



Dicoogle Open Source: The Establishment of a New Paradigm in Medical Imaging

Rui Lebre^{1,2} · Eduardo Pinho³ · Rui Jesus^{1,3} · Luís Bastião³ · Carlos Costa²

Received: 15 April 2021 / Accepted: 16 September 2022 / Published online: 6 October 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022, corrected publication 2022

Abstract

The rapid and continuous growth of data volume and its heterogeneity has become one of the most noticeable trends in healthcare, namely in medical imaging. This evolution led to the deployment of specialized information systems supported by the DICOM standard that enables the interoperability of distinct components, including imaging modalities, repositories, and visualization workstations. However, the complexity of these ecosystems leads to challenging learning curves and makes it time-consuming to mock and apply new ideas. Dicoogle is an extensible medical imaging archive server that emerges as a tool to overcome those challenges. Its extensible architecture allows the fast development of new advanced features or extends existent ones. It is currently a fundamental enabling technology in collaborative and telehealthcare environments, including research projects, screening programs, and teleradiology services. The framework is supported by a Learning Pack that includes a description of the web programmatic interface, a software development kit, documentation, and implementation samples. This article gives an in-depth view of the Dicoogle ecosystem, state-of-the-art contributions, and community impact. It starts by presenting an overview of its architectural concept, highlights some of the most representative research backed up by Dicoogle, some remarks obtained from its use in teaching, and worldwide usage statistics of the software. Finally, the positioning of Dicoogle in the medical imaging software field is discussed through comparison with other well-known solutions.

Keywords Dicoogle · PACS · Medical imaging · DICOM · Teaching · Telehealth-care

Introduction

Nowadays, nearly all scientific advancements are backed by data. In the field of healthcare, this scenario is no different. Medical imaging data has become essential and almost mandatory in clinical practice [1, 2]. Over the years, the data generated in medical environments has increased substantially. The ever-increasing amount of data is justified by

the development of new and innovative modalities and continuous improvements in digital medical imaging systems. Additionally, central regulations usually make it mandatory to store the data gathered by modalities for several years, increasing the required capacity.

Picture Archiving and Communication System (PACS) is the concept to designate a set of distinct technologies responsible for the acquisition, archiving, distribution, and visualization of medical digital images using a network workstation for diagnosis or revision [3, 4]. The core component is the storage server, also designated as an archive, which receives, stores, and distributes the images from modalities and to terminals.

The success of PACS over the years is based on implementation of the Digital Communications in Medicine (DICOM) standard. This standard is widely accepted by the medical imaging community [5] and defines specifications about the organization of the data in each file, the way of storing those files, and how the communication between medical imaging devices is carried out [6]. In this way,

This article is part of the Topical Collection on *Education & Training*

✉ Rui Lebre
ruilebre@ua.pt

¹ University of A Coruña, Campus de Elviña, A Coruña, Spain

² University of Aveiro, Campus Universitário de Santiago, Aveiro, Portugal

³ BMD Software, PCI - Creative Science Park Via do Conhecimento, Aveiro, Portugal

DICOM can guarantee the interoperability and compatibility of proprietary equipment and information systems [7]. When implementing DICOM, PACS ensures normalized data formats and communication between each component of the network [5].

These technological advancements have proved to be crucial to ensure workflows where the different actors in the medical imaging field may be geographically distant. For instance, physicians practicing at home or using the departmental intranet, sharing data backed by telehealth services.

The establishment of those data-rich environments generated opportunities for the development and spread of tools specially designed for medical image acquisition, data analysis and other innovative systems [8]. The growth in the quantity of data produced in the healthcare field generates challenges. The diversity and complexity of the image data and associated metadata leads to performance problems when querying and retrieving. There are also concerns about regulating access to the data in centralized installations that may be supporting different institutions.

In the past few years, these key issues have been addressed by research and described in the literature [9–14], presenting innovative ideas to search and fetch relevant information. The evolution described resulted in more complex environments, requiring special training to interact and get the best out of them.

To address the challenge of building a PACS system that is both easy to manage and develop but also addresses all the mentioned challenges, the UA.PT Bioinformatics and Computational Biology research group based at the University of Aveiro has been developing the Dicoogle PACS archive¹ during the last decade. This open-source PACS archive became a tool to tackle and solve complex state-of-the-art problems, such as content-based image retrieval, a multimodal search engine for medical images, automated labelling systems, integration of machine learning tools, and support for distributed and scalable environments. Its flexible architecture facilitates the extension of its core capabilities, placing Dicoogle as a tool suitable for industrial usage, research, and teaching. Dicoogle seeks to provide a solution for regular users and developers by providing a detailed view of the PACS and DICOM technologies operations, and by providing means to expand the default features.

This paper presents a description of Dicoogle most relevant aspects, from its architecture to its main features. It emphasizes the characteristics that fit best in research and education activities. It presents the Dicoogle Learning Pack, which helps to understand the fundamental concepts of medical imaging informatics in the Dicoogle ecosystem, facilitating the learning process of how to use a modern PACS archive. To clarify its usage potential better and help to

understand its positioning in the community of open-source software for medical imaging, its characteristics are compared against other well-known solutions. Finally, we point out some of the most relevant projects based on Dicoogle, not only carried out in research activities but also in the academic and industry context.

Related work

PACS for research and teaching

Traditionally, the PACS-DICOM repositories were not much used in the academic field, for teaching purposes and research, as they did not offer easy-to-use functionality or the ability to extend them in a decoupled way. Commonly, the project time frame was not compatible with the PACS-DICOM development effort. The setup costs, the high content density, and the complexity of the DICOM protocol, which resulted in steep learning curves, were other major barriers. However, the creation and usage of PACS in the academic environment became inevitable [15]. Multiple authors [16–19] made efforts to create teaching resources linked with institutional PACS. Lim and Yang [18] developed a computer-based radiology simulator to create a learning tool to help prepare students.

In the research field, over the years, notable concerns regarding the flexibility of PACS systems have been identified in the literature. Doran et al. [20] described a prototype of a framework, based on XNAT, with data selection and archiving tools. In [21], the authors describe a set of libraries and applications which facilitate the handling of DICOM files. In the same line, the work in [22] describes the dcm4chee archive, a PACS archive based on a modular architecture. Woodbridge et al. [23] present MRIDb as a tool to manage MRI datasets. More recently, Jodogne [24] described an open-source PACS archive whose goal is mainly to stimulate cooperation and sharing of medical imaging workflows in hospitals and academia.

PACS for collaborative telehealth

During the last decades, science has made efforts regarding the implementation of innovative collaborative platforms [25, 26]. The literature reports several benefits of collaborative work, teaching and research [27, 28]. In the scope of telehealth collaborative platforms supported by PACS archives, an in-depth, state-of-the-art analysis was made. Daniel et al. [29] state that collaborative forms of work are only possible via accepted and widespread standards of medical informatics.

In [30], the authors report the architecture and implementation of an image-enabled electronic healthcare record system (i-EHR), where the goal was to provide images available in EHR rather than only radiology textual reports. The

¹ Online: www.dicoogle.com

solution allows the sharing of medical information across several hospitals in the Shanghai region. The authors used the IHE XDS-I integration profile to enable a system for collaborative imaging diagnosis. The results show that the system was well accepted by healthcare providers and physicians.

In [31], the authors describe Cytomine, a web-based platform that enables collaborative analysis of digital pathology images. The article claims that the proposed platform can trigger the collaboration between distributed groups of scientists, by supporting sharing of studies and reviewing semantic and quantitative information related to the images.

Maglogiannis et al. [32] presented a collaborative platform enabling physicians to do telework, even when geographically distant from each other. The authors point out the advantage of using pure web technologies as they can be accessed anywhere at any time from any device, from a small smartphone to a workstation.

Díaz et al. proposed, in [33], a web-based telepathology system for collaborative work. However, the authors do not report DICOM compliance, which would enable communication between multiple PACS equipment, from the modalities to the archive. Additionally, this solution does not address collaborative academic scenarios.

In [34], Pedrosa et al. presented a collaborative platform to annotate diabetic retinopathy studies. The work allows identification of the regions of interest (ROI) within the image as well as annotation of other metrics by answering a form and free-text reporting. The goal of this platform is to gather a dataset for machine learning algorithms to facilitate the screening process of diabetic retinopathy.

In [35], the authors deployed a collaborative viewer for digital telepathology. The solution is web-based and backed by DICOM Web services, implemented as Dicoogle plugins.

In recent years, there has been a trend to deploy telehealth collaborative platforms using web technologies [32–35] that are easy to use and provide anytime access to the services, making use of the Internet and common Web browser. Moreover, they reduce the time to install and train practitioners.

Framework architecture and services

Dicoogle paradigm

Dicoogle's initial concept was to create a flexible open-source PACS archive with an enhanced query and retrieve engine for multi-modal medical imaging data, i.e. pixel, DICOM metadata, and reports. The Dicoogle PACS archive follows an extensible architecture [36, 37] supported by the addition of independently developed modules to its core components. The core features seek to replace the traditional database models with an index and retrieval framework that allows free-text queries. Extension of the core features is

achieved by developing plugins, and the platform overcomes the limitations of the traditional DICOM services [38]. Plugins are responsible for indexing and storing metadata, and relevant imaging pixel-data, or extracting and accommodating automatic information. The extension of the features is backed by the Dicoogle Learning Pack, which includes a Software Development Kit (SDK), documentation and code examples. The SDK is the component that facilitates communication between the Core and the plugins, enabling developers to extend Dicoogle with new features without making changes to the core platform.

Over the years, Dicoogle features, and its extensible architecture enabled the easy development of innovative technology in the research and healthcare industry, with many use cases reported in the literature [8, 34, 35, 39–42]. The usage of pluggable architecture is very relevant nowadays, given the need to improve, monitor, and measure the efficiency of systems. This is especially true in the PACS universe as there is a constant need to integrate the advancements in technology applied to healthcare.

In the case of Dicoogle, the extension of its capabilities only by developing small pieces of software based on an SDK enables, for instance, the extraction of knowledge from the produced medical images or the analysis of healthcare quality indicators. In fact, studies in the field of data mining DICOM datasets [43, 44] reported the successful use of Dicoogle to get their results. Also, [11] refers to the use of Dicoogle as an archive for exploiting multi-modal information retrieval methods. Finally, work has been done to enable telehealth and collaborative solutions for Dicoogle [34, 35, 40–42]. Supporting such use cases in conventional non-modular systems is not possible without a deep understanding of the software and all its pieces.

Dicoogle architecture creates an isolation layer between the plugins and the core software which enables developers to extend its functionality easily without needing to interact or understand the whole existent solution, core components or possible third-party modules. This modular architecture is then more suited to research and classroom environments where there is a need for fast prototyping and deployment of new features.

Architecture and services

The architecture of Dicoogle is presented in Fig. 1. As can be seen from the top down, Dicoogle provides a Service Interface. This layer enables communication with DICOM compliant hardware and software, such as Storage or Query/Retrieve network services. Those services are backed by the Dicoogle Runtime Framework, which relies on third-party components such as *ImageIO*, *dcm4che*, and *jetty*. Below the runtime framework, Dicoogle SDK provides the application interfaces for the plugins to connect to the Dicoogle

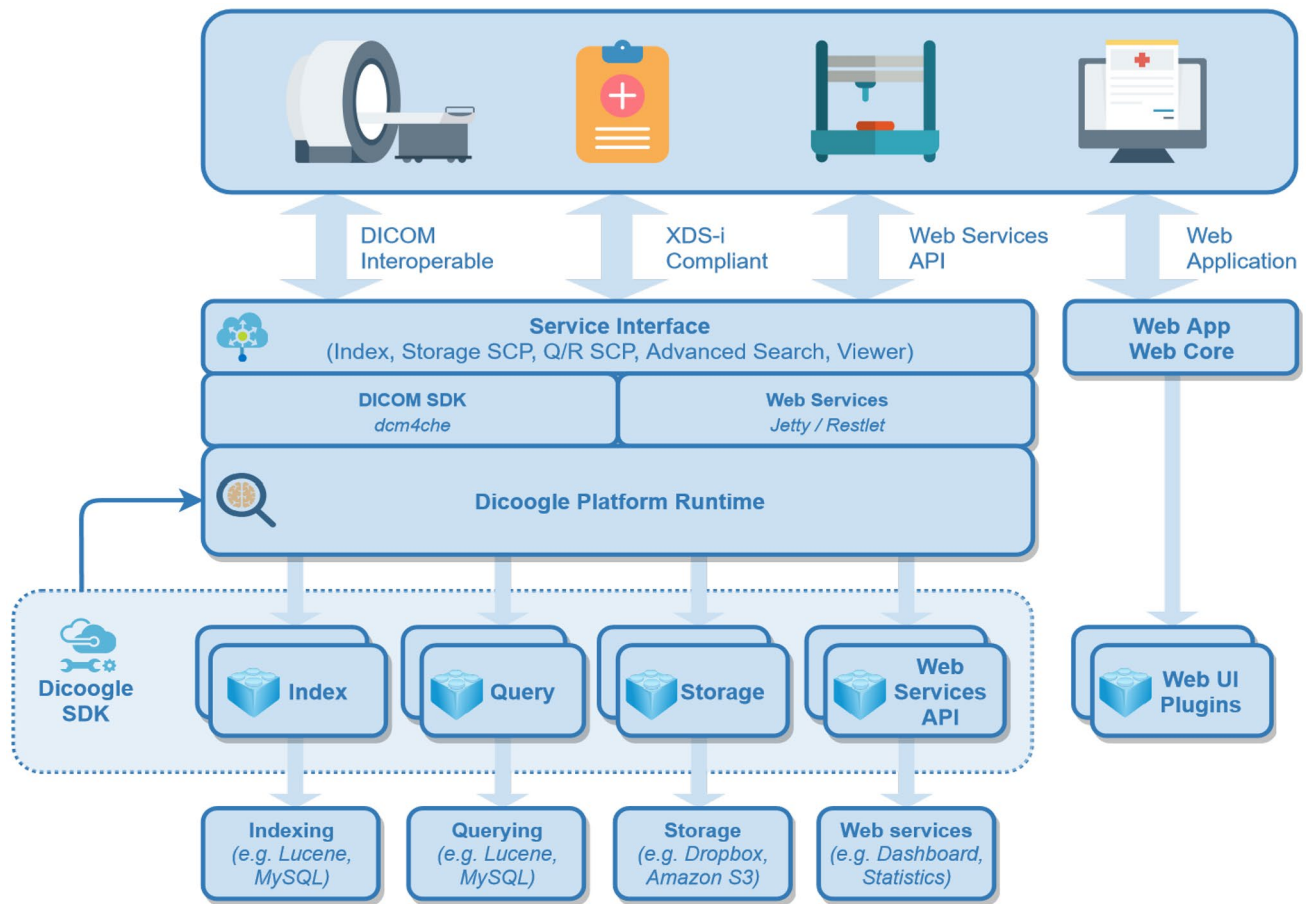


Fig. 1 Dicoogle Architecture

core system. As also presented, Dicoogle splits the plugins into four types: *Index*, *Query*, *Storage* and *Web Services*. It also provides a different type of plugin, the *Web UI plugin*, which allows developers to extend the graphical interface. Each of these categories is tied to a specific application programming interface (API), different for each type, which is implemented by plugins. Through these interfaces, the plugins provide operations that are orchestrated by the core Dicoogle platform.

The plugin's life cycle is controlled by Dicoogle core. During boot time, it scans the default plugins directory, identifies each plugin, loads the configuration files, and boots the plugins during run-time. Dicoogle core sees the plugins as completely independent from each other, being only accessible via the respective interfaces. In the case of the need to share state or resources between plugins, the dependencies are bundled inside a Plug-in Set, a class that represents a set of plugins and serves as an entry point and management entity to the overall structure of the extension. Here, plugins of several categories are aggregated into one functionally consistent unit, thus simplifying the development and deployment process.

The SDK is the bridge between the plugins and the Dicoogle Core component, as it provides the required interfaces to interconnect modules. It simplifies the process of plugins development by establishing specific APIs that must be followed for the features to be loaded during the Dicoogle boot time.

The development of a new Dicoogle plugin usually follows a well-defined pipeline. When an extension to Dicoogle default services is considered, developers should follow some steps to comply with best practices:

1. The design of the feature idea that is closely related to the PACS repository;
2. Creation of a new Java project for the base plug-in set, using a bare-bones template. In practice, it can start by copying a sample provided in the Learning Pack;
3. If the new feature will support additional sources of information, it is usual to create an *indexer* to process that information from incoming medical images, and a *query* provider to expose that information to the system;
4. When the plugin needs to support a new media storage device, for instance, remote storage at the cloud, a *storage*

plugin must be created for the intended storage provider. The same type of plugin can also be used to construct pre-processing pipelines, which can apply a pipeline to the files and delegate the storage to another plugin;

5. *Web services* are the most flexible, as they enable the web application, and other client programs, to fetch new kinds of information from the server and perform new operations;
6. To extend the main user interface, *web user interface* plugins should be considered instead.

Dicoogle Learning Pack provides in-depth instructions to assist the development of Dicoogle plugins, as we discuss later in "[Deploying Dicoogle](#)". The resources are available online at its web portal.²

Dicoogle provides a modern single-page application (SPA) user interface, which can be delivered to local and remote users without a previous installation process on the client machine. This approach empowered Dicoogle to be accessed via a web browser from any device such as a workstation, tablet, or smartphone. Moreover, the characteristics of SPAs assure increased efficiency and responsiveness.

Dicoogle's flexibility also accommodates extension of the front-end interface. The flexible plug-able architecture for web components allows extension of the core functionalities of the web app, by including support for new and innovative use cases.

The Dicoogle Web Core establishes an API for loading and rendering web UI components during the run-time in the web browser, thus, enabling the exposure of new features without the need to rebuild the whole front-end source code. These features allow extension of the web app regarding, for instance, slider menus, buttons on search result menus, and settings tabs.

Putting this architecture into practice involved the development of three main components:

1. The creation of a set of web services in the Dicoogle main application. These were mandatory to fetch the web-based plugins and associated configuration data;
2. The Web Core component, which runs in the client's web browser to fetch and expose the new user interface(s);
3. A tool to promote the creation of new web UI plugins.

The plugins loaded to the user interface are scoped to a container named slot. Each slot has its own properties and behaviour. They may be of the type settings—available in the Management section; result-entry—created in each row of a search result and allowing an action to be performed on the selected item; result-batch, scoped to the whole result set of a search, allowing the visualization, manipulation, and export of the result set. After the login page, the web application core fetches the list of web UI descriptors available

for that specific user. When the list of Javascript modules is received, the web core fetches, one by one, the slots and renders them on the web page.

Extension interfaces

Index and query

Dicoogle provides indexing and querying features, as the common PACS archives, to store and query over DICOM metadata. Despite providing a default index and query plugin based on Apache Lucene, Dicoogle is not bound to this single implementation and can have multiple implementations, with distinct database technologies, serving at the same time. This is achieved with the Query and Index plugin pair, which usually go together. These kinds of plugins are triggered in specific situations:

- Dicoogle triggers every index plugin when a new DICOM object reaches the interfaces; the index plugins process the data to be accessible in the future;
- Query plugins are triggered every time Dicoogle receives a C-FIND request or a request to its REST API.

Storage

Data storage is one of the main goals of a PACS archive. By default, a storage plugin is included in Dicoogle. It enables the storage and retrieval of DICOM files to or from the local file system where the Dicoogle instance is running. Nevertheless, as the technology evolves and more patient data is acquired, the server's local file system has become inadequate in some use cases. As a result, medical imaging storage requirements have become increasingly demanding, and delegating storage to cloud services, for instance, has been thoroughly discussed in recent years [44–46].

The storage plugins are triggered mainly by the occurrence of two situations. Firstly, every time the PACS archive receives a C-STORE request and handles the persistence of the DICOM object, Dicoogle runs a specific set of operations that each plugin must implement to match the plugin API. Secondly, storage plugins are triggered every time a C-MOVE is requested. Dicoogle calls storage plugins to retrieve the requested DICOM object and the plugins must access the location where the file is located for subsequent retrieval.

The Dicoogle API for storage plugins has two main goals. Firstly, it creates an abstraction of the storage technology so a developer can build a different type of storage in an arbitrary location that is not the default local storage, such as cloud services or document-based databases. Secondly, the common storage API makes it possible to read DICOM objects regardless of their origin or underlying technology.

² Available at: <https://bioinformatics-ua.github.io/dicoogle-learning-pack/>

Furthermore, it enables pre-processing of the data before storing them. Among the possible use cases are anonymization or encryption needs.

Web services

This type of plugin allows extension of the default REST API of Dicoogle. The ability to add Jetty extensions and handlers provides support to other kinds of services, such as SOAP, useful when implementing some IHE profiles [47]. This flexible solution may be achieved using one of two modes:

1. The most common mode is to attach Jetty³ servlets using the Jetty interface available in the Dicoogle SDK. It allows the design of new web API endpoints and services.
2. Dicoogle SDK offers an alternative method to extend the web API using a RESTlet API. It offers a more straightforward way to develop web services, although more limited, by using annotations.

Web UI

Dicoogle web UI offers a plug-able architecture that can be extended by the integration of external components. This type of plugin requires the developer to specify the type of plugin to be attached in the correct “slot”. The web UI plugins are inserted in one of four types, each one attached to different parts of the web app:

- *menu*: allows extension of the left drawer menu by creating a new entry. Once the user navigates to that entry, a new page is shown;
- *result-option*: creates a new entry on the “Advanced Options” in the search result table. It allows extending the default actions that may be executed on a particular result of the result list;
- *result-batch*: allows extension of the actions that can be performed on a list of results. It adds a button on the bottom of the result table that may implement, for instance, the exporting of the results to a CSV file;
- *settings*: attaches new sections to the management page, under “Plug-ins & Services”.

Documentation pack

Medical imaging informatics is a major subject in healthcare, and teaching is becoming frequent, trying to address the increasing service requests. The growing multidisciplinary

demands have an impact on the education provided, with an increasing number of courses combining medical informatics and radiology.

Therefore, teaching students, developers and researchers how to use an open-source PACS archive in a medical imaging informatics context is an essential counterpart to development of the actual software. The Dicoogle Learning Pack seeks to soften the learning curve of users with no prior knowledge about medical imaging informatics and research PACS archives, namely, Dicoogle. The resources were designed to be as simple and objective as possible. The learning resources were deployed to minimize and facilitate the open-source PACS and DICOM standard learning curve.

The Dicoogle Learning Pack, Dicoogle Web API definition, and Dicoogle Javadoc⁴ are available as an open-access static site hosted on GitHub Pages. The documentation is also open-source and is therefore freely available and suitable for modification and contributions. Dicoogle users are invited and encouraged to provide their feedback in the GitHub repository issue tracker. This method allows the community’s active participation in the discussion of bugs, and innovative new features, improving the development of Dicoogle and its documentation. In fact, since the launch of the resources, Dicoogle users have interacted with the team requesting new guides or asking questions. Direct interaction causes greater proximity between users and Dicoogle maintainers, allowing continuous adaptation of the development strategy to meet needs.

Deploying Dicoogle

Installation and configuration guidelines for Dicoogle can be found on the documentation page.⁵ Dicoogle is built on the Java platform, making it a very portable solution. The whole application, including the web server, is contained in a single jar file. Each Dicoogle plugin will have its own jar that needs to be placed in the plugin’s directory which, by default, should be named “Plugins”. During the initialization process, Dicoogle will load all plugins contained in this folder. A download package, containing the application jar file and two basic plugins for indexing and storing DICOM files is available on the project website.⁶

Dicoogle possesses two installation methods, manual and Docker. To run Dicoogle manually, the user will need to have installed an updated version of the java virtual machine environment. The “dicoogle.jar” file, the main application file, needs to be placed in the same directory as the plugins

³ Jetty: <https://www.eclipse.org/jetty/>

⁴ Dicoogle Resources: <https://bioinformatics-ua.github.io/dicoogle-api>

⁵ <https://bioinformatics-ua.github.io/dicoogle-learning-pack/docs/configuring/>

⁶ <https://dicoogle.com/downloads>

folder. From here, the command “java -jar dicooogle.jar” can be used on a command line terminal (CLI) to start the application.

An alternative installation process, using the popular Docker framework is also available. The Docker Dicooogle image is freely available and provides a simple installation with the same indexer and storage plugins available on the linked download page. This method requires users to have Docker installed on the host machine. Using the “docker pull” command, users can download the Dicooogle image and create a container that will host the application. This is the preferred method of installation since all the dependencies are contained within the image and no configuration is required from the user. By default, Dicooogle’s web server will be available at port 8080 and the storage and query DICOM services will be available at ports 6666 and 1045 respectively, under AETitle DICOOGLE-STORAGE. When using the Docker container, the user must properly configure the port mappings to access these services. To install new plugins, or update existing ones, the jar files under the plugin’s directory must be updated and the application restarted in order to reload the changes.

Use cases and scenarios

Since its conception, Dicooogle has been used by a large range of people both in industry and academia. For instance, at the industry level, BMD Software⁷ company has been using and developing features for Dicooogle since 2015. The experience gathered by this company while providing innovative solutions for telehealth care is detailed in “[TeleHealthcare@Cloud](#)”. At the academic level, Dicooogle supported more than twelve PhD students who made use of the Dicooogle framework in their research at the University of Aveiro.

This section will present the experience of BMD Software in teleradiology in distinct continents and multiple contexts. Additionally, we will cover Dicooogle as the backbone of several research projects, including collaborative platforms. Finally, Dicooogle and its Learning Pack will be introduced as a tool for teaching students in academia.

TeleHealthcare@Cloud

BMD software team is nowadays one of the biggest Dicooogle developer forces. The company has been using Dicooogle as the cornerstone of its cloud-based PACS/teleradiology platform (i.e. PACScenter⁸), and also in its DICOM toolset

environment for common daily workflow. One of the advantages of being in the industry backed by the Dicooogle open-source platform lay in creating many opportunities, such as establishing relationships between customers to have an open platform, as well as including them in the design and development of their own needs and integration assets. Another important advantage is the possibility to integrate with third-party entities without requiring deep knowledge of the DICOM world. The flexibility of the plugin strategy makes it possible to have different teams developing different and independent components, which can then enter production systems sharing the same runtime.

The PACScenter archive server is an ecosystem of public (open-source) and proprietary plug-ins. To provide a higher-level integration mode, its services are available also through an advanced Web Services *middleware*, allowing the development of other components in a technology-agnostic fashion. Moreover, BMD members also pushed Dicooogle and PACScenter wrappers for different languages, which include client libraries for Java, JavaScript, and Python, thus facilitating further integration with the platform.

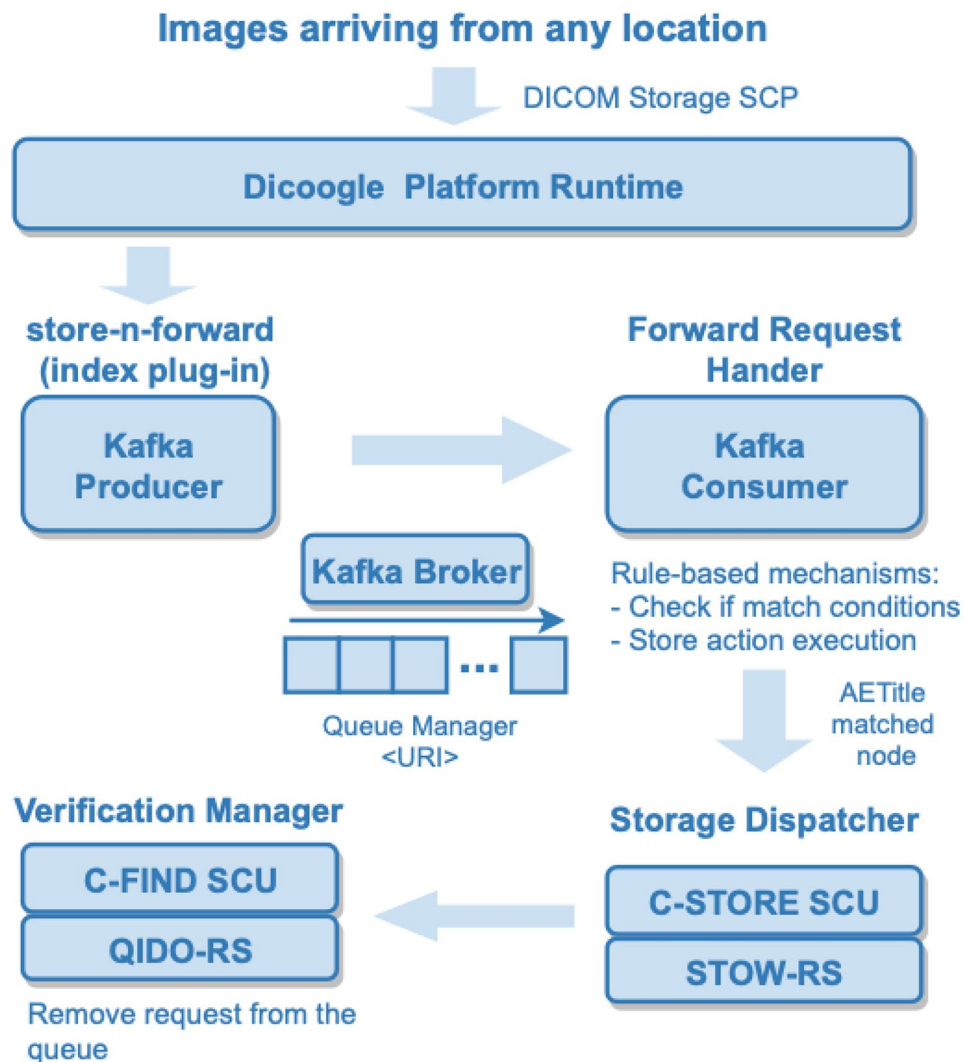
A technical example of a simple and important plugin is the study notification plugin that notifies the whole platform when a new study reaches the PACS server. While a single memory queue may be enough for a traditional intranet archive, a reliable queue system was required to sustain the system at the cloud level, with a multi-organization environment. As the plugin system is dynamic, the extensible Dicooogle architecture enables the building and provisioning of micro-services with high availability. These notifications will raise and trigger different actions on the platform, such as distributing studies across different places with pre-fetch mechanisms or distributing the studies according to rules in an established revision worklist, or it may also push the notifications to third-party information systems. This example demonstrates the flexibility of the Dicooogle architecture, which enables a more complex implementation of the service provided by default, by just adding or replacing a plugin.

The PACScenter platform, backed by Dicooogle, has been used to support regional screening programs and teleradiology workflows with centralized endpoints, but with high-availability clusters relying on containers and cluster toolsets. The solution is purely web-based and runs in any cloud provider. It has been used to receive medical imaging studies from hundreds of medical facilities and thousands of modalities from a huge variety of modality equipment, vendors, and countries. For example, one of the deployments now receives more than 70,000 studies per month with hundreds of users, demonstrating well Dicooogle’s robustness and performance. The medical images are concurrently received and indexed, while also providing the images to be visualized by radiologists in different regions through the Web. In another

⁷ Official website: <https://www.bmd-software.com/>

⁸ PACScenter information: www.bmd-software.com/solutions/pacscenter/

Fig. 2 Architecture of the routing mechanism of incoming objects to Dicoogle



global scale installation, the system is being used 24 h a day without interruptions and has already been accessed by more than 95 countries.

The flexibility of the plugin-based system also created opportunities during the COVID-19 pandemic, by supporting a screening program.⁹ In this program, the platform allowed easy incorporation of new AI tools that were developed to detect the COVID-19 virus. This service was used at a national level with X-ray images, allowing doctors to confirm the diagnosis quickly and efficiently.

Large-scale Dicoogle network

The storage and indexing technologies used by Dicoogle are fully customizable thanks to the storage and indexing

interfaces. This flexibility allows Dicoogle to work both locally, on a simple installation, and on the cloud in a multi-node cluster of machines. In the latter context, Dicoogle has been successfully deployed in large-scale scenarios, handling more than one million DICOM files each day.

BMD Software developed a DICOM router and queue mechanism for PACS environments that follow the standard, relying on the Dicoogle SDK. Figure 2 shows the solution architecture based on Apache Kafka¹⁰ technology. The DICOM Router is fully developed on top of Dicoogle SDK and takes advantage of other plugins already available such as any storage mechanisms. The DICOM objects may arrive from any location, sent by a DICOM node, for instance, a modality device that sends the acquired images to the

⁹ <https://healthmanagement.org/pdf/article-download/how-to-support-a-national-network-for-covid-19-identification-in-medical-imaging-studies/hm3-v20-5may-how-to-support-a-national-network.pdf>

¹⁰ Website: <https://kafka.apache.org/>

Dicoogle DICOM Storage (C-STORE SCP). A new indexer plugin has been developed, and it is called for each arriving DICOM object, with the received URIs as parameters. The plugin will introduce it in a reliable queue, provided by a Kafka cluster. The URI is inserted into the queue by a Kafka producer and will be handled by a Kafka consumer belonging to the same group, and later routed to the applicable DICOM node destination.

The architecture possesses one main topic (Kafka queue) where all the DICOM objects received by the indexer plugin are placed. A secondary topic for lower priority files was created to support priority queueing. This operation is asynchronous for increased performance. Each object in this queue will be matched against a set of filters that are applied over the contents of the DICOM file. This allows the plugin for example to filter objects by a specific patient or modality. If the condition of the specified filters is met, the selected DICOM object will be removed from this queue and inserted in a node-specific queue defined by the configured list of AETitle nodes. Each node has its own queue and is processed individually. A Storage Dispatcher will be selected to process the URI and send it via C-STORE SCU or STOW-RS. This operation is synchronous since the plugin must guarantee that the objects reach their final destination.

The final step is to validate the storage operation, and if the object reached the AETitle node. If these operations finish successfully, the DICOM object is removed from the node queue. If it fails, however, the object is re-inserted in the queue. This process takes place during the consumer lifecycle. This architecture allows the distribution of operations, leading to a lower latency between the moment when the study store was requested and the moment when the storage was completed. Moreover, the Kafka strategy allows the architecture to scale, as the Storage Dispatcher can easily be instantiated in different computing nodes, and sent to multiple nodes, at exactly the same time. Finally, resilience to failure during both the indexing procedure, where the DICOM objects are assigned their destinations and during the storage procedure, where the objects are sent, is inherently guaranteed by Kafka since, in the occurrence of a system failure, the queue entries will not be lost and will be reprocessed when the system is back online.

This system was successfully integrated into a nationwide installation of PACScenter. With more than 90 medical facilities connected to one single central server, the infrastructure could not handle the demand for service. This routing mechanism was implemented to split the network into smaller chunks, comprising many Dicoogle nodes that act as forwarding agents. The store and forward plugin were able to successfully reduce the workload on the central server and handle the high demand for service.

Table 1 Analysis of the most updated open-source PACS in August 2022

	Last Update	Source code Availability	Supported Platforms	Programming Language	User Guide	Web Viewer	Plug-in Extensible	Development Framework	Documentation	REST API	DICOMWeb Support	DICOM Pathology Support
MRIdb	10/08/2018	GitHub	Linux	Java		X			X		X	
DCMTK (Offis)	28/04/2022	GitHub	Windows Linux Mac	C/C++					X	X		
Orthanc	16/08/2022	Self hosted	Windows Linux Mac	C++	X	X	X		X	X		X
dcm4chee	26/07/2022	GitHub	Windows Linux Mac	Java	X	X	X		X	X	X	
Dicoogle	09/08/2022	GitHub	Windows Linux Mac	Java	X	X	X	X	X	X	X	X

Collaborative and research projects

Several collaborative and research projects have been backed by the Dicoogle ecosystem. The framework has been a key-stone in diversified projects because of the easy-to-integrate functionalities. In this subsection, we review and analyze those projects, developed with Dicoogle or its sub-products, in recent years. Appendix 1, Table 1 lists and gives a brief description of some of the main projects found in the literature.

The various research topics which have relied on Dicoogle present a broad spectrum of domains. For instance, the authors of [42] proposed a system capable of identifying automatically relevant information to help diagnosis. In this work, Dicoogle was used to extract DICOM data for subsequent data mining. It was also used to index visual features extracted from the pixel data. In the same year, Silva et al. [48] explored the capabilities of Dicoogle to detect and process inconsistencies in radiology departments. The authors focused on the radiation dose, identifying abnormal values, and improving healthcare service quality and safety.

In [49], the authors developed multiple Dicoogle plugins for the same purpose of indexing and querying DICOM data. The idea was to evaluate the usage of distinct database models in the medical imaging context, including relational and document-based databases, comparing the performance in production and research environments. They concluded that the choice of database technology depends on the real-world use case and that Dicoogle was the ideal tool for the assessment since it supported several plugins (i.e. database types) working at the same time in the same solution instance.

Since Dicoogle supports multiple data sources, including DICOM metadata, visual features, annotations, and reports, Pinho et al. [10] decided to design and implement an extension to Dicoogle for a multimodal search engine. This work encompassed an architecture enabling complex heterogeneous queries to the archive, both text-based and image-based via query-by-example while providing a web-based user interface for their construction. Silva et al. [8] presented a reversible de-identification mechanism where the DICOM objects indexed by Dicoogle are fully de-identified. The work developed by the authors can retain the querying capabilities as the original DICOM objects were indexed without anonymization.

In recent years, Digital Pathology has been gaining prominence as a new branch in medical imaging [50]. It replaces the traditional pathology methods with innovative scanned images and viewers. Godinho et al. [41] used Dicoogle capabilities to store, index, and distribute via DICOM Web whole-slide images in DICOM format. The solution is based on a tiling mechanism backed by Dicoogle plugins.

In [40], the authors developed a community-driven validation service for DICOM standard objects, supported by a web information system. This de-identifies uploaded medical images on the web on the client-side, assuring the

privacy of the patient images in the uploaded study, and are then stored in a Dicoogle archive instance. This work claims that using a simple UI, without requiring extra configurations, it is possible to address problems such as non-conformity and inconsistencies created by medical staff or non-compliant DICOM products.

SCREEN-DR [34] is a collaborative platform whose main goal is to collaboratively annotate medical images of patients that are suspected of having diabetic retinopathy (DR). In the second phase of the work, the authors look forward to training machine learning models to aid the screening and diagnosis process.

In [51], the authors present an architecture to automatically extract extra information from DICOM images based on automatic labelling through classification algorithms. The authors implemented these methods in Dicoogle and added support for the users of this PACS archive to perform content discovery via multimodal querying. The implemented methods allow, for instance, the search for DICOM objects with a particular body part, even without the metadata of that DICOM object ever referring to that body part.

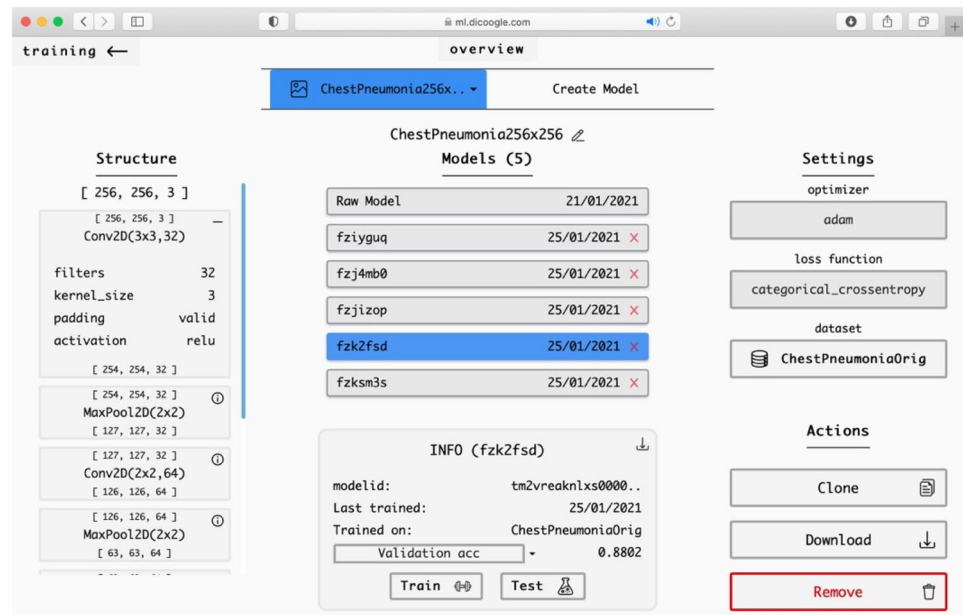
Exploration and analysis of medical data indicators from medical imaging repositories are becoming a normal procedure in the business intelligence of healthcare institutions. The analysis of big data gathered from DICOM metadata may help to improve the quality of the services, identify anomalies, and optimize workflows. In [52], the authors describe a business intelligence framework that can be integrated with the data indexed by Dicoogle, allowing the visualization of performance indicators on a web dashboard.

Following the work developed in [41], the authors of [35] extended the concept by integrating collaborative functionalities to support real-time telepathology. The improved web-based, whole-slide imaging viewer enables the creation of working sessions where the actions performed are synchronized and visible in real-time to, and only to, the session users. Additional capabilities such as live chat and session actions replay and/or catch-up are available. Similar to [41], this work relies greatly on the WADO-RS Dicoogle plugin and Dicoogle's web service plugin API. The solution also contemplates a multi-organizational archive managed by a role-based access control mechanism.

Almeida et al. [53] designed and implemented a database architecture to support a distributed index over the network, based on new indexing, querying, and storage plugins for Dicoogle. The approach relies on document-based databases like MongoDB to split the data across multiple nodes. The deployed architecture also makes use of replication capabilities. This work proved the advantages of using a flexible architecture when dealing with amounts of data that will require horizontal scaling and high availability.

Baptista et al. [54] proposed a scalable PACS architecture based on the cloud technology Kubernetes using Dicoogle.

Fig. 3 Screenshot of the WebML web application



The goal of the system was to address the challenge of serving nationwide PACS installations. The high traffic demand requires an intelligent architecture capable of dynamically allocating resources to meet service demand. This work took advantage of Dicoogle's extensibility to develop a metrics plugin to expose custom metrics such as the current number of DICOM associations to be used by Kubernetes to automatically scale the network. The work proved that Dicoogle is ready to be integrated into a multi-node environment with significant performance gains.

In [55], the authors designed an architecture for Dicoogle which implements a role-based access control mechanism. The authors implemented a framework capable of restricting and controlling the DICOM Web services, such as QIDO-RS, WADO-RS, and STOW-RS. In this work, Lebre et al. argue that this framework is a key cornerstone for future work to deal with and allow the support of medical archives in the cloud, thereby potentiating the establishment of more efficient teleradiology services.

The WebML (Fig. 3) is an ongoing project whose goal is to search and consume data indexed by Dicoogle for deep learning processes. Automatic image analysis is growing rapidly in the medical field due to the workflow improvements it brings. However, designing these algorithms requires specific know-how that is not present in medical institutions. Exploiting the advancements in automatic machine learning, the platform provides a set of tools to easily develop neural networks for a variety of use cases, including teaching and research. This platform enables the rapid development and prototyping of image analysis solutions without requiring the knowledge to develop the algorithms themselves. The models produced on this platform

will be made available in the Dicoogle CBIR module, supporting the query-by-example functionality. The first phase of the project is already available online at the ml.dicoogle.com portal as shown in Fig. 3. WebML allows a set of complex operations with just a few mouse clicks, including the association of new computational nodes with GPUs, the management of datasets and the design and training of models. In the future, this architecture will allow the usage of data and metadata stored in instances of the Dicoogle PACS archive to automate findings and diagnosis.

Learning medical imaging informatics

Once a year, in every first academic semester, the Electronics, Telecommunications, and Informatics department at the University of Aveiro offers the optional subject of *Networks and Services in Imaging*. This subject is attended every year by an average of 30 students with a background in two main fields: radiology and computer science. Prior to their enrolment, Medical Imaging Technology students have limited knowledge of computer science, while computer engineering students have no prior knowledge of DICOM and PACS concepts. The teaching of systems and networks in medical imaging has, then, to balance the knowledge of students with different backgrounds:

- On one hand, the students of medical imaging understand the fundamentals of PACS archives from the user's point of view in research and clinical practice; additionally, students with this background usually have an understanding of techniques to work with these systems; however, the majority of these students do not have prior

knowledge of developing new features for open-source PACS;

- On the other hand, the students of computer engineering and software engineering need to be given the necessary background regarding what PACS and DICOM are and what they are used for, seeking the best perception of the real world in order to develop new solutions;

As such, this subject starts by introducing the theoretical fundamentals of medical imaging laboratories, DICOM and PACS. It addresses multiple medical imaging modalities, different IODs, and the most commonly used transfer syntax. Furthermore, students are introduced to quality management and common issues and challenges in PACS and RIS.

At the end of the subject, students are evaluated with a final practical project, in addition to theoretical evaluation. The objective of practical assessment is to consolidate the concepts acquired during the introductory part of the subject. Over the years, students have developed solutions based on Dicoogle and plugin development backed by the instructions found in the Dicoogle Learning Pack. The studies [52, 53, 56] are examples of *Networks and Services in Imaging* practical projects that culminated in scientific contributions, given the interest in the innovation of PACS archives.

The acceptance metrics were then evaluated anonymously. The results are described in "Results".

Results

Community positioning

The positioning of Dicoogle in the medical imaging software field can be done through comparison with other well-known open-source PACS archives. The survey to discover available software to compare was based on past review papers [57, 58] and through the query "open-source PACS" in the Google Scholar search engine. The list of the 12 solutions gathered included Dicoogle [36, 37], Orthanc [24, 59], MRIdb [23], DCMTK (Offis) [21], dcm4chee [22], Xebra, OSPACS, CDMedic PACS Web, JSVdicom server, Dicom-Server@Medical Connections, ClearCanvas, K-PACS and ConquestDICOM.

Analysis of the solutions started by verifying the availability of the source code of each solution. To this end, we checked the date of the last update of the solution—by 1st August 2022. We then discarded the solutions that had no available public source code or did not receive an update in the last four years. The remaining open-source PACS archive solutions were DCMTK (Offis), Orthanc, dcm4chee, MRIdb, and Dicoogle.

The last step was to analyze the solutions based on parameters previously set for this evaluation: supported operating

systems, user guide, the existence of an integrated web viewer, expandable architecture, development framework or SDK available, technical documentation available, REST API, and DICOMWeb support. The results were serialized and are presented in Table 2.

Going over each entry in detail, dcm4chee is a PACS application with a rich web interface for management of DICOM data but also patient data and administrative data. This project exposes a rich toolkit, written in java, to handle the DICOM files and this toolkit is used by both MRIdb and Dicoogle. MRIdb is a PACS application, built on top of the dcm4chee toolkit, built specifically for the handling of MRI images. It integrates a management interface for the DICOM files and administrative data and a web viewer for visualization of the MRI data. This product, however, cannot support other imaging modalities, which is a major limitation. Dcmtk is a collection of libraries and toolkits for the handling of DICOM files written in C/C++. It competes with the dcm4chee toolkit. This software, however, is not meant for standalone use, but rather to be integrated in other projects or used as a tool to communicate with other PACS systems. Orthanc uses the DCMTK toolkit to manage the DICOM files.

Dicoogle and Orthanc have stood out in terms of satisfying criteria, establishing themselves as the strongest deployed systems both in academic and clinical production environments. These solutions share many similarities. They can both be extended by plugins and are the only solutions to support digital pathology. Dicoogle distinguishes itself from Orthanc by providing separate interfaces for indexing and storage of the DICOM files. This gives Dicoogle greater flexibility, as demonstrated by the use cases presented in Sects. "TeleHealthcare@Cloud" and "Large-scale Dicoogle network". By having two separate interfaces it becomes possible to swap storage solutions without touching the indexing solution and vice-versa. Meanwhile, Orthanc opted for a single interface that allows developers to swap the database technology used by the application. While more restrictive, it still allows developers to have some control over their storage/indexing solution. Another distinctive feature of Dicoogle is the possibility to extend the core web user interface with plugins. Orthanc also possesses capabilities to integrate web plugins but these are built as separate web applications while Dicoogle web plugins are integrated seamlessly in the core interface. Out of the box, Orthanc possesses more functionality than Dicoogle, as users can download prepared packages with all the official plugins that include a more advanced web viewer, REST APIs for CRUD operations over DICOM files, and ImageJ integration for image processing and analysis pipelines. Dicoogle opted instead to distribute a minimalist PACS, with only the necessary tools to manage the DICOM files.

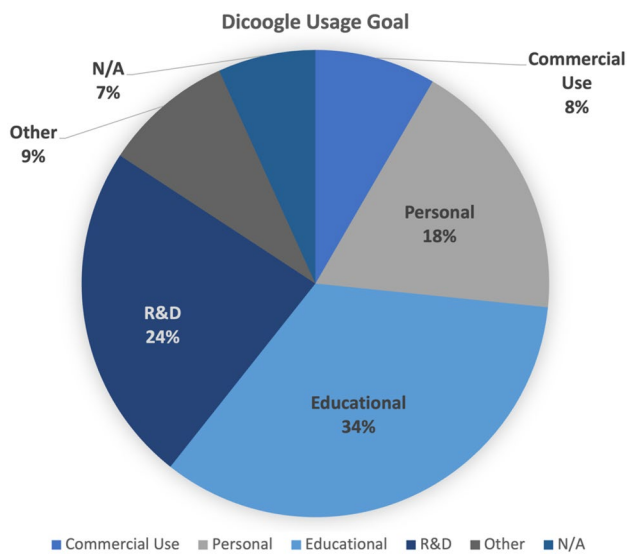


Fig. 4 Reported interests by 9041 users that have downloaded Dicoogle since 2018 (data acquired from the download form on the Dicoogle website)

Usage evaluation

Dicoogle started to attain worldwide visibility early in 2014, the time of its first official release together with the launch of the official website. Since then, the different release versions have accumulated more than 20,000 downloads. However, the solution can also be downloaded directly from GitHub and the absolute number of downloads in this platform is not available. As shown in Fig. 4, the interests of users when downloading Dicoogle is diversified. Users reported that their goal when downloading Dicoogle was mainly “Educational” (34%), followed by users who claimed “R&D” purposes (24%). Dicoogle was also used for “Commercial Use” by 8% of users and for personal reasons by 18%. Finally, 9% of users claimed “Other” reasons and the remaining 7% of users used their right not to answer. However, it is our perception that many commercial usages

are not being divulged and the solution may have been rebranded, violating the license terms.

Dicoogle is downloaded and used worldwide in distinct contexts, with the largest number of users being from the United States, Europe, Brazil, and India.

The Dicoogle website was re-launched at the end of 2017 after a breaking update that changed the interface completely and introduced the Learning Pack to the community. The impact of the new portal increased the number of visits per month in the following years (see Fig. 5).

Over the years, the number of downloads per year has also been increasing. The increase of such indicators makes us believe that the use and adoption of Dicoogle are increasing in the medical imaging community. A temporary drop in the number of downloads was recorded in the beginning of 2020, resulting from the pandemic context [60] and the period of adapting to teleworking, returning to normal in the second half the of year and staying constant since. These results indicate that the community is still interested in Dicoogle.

Acceptance in academia

At the end of the academic year, students of Networks and Services in Imaging at the Electronics, Telecommunications, and Informatics department of the University of Aveiro were asked to answer a set of questions evaluating the acceptance of Dicoogle, Dicoogle Learning Pack, and API definition as learning resources in academia. This study started in 2018 and finished at the end of the first semester of the 2020/2021 academic year, meaning that for three consecutive years, approximately 90 students from three different masters and one doctoral program voluntarily answered.

Figure 6 shows the student distribution in Networks and Services in Imaging during the academic years of 2018/2019, 2019/2020, and 2020/2021. The graph shows that more than half the students attending these classes had previous contact with programming languages. However, 43% of the students (master students in Medical Imaging Technologies) do not

Fig. 5 Downloads per month from the beginning of 2018 until December 2021

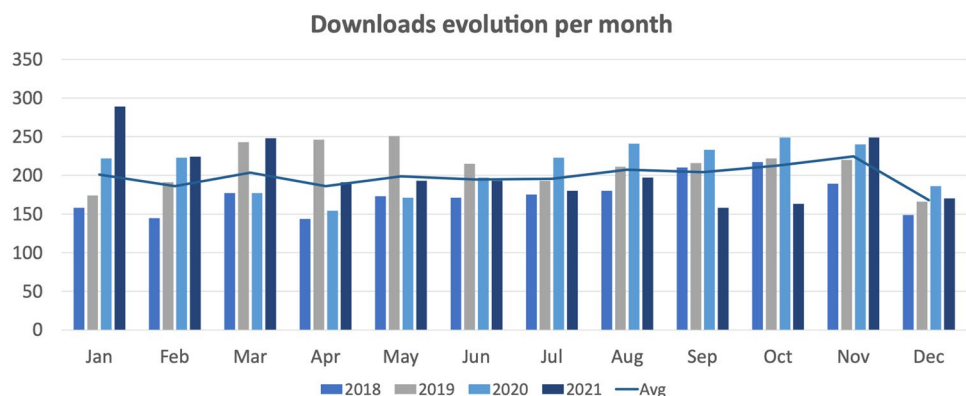
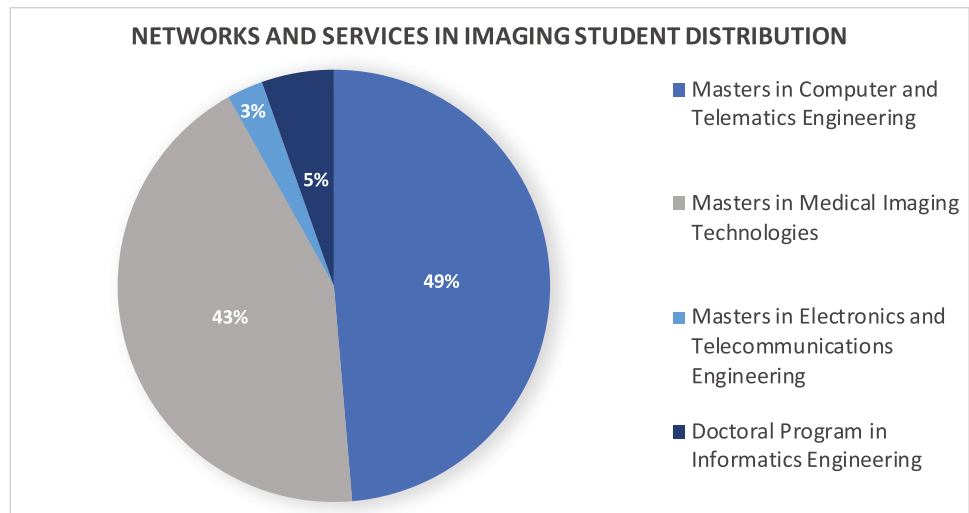


Fig. 6 Distribution of students per course from the academic year 2018/2019 until 2020/2021



have any previous curricular computer science background. This poses challenges as it is mandatory to address both medical imaging and computer science students.

The solution found was to propose to the student's collaboration between members of the two distinct fields. The Dicoogle ecosystem was introduced in the classes following the Dicoogle Learning Pack and the students were asked to develop small projects based on the platform.

The usage of Dicoogle was not, however, mandatory. During the reported years, 63% of students chose to develop the final project using Dicoogle. Of the students who developed solutions using Dicoogle, 100% of them used the available learning pack. Additionally, 95% of the students who used the Dicoogle Learning Pack in their final project found the resources useful.

Conclusion

This paper gives an overview of the Dicoogle project as a tool designed to tackle multiple fields of action in industry, research, and academia. It also gives an in-depth analysis of the Dicoogle ecosystem, focusing on the learning resources

and associated projects that extend its features. Dicoogle is an open-source PACS archive, fully DICOM compliant, the processes of which enable researchers to quickly set up and use the platform.

With Dicoogle, complex use cases in distinct application areas may be satisfied by different technologies that can be appended to the platform in the form of plugins, as shown in the representative cases. The web-based interface allows navigation over the extracted metadata of the DICOM objects, and its extensible characteristics facilitate the introduction of new web components.

Dicoogle is currently a fundamental enabling technology in collaborative and tele-healthcare environments, including national and European research projects, regional screening programs, and teleradiology services based on Cloud infrastructure. The paper also evaluates the positioning of Dicoogle in the medical imaging software community through a comparative analysis against well-known open-source PACS archives. It discusses the evidence of the growing popularity of Dicoogle in recent years, analysing the data of platform usage and applications in different areas. The results show the worldwide acceptance of Dicoogle as a relevant tool with a big impact on industry, research, and teaching medical imaging informatics.

Appendix 1

Table 2 Examples of research studies conducted in the past five years

Project	Year	Field	Description	Ref.
A Cloud-Ready Architecture for Shared Medical Imaging Repository	2020	Security Plugin	In this work, authors designed an architecture to enable Dicoogle to restrict accesses for multiple users. This feature allowed the protection of DICOM Web services as QIDO-RS, WADO-RS, and STOW-RS. Additionally, the architecture allows the support of archives in the cloud	[55]
NoSQL distributed database for DICOM objects	2020	Databa Indexing, Plugin	Almeida et al. built plugins to support the index and query over data stored in MongoDB. Furthermore, the authors deployed the system in a distributed architecture to test the distribution performance of the DICOM objects converted into JSON. This work powers the horizontal scaling when the resources are reaching the limit	[53]
Collaborative Framework for a Whole-Slide Image Viewer	2019	Informa systems, Plugin	Lebre et al. created a collaborative viewer using Dicoogle ecosystem and the architecture developed by Godinho et al. [40]. The deployed system allows the creation of working sessions, synchronization of operations, and replaying of the actions performed in a session, among other features. The whole system is backed by Dicoogle and its DICOM Web plugins	[35]
ETL Framework for Real-Time Business Intelligence over Medical Imaging Repositories	2019	Informa Systems	The authors propose and Extract, Transform and Load framework for a business intelligence application. The data that feeds the framework is backed by the Dicoogle archive ecosystem which provides real-time data about the stored objects. The final solution features an extensible web-based dashboard providing tools for data mining, cleansing, filtering, and clustering functions	[52]
Automated Anatomic Labeling Architecture for Content Discovery in Medical Imaging Repositories	2018	CBIR	In this work, Pinho et al. proposed an anatomic labeling architecture integrated with Dicoogle. The solution includes a technical specification for classifiers, a classification database, and proof-of-concept convolutional neural network classifiers to identify the presence of organs in medical imaging studies. The system also extracts regions of interest, saving them in a database for further usage in multi-modal querying mechanisms	[51]
SCREEN-DR: Collaborative platform for diabetic retinopathy	2018	Informa systems, CAD	The authors proposed a multidisciplinary collaborative platform for textual and visual annotation of image datasets. The platform allows the visualization of diabetic retinopathy images. Furthermore, the authors developed the system in close collaboration with physicians and researchers, aiming to create datasets and machine learning algorithms for automated diagnosis	[34]
An efficient architecture to support digital pathology in standard medical imaging repositories	2017	Information systems, Plugin	Godinho et al. developed a solution to support digital pathology and whole-slide imaging in Dicoogle. The authors implemented a viewer in Dicoogle which uses the DICOM Web plugins for Dicoogle to access the DICOM objects and instance frames required to display the whole image in the viewer. The system uses pure web technologies, allowing usage in all devices with an internet connection and web browser	[41]

Table 2 (continued)

Project	Year	Field	Description	Ref.
Controlled searching in reversibly de-identified medical imaging archives	2017	CBIR, Pseudo sation	In this work, the authors integrated with Dicoogle a system which fully de-identified DICOM objects, including metadata and pixel data. At the same time, the solutions assure the search capabilities from the original data. This work allows the sharing of anonymized studies with the community, while still searchable for authorized personnel	[8]
A multimodal search engine for medical imaging studies	2016	CBIR, Storage	Pinho et al. describe how the authors developed an extensible platform for multi-modal medical imaging querying and retrieval, integrated into Dicoogle. This enabled Dicoogle with profile-based CBIR capabilities. The outcome comprises a web platform integrated into the Dicoogle Web UI for users to apply multi-modal query techniques	[10]
Assessing the relational database model for optimization of content discovery services in medical imaging repositories	2016	Databa Plugin	This work describes the performance of multiple DBMS when indexing and querying DICOM metadata. The authors developed several plugins for Dicoogle based on PostgreSQL and MongoDB, comparing them and presenting performance numbers	[49]
Normalizing Heterogeneous Medical Imaging Data to Measure the Impact of Radiation Dose	2015	Data mining, QA	Santos et al. explored the Dicoogle functionalities to identify data and process inconsistencies to improve radiology departments' quality and safety. The authors address problems like dose surveillance and image quality control. Additionally, the authors describe a method to examine and study medical imaging repositories	[48]
A Recommender System for Medical Imaging Diagnostic	2015	CBIR, Data mining	In this study, the authors used Dicoogle to build a context-based recommender system for diagnostic using medical images. The developed system uses data mining and context-based retrieval mechanisms to automatically identify information relevant to physicians during the diagnostics. Furthermore, the authors reported the use of Dicoogle to extract and index image-based features	[42]

Acknowledgements This work received support from FCT, Fundação para a Ciência e a Tecnologia, under the project UID/CEC/00127/2019 and POCI-01-0145-FEDER-016385 (NETDIAMOND). This work was also funded by IMAGE-IN (EU H2020 MSCAITN No 813920).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. R. M. Rad, P. Saeedi, J. Au, J. Havelock, A hybrid approach for multiple blastomeres identification in early human embryo images, *Computers in biology and medicine* 101 (2018) 100–111.
2. D. M. Dhane, M. Maity, T. Mungle, C. Bar, A. Achar, M. Kolekar, C. Chakraborty, Fuzzy spectral clustering for automated delineation of chronic wound region using digital images, *Computers in biology and medicine* 89 (2017) 551–560.
3. K. J. Dreyer, J. H. Thrall, D. S. Hirschorn, A. Mehta (Eds.), *PACS, 865* Springer-Verlag, New York, 2006. <https://doi.org/10.1007/0-387-31070-3>.
4. H.K. Huang, *PACS and imaging informatics: basic principles and applications* (John Wiley & Sons, 2011)
5. M. Ruiz, A. Julia, I. Boada, Starviewer and its comparison with other open-source DICOM viewers using a novel hierarchical evaluation framework, *International journal of medical informatics* 137 (2020) 104098.
6. O.S. Panykh, *Digital Imaging and Communications in Medicine (DICOM) - A Practical Introduction and Survival Guide* (Springer, 2012). <https://doi.org/10.2967/jnumed.109.064592>
7. P. Mildenerger, M. Eichelberg, E. Martin, Introduction to the DICOM standard, *European Radiology* 12 (4) (2002) 920–927. <https://doi.org/10.1007/s003300101100>.
8. J. M. Silva, E. Pinho, E. Monteiro, J. F. Silva, C. Costa, Controlled searching in reversibly de-identified medical imaging archives, *Journal of Biomedical Informatics* 77 (2018) 81 – 90

9. P. H. Oliveira, L. C. Scabora, M. T. Cazzolato, W. D. Oliveira, R. S. Paixao, A. J. Traina, C. Traina, Employing domain indexes to efficiently query medical data from multiple repositories, *IEEE journal of biomedical and health informatics* 23 (6) (2018) 2220–2229.
10. E. Pinho, T. Godinho, F. Valente, C. Costa, A multimodal search engine for medical imaging studies, *Journal of Digital Imaging* 30 (1) (2017)39–48. <https://doi.org/10.1007/s10278-016-9903-z>.
11. E. Pinho, C. Costa, Extensible architecture for multimodal information retrieval in medical imaging archives, in 2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), 2016,890 pp. 316–322. <https://doi.org/10.1109/SITIS.2016.58>.
12. K. Fan, S. Wang, Y. Ren, H. Li, Y. Yang, Medblock: Efficient and secure medical data sharing via blockchain, *Journal of medical systems* 42 (8) (2018) 136.
13. M. Vikram, A. Anantharaman, S. BS, An approach for multimodal medical image retrieval using latent dirichlet allocation, in: *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 2019, pp. 44–51.
14. R. Sigal, PACS as an e-academic tool, *International Congress Series* 1281(2005) 900–904.900
15. J. M. Silva, A. Guerra, J. F. Silva, E. Pinho, C. Costa, Face deidentification service for neuroimaging volumes, in: 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), IEEE, 2018, pp. 141–145.
16. S. Sinha, U. Sinha, H. Kangarloo, H. K. Huang, A PACS-based interactive teaching module for radiologic sciences. *AJR. American journal of roentgenology* 159 (1) (1992) 199–205.
17. C. C. Tchoyoson Lim, G. L. Yang, W. L. Nowinski, F. Hui, Medical Image Resource Center—making electronic teaching files from PACS, *Journal of Digital Imaging* 16 (4) (2003) 331–336.
18. C. Lim, G. Yang, Electronic teaching files and continuing professional development in radiology., *Biomedical imaging and intervention journal*910 2 (2) (2006) e5. <https://doi.org/10.2349/bij.2.2.e5>.
19. L. E. Wilkinson, S. R. Gledhill, An Integrated Approach to a Teaching File Linked to PACS, *Journal of Digital Imaging* 20 (4) (2007) 402–410. <https://doi.org/10.1007/s10278-006-1045-2>.
20. S. J. Doran, J. D'Arcy, D. J. Collins, R. Andriantsimavona, M. Orton, D. M. Koh, M. O. Leach, Informatics in Radiology: Development of a Research PACS for Analysis of Functional Imaging Data in Clinical Research and Clinical Trials, *Radio Graphics* 32 (7) (2012) 2135–2150
21. M. Eichelberg, J. Riesmeier, T. Wilkens, A. J. Hewett, A. Barth, P. Jensch, 20 Ten years of medical imaging standardization and prototypical implementation: the DICOM standard and the offis DICOM toolkit (DCMTK), in: *Medical Imaging 2004: PACS and Imaging Informatics*, Vol. 5371, International Society for Optics and Photonics, 2004, pp. 57–69.
22. M. J. Warnock, C. Toland, D. Evans, B. Wallace, P. Nagy, Benefits of us925 ing the dcm4che dicom archive, *Journal of Digital Imaging* 20 (1) (2007)125–129.
23. M. Woodbridge, G. Fagiolo, D. P. O'Regan, MRIdb: medical image management for biobank research, *Journal of digital imaging* 26 (5) (2013)886–890.930
24. S. Jodogne, The Orthanc ecosystem for medical imaging, *Journal of digital imaging* 31 (3) (2018) 341–352.
25. E. J. Gomez, F. del Pozo, E. J. Ortiz, N. Malpica, H. Rahms, A broadband multimedia collaborative system for advanced telerradiology and medical imaging diagnosis, *IEEE Transactions on information technology in935 biomedicine* 2 (3) (1998) 146–155.
26. Y. S. Lim, D. D. Feng, T. W. Cai, A web-based collaborative system for medical image analysis and diagnosis., in: *VIP*, 2000, pp. 93–95.
27. O. Sumtsova, T. Aikina, L. Bolsunovskaya, C. Phillips, O. Zubkova, P. Mitchell, Collaborative learning at engineering universities: Benefits and challenges, *International Journal of Emerging Technologies in Learning (IJET)* 13 (1) (2018) 160–177.
28. M. C. Makel, K. N. Smith, M. T. McBee, S. J. Peters, E. M. Miller, A path to greater credibility: Large-scale collaborative education research, *AERA Open* 5 (4) (2019)
29. C. Daniel, F. Macary, M. G. Rojo, J. Klossa, A. Laurinavicius, B. A. Beckwith, V. Della Mea, Recent advances in standards for collaborative digital anatomic pathology, in: *Diagnostic Pathology*, Vol. 6, BioMed Central, 2011, p. S17.
30. K. Zhang, T. Ling, Y. Yang, J. Sun, M. Wang, J. Zhang, Clinical experiences of collaborative imaging diagnosis in shanghai district healthcare services, in: *Medical Imaging 2016: PACS and Imaging Informatics: Next Generation and Innovations*, Vol. 9789, International Society for Optics and Photonics, 2016, p. 97890X.
31. R. Maree, L. Rollus, B. Stevens, R. Hoyoux, G. Louppe, R. Vandaele, 955 J.M. Begon, P. Kainz, P. Geurts, L. Wehenkel, Collaborative analysis of multi-gigapixel imaging data using cytomine, *Bioinformatics* 32 (9) (2016) 1395–1401.
32. I. Maglogiannis, C. Andrikos, G. Rassias, P. Tsanakas, A DICOM based collaborative platform for real-time medical teleconsultation on medical960 images, in: *GeNeDis 2016*, Springer, 2017, pp. 79–91.
33. D. Díaz, G. Corredor, E. Romero, A. Cruz-Roa, A web-based telepathology framework for collaborative work of pathologists to support teaching and research in latin america, in: *Sipaim–Miccai Biomedical Workshop*, Springer, 2018, pp. 105–112.965
34. M. Pedrosa, J. M. Silva, J. F. Silva, S. Matos, C. Costa, Screenshot: Collaborative platform for diabetic retinopathy, *International journal of medical informatics* 120 (2018) 137–146.
35. R. Lebre, R. Jesus, P. Nunes, C. Costa, Collaborative framework for a whole-slide image viewer, in: 2019 IEEE 32nd International Symposium970 on Computer-Based Medical Systems (CBMS), IEEE, 2019, pp. 221–224.
36. F. Valente, L. A. B. Silva, T. M. Godinho, C. Costa, Anatomy of an extensible open source PACS, *Journal of Digital Imaging* 29 (3) (2016) 284–296. <https://doi.org/10.1007/s10278-015-9834-0.975>
37. F. Valente, C. Costa, A. Silva, Dicoogle, a PACS featuring profiled content-based image retrieval, *PloS one* 8 (5) (2013) e61888.
38. C. Costa, F. Freitas, M. Pereira, A. Silva, J. L. Oliveira, Indexing and retrieving DICOM data in disperse and unstructured archives, *International Journal of Computer Assisted Radiology and Surgery* 4 (1) (2009) 71–77.980. <https://doi.org/10.1007/s11548-008-0269-7>.
39. T. M. Godinho, C. Viana-Ferreira, L. A. Bastião Silva, C. Costa, A routing mechanism for cloud outsourcing of medical imaging repositories, *IEEE Journal of Biomedical and Health Informatics* 20 (1) (2016) 367–375. <https://doi.org/10.1109/JBHI.2014.2361633.985>
40. J. M. Silva, T. M. Godinho, D. Silva, C. Costa, A community-driven validation service for standard medical imaging objects, *Computer Standards& Interfaces* 61 (2019) 121–128.
41. T. M. Godinho, R. Lebre, L. B. Silva, C. Costa, An efficient architecture to support digital pathology in standard medical imaging repositories, *Journal of Biomedical Informatics* 71 (2017) 190–197.
42. E. J. M. Monteiro, F. Valente, C. Costa, J. L. Oliveira, A recommender system for medical imaging diagnostic., in: *MIE*, 2015, pp. 461–463.
43. M. Santos, L. Bastiao, A. Silva, N. Rocha, DICOM metadata analysis for population characterization: A feasibility study, *Procedia Computer Science* 100 (2016) 355 – 361.
44. M. Santos, L. Bastiao, C. Costa, A. Silva, N. Rocha, DICOM and clinical data mining in a small hospital PACS: A pilot study, in: *M. M. Cruz Cunha, J. Varajao, P. Powell, R. Martinho (Eds.), ENTERprise Information Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp.1000 254–263.

45. J. Philbin, F. Prior, P. Nagy, Will the next generation of PACS be sitting on a cloud?, *Journal of digital imaging* 24 (2) (2011) 179–183.
46. T. Rostrom, C.-C. Teng, Secure communications for PACS in a cloud environment, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2011, pp. 8219–8222.
47. L. S. Ribeiro, C. Viana-Ferreira, J. L. Oliveira, C. Costa, XDS-I out-sourcing proxy: ensuring confidentiality while preserving interoperability, *IEEE journal of biomedical and health informatics* 18 (4) (2013) 1404–1412
48. L. A. B. Silva, L. S. Ribeiro, M. Santos, N. Neves, D. Francisco, C. Costa, J. L. Oliveira, Normalizing heterogeneous medical imaging data to measure the impact of radiation dose, *Journal of digital imaging* 28 (6) (2015) 671–683.
49. T. M. Godinho, C. Costa, et al., Assessing the relational database model for optimization of content discovery services in medical imaging repositories, in: 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), IEEE, 2016, pp. 1–6.
50. S. Al-Janabi, A. Huisman, P.J. Van Diest. Digital pathology: Current status and future perspectives (2012). <https://doi.org/10.1111/j.1365-2559.2011.03814.x>
51. E. Pinho, C. Costa, Automated anatomic labeling architecture for content discovery in medical imaging repositories, *Journal of medical systems* (8) (2018) 145.
52. T. M. Godinho, R. Lebre, J. R. Almeida, C. Costa, ETL framework for real time business intelligence over medical imaging repositories, *Journal of digital imaging* (2019) 1–10.
53. A. Almeida, F. Oliveira, R. Lebre, C. Costa, NoSQL distributed database for DICOM objects, in: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2020, pp. 1882–1885.1030
54. T. Baptista, L. B. Silva, C. Costa, Highly scalable medical imaging repository based on Kubernetes, in: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2021, pp. 3193–3200.
55. R. Lebre, L. B. Silva, C. Costa, A cloud-ready architecture for shared medical imaging repository, *Journal of Digital Imaging* 33 (6) (2020) 1035 1487–1498.
56. J. R. Almeida, T. M. Godinho, L. B. Silva, C. Costa, J. L. Oliveira, Services orchestration and workflow management in distributed medical imaging environments, in: 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), IEEE, 2018, pp. 170–175.1040
57. K. Bjorn, Evaluation of open source medical imaging software: A case study on health technology student learning experience, *Procedia Computer Science* 121 (2017) 724–731.
58. S. Wideł, A. Wideł, D. Spinczyk, Overview of available open source PACS frameworks, *Studia Informatica* 37 (2016) 21–30.
59. S. Jodogne, C. Bernard, M. Devillers, E. Lenaerts, P. Coucke, Orthanc - A lightweight, restful DICOM server for healthcare and medical research, in: 2013 IEEE 10th International Symposium on Biomedical Imaging, 2013, pp. 190–193. <https://doi.org/10.1109/ISBI.2013.6556444>.
60. M. Pachetti, B. Marini, F. Giudici, F. Benedetti, S. Angeletti, M. Ciccozzi, C. Masciovecchio, R. Ippodrino, D. Zella, Impact of lockdown on covid19 case fatality rate and viral mutations spread in 7 countries in europe and north america, *Journal of Translational Medicine* 18 (1) (2020) 1–7

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.