	SHOULD	MUST	MAY	SHOULD	MAY	MAY	MAY	MAY	MAY	SHOULD
13	Cross-referencing of data	No duplication of data	SHOULD	SHOULD	SHOULD	MAY	SHOULD	SHOULD	MAY	MAY	MAY	MAY	MUST
14	Sharing data in a collaborative environment	Easy access to other partners' data (anonymized)	SHOULD	SHOULD	SHOULD	MUST	MAY	MAY	MAY	SHOULD	SHOULD	SHOULD	SHOULD

Table B.1: Partners' answers for Functional Requirements

NON-FUNCTIONAL REQUIREMENTS													
No.	Description	Comments/ Examples	UoC	SERGAS	SCO	UNIGE	UGENT-WP2	SKANDION	SCK·CEN	SU	UJK	UGENT-WP4	QAELUM
1	Security	React to unauthorised, accidental, or unintended usage and provide access only to legitimate users	MUST	MUST	MUST	MUST	MUST	MUST	SHOULD	MUST	MUST	MUST	MUST
2	Capacity / scalability	Meet the agreed capacity requirements and designed to accommodate increased data traffic, storage volumes, workloads, and users	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST	MUST
3	Volumetric / performance	Meet the agreed capacity requirements	MUST	MUST	MUST	MUST	MUST	MUST	MUST	SHOULD	SHOULD	SHOULD	MUST
4	Location	HPC system	-	SHOULD	SHOULD	SHOULD	MAY	SHOULD	MAY	SHOULD	SHOULD	SHOULD	-
5	Availability/failover	Uptime 24/7/365, minimizing downtime	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD	MAY	MUST	SHOULD	SHOULD	SHOULD	SHOULD
6	Recoverability	Regular backups/snapshots	-	-	MUST	SHOULD	SHOULD	MUST	SHOULD	SHOULD	SHOULD	SHOULD	MUST
7	Compliance with privacy and protection standards	EU GDPR, other legal requirements in users countries Partner uploading data is responsible for the anonymization process	MUST	MUST	MUST	MUST	MUST	MUST	SHOULD	SHOULD	SHOULD	SHOULD	MUST
8	Deployment	Release a new minor or patch version, replacing the previous minor or patch version	SHOULD	MUST	SHOULD	SHOULD	SHOULD	SHOULD	MAY	SHOULD	SHOULD	SHOULD	SHOULD
9	Serviceability	Be designed so that technical support personnel are able to monitor and manage it in operation	SHOULD	SHOULD	SHOULD	SHOULD	MAY	SHOULD	MAY	SHOULD	SHOULD	SHOULD	SHOULD
10	Test environment	An alternative (staging) testing platform may be designed for testing and assurance before main platform is launched	-	-	-	SHOULD	SHOULD	MAY	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD
11	Number of simultaneous accesses greater than number of partners	The platform does not crash when there are more than X visitors (the minimum of X being the number of project partners).	MAY	SHOULD	MAY	MUST	MAY	SHOULD	SHOULD	MAY	MAY	MAY	SHOULD
12	Uploading time lesser than 2 minutes/500MB	High upstream speed depending on the size of the uploaded files	-	-	MAY	MAY	MAY	MAY	MAY	MAY	MAY	MAY	-
13	Human-friendly interfaces	Provide a simple, user-friendly interface for uploading and downloading data	SHOULD	SHOULD	MUST	MUST	SHOULD	SHOULD	MAY	SHOULD	SHOULD	SHOULD	SHOULD
14	Storage needs	Sufficient storage must be foreseen (avoid slow use of the data repository due to 'overloaded' or inefficient data storage)	SHOULD	MUST	MUST	MUST	MUST	MUST	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD
15	Data security	Protection from malware attacks	SHOULD	MUST	MUST	MUST	MUST	MUST	SHOULD	SHOULD	SHOULD	SHOULD	MUST
16	Data preservation	History of data versions	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD	SHOULD	MAY	SHOULD	SHOULD	SHOULD	SHOULD

Table B.2: Partners' answers for Non-Functional Requirements