PARADIM: a platform to support research at the interface of AI and medical imaging

Yannick Lemaréchal[2], Gabriel Couture[2], François Pelletier[1], Pierre-Luc Asselin[1], Ronan Lefol[1], Samuel Ouellet[1], Leonardo Di Schiavi Trotta[1], and Philippe Després[1,2]

[1]Département de physique, de génie physique et d'optique, Université Laval, Québec (Québec), Canada.
[2]Centre de recherche de l'Institut universitaire de cardiologie et de pneumologie de Québec-Université Laval, Québec, Québec, Canada.

Abstract This paper describes a digital infrastructure designed to support AI research in medical imaging, with a focus on Research Data Management best practices. The platform (PARADIM, Platform for the annotation, reuse and analysis of medical images - *Plateforme d'Annotation, de Réutilisation d'Analyse D'Images Médicales* in French) is rooted in the FAIR principles, through strict compliance with the DICOM standard, and aims at fulfilling several needs arising from the research community at the intersection of AI and medical imaging: robust data curation procedures, responsible data access and effective data governance, robust de-identification and annotation pipelines, and streamlined data operations (e.g. model training, analysis). The platform, built from open-source components, is designed to facilitate and automate the execution of large-scale, computationally-intensive pipelines (e.g. automatic segmentations, dose calculations, AI model training). PARADIM fills a gap at the interface of AI and medical imaging, where data and digital infrastructures were historically not given the attention and resources they deserve.

1 Introduction

Considerable resources have been allocated to advancing AI in recent years. Meanwhile, investments have been relatively scarce in elements fueling AI, namely data and the fabric that supports it, digital infrastructures. Research and development efforts towards data and infrastructure appears essential for AI to fulfill its promises. This paper describes a digital infrastructure designed to support AI research in medical imaging, with a focus on Research Data Management (RDM) best practices. The platform (PARADIM, Platform for the annotation, reuse and analysis of medical images - *Plateforme d'Annotation, de Réutilisation et d'Analyse D'Images Médicales* in French) is rooted in the FAIR principles [1], through strict compliance with the DICOM standard, and aims at fulfilling several needs arising from the research community at the intersection of AI and medical imaging: robust data curation procedures, responsible data access and effective data governance, robust de-identification and annotation pipelines, and streamlined data operations (e.g. model training, analysis). The development of PARADIM is inspired by the data-centric AI movement – stating that models are mature but quality data to train them are lacking – and by MLOps/DataOps approaches where research activities are integrated as much as possible in the operations of the healthcare enterprise. The platform, built from open-source components, is designed to facilitate and automate the execution of large-scale, computationally-intensive pipelines (e.g. automatic segmentations, dose calculations, AI model training). It is also well-suited for federated learning approaches, where models (but not data) are exchanged across institutions [2]. The platform stems from work presented at the 19^{th} ICCR, which featured the use of Orthanc [3] as a research DICOM server [4].

2 Materials and Methods

2.1 Overview

The PARADIM architecture is presented in Figure 1, showing the main technological components of the platform as well as high-level functionalities. The components are deployed in two distinct environments (blue and orange zones in the figures), corresponding to network areas belonging to the medical institutions on one hand and to other service providers on the other (in this case VALERIA, institutional Research IT services at Université Laval). From a legal perspective, at least in the jurisdiction PARADIM is deployed in (Québec, Canada), data containing Personal Health Information (PHI) can only reside in the medical network area; they must be de-identified before they can be transferred outside this domain. The de-identification component of PARADIM (Karnak) therefore lives in this area. De-identified images are then transferred to the main PARADIM ecosystem (orange frame in Figure 1) where they are stored in a DICOM server (Orthanc) [3]. An authentication/authorization layer (Kheops) manages access to images, either through a web portal (identities managed by institutional AzureAD) or through an API (revocable tokens). This component (Kheops) is the main entry point to images, by humans or machines, and allows the creation of image collections that can be enriched with annotations, typically by medical experts contouring regions of interest. The rest of the platform is dedicated to data operations (e.g. model training), which are orchestrated by a custom-built job dispatcher handling containerized applications. Components are deployed in an institutional OpenShift (RedHat) Kubernetes environment, and the majority of them offer a web-based interface and REST API functionalities that are used for orchestration and connectivity.

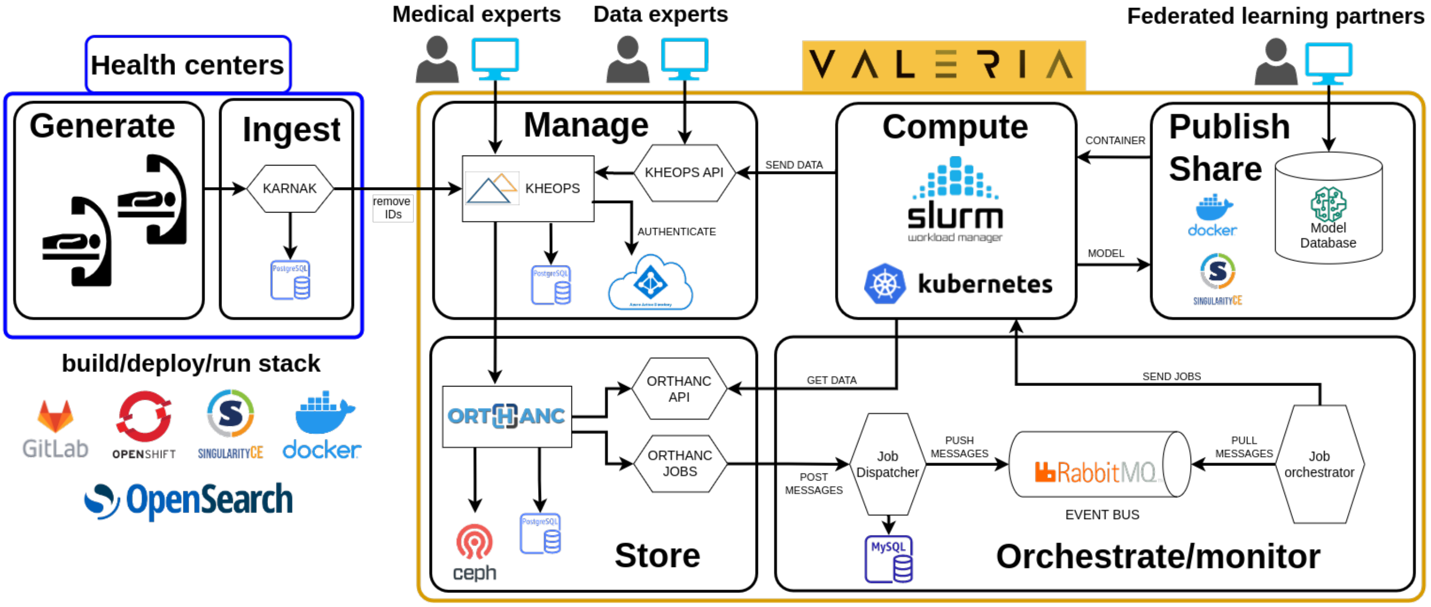

Figure 1: High-level system architecture of PARADIM.

3 Data de-identification

The Karnak software developed by the Osirix Foundation [5] is used in PARADIM as a DICOM de-identification gateway. Karnak provides several functionalities related to data ingestion and de-identification, which is not trivial considering the large number of DICOM tags that can contain PHI. De-identification, as offered by Karnak, typically consists in applying the default de-identification profile defined in the DICOM standard and to add customizations (e.g. keep the original value of a particular field, replace with a fixed value, or shift dates by a random number of days). It is also possible to apply a mask to images, which is useful to remove information engraved in images. De-identification profiles can be defined and reused in Karnak, depending on the requirements of each project (some project might actually need PHI to conduct research activities [6]). Karnak can generate new IDs for patients (based on hash+salt) or use a list provided by users mapping identities. The Clinical Trial Study DICOM module is used to document project information upon ingestion of images in PARADIM (including ERB approval number and type of consent associated with images). This marking of images on ingestion facilitates long-term management with respect to ERB and/or legal approvals.

3.1 Storage backend

Basic data management functionalities in PARADIM are provided by an Orthanc server [3], which natively manages DICOM instances and can be customized and extended in a modular way. In PARADIM, the Python plugin is used to notify the arrival of new studies and series, and to add new endpoints to the existing API. The DICOMweb plugin provides an unified interface to communicate with Orthanc, while the PostgreSQL plugin is used to store the main DICOM tags in a robust database for increased performance. Additional elements were added to the main DICOM tags list to interface with external components such as the OHIF viewer [7]. The AWS S3 plugin was configured to use the Ceph (RedHat) storage provided by our institutional research IT services. DICOM instances stored in Orthanc are immutable; additional information such as annotations, segmentations or derived information are stored in distinct DICOM objects within a study.

3.2 Data governance

Orthanc is well-suited to act as a DICOM data store but has limited functionalities to implement data governance rules, including user management, authentification and authorizations. For these purposes, the Kheops software component developed by the Osirix foundation was used [5]. Kheops uses an album system to constitute image collections, allowing DICOM studies to belong to several albums. Album parameters and metadata generated by Kheops are stored in a dedicated PostgreSQL database. Each album is managed by a data steward who controls access and rights, including adding and removing data or users, and the permission to share or download data. In PARADIM, identities are managed through our institutional AzureAD instance providing single sign-on (SSO) and multifactor authentification (MFA) functionalities. Revocable access tokens can be generated within Kheops to allow services to access specific data collections. Kheops therefore acts as an auth/autz layer on top of Orthanc, with functionalities mainly related to the curation of image collections. An NGINX reverse proxy was set in front of Orthanc to receive network requests and set up Cross-Origin Resource Sharing (CORS) rules that will limit sharing to certain authorized external applications only, including Kheops.

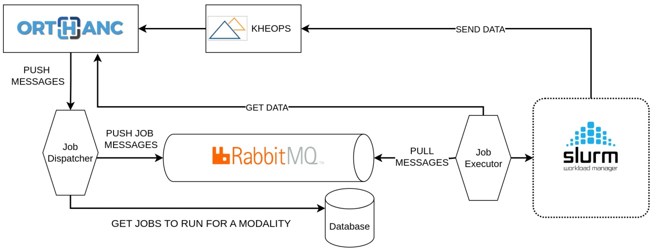

Figure 2: Job dispatching and execution in PARADIM.

3.3 Data analysis

Given the relative complexity of curating data and training AI models, it is important to rely on robust, traceable and ideally automated methods to generate results, even more so that research activities at the interface of AI and medical imaging require significant and expensive computational resources. PARADIM was designed with this in mind, and provides functionalities to automatically trigger compute jobs, and capture execution artefacts (e.g. logs) and metadata on all data operations, from annotations to versions of software used in analysis. These traces are kept in DICOM SR objects and stored along data within studies.

PARADIM users are required to provide their code in application containers that respects the Open Container Initiative specifications. These containers are stored in a private registry, and require only two parameters for execution: one input and one output directories. Internally, user-provided containers are converted to Apptainer images. PARADIM enforces the use of the DICOM format as input and output for all applications. Code snippets and utility functions packaged in libraries (PyOrthanc [8], PyGRPM [9]) are provided to users to ease the capture of execution artifacts and to convert to other formats (e.g. NifTI) if necessary, as some libraries natively require them.

For job execution, two software components were developed: a job dispatcher and a job executor. The first one is connected to the Orthanc API, which triggers contextual messages upon the reception of new images (e.g. a new lung CT study was received). These are sent to a message bus (RabbitMQ) acting as a buffer preventing the overload of compute instances. Messages accumulated in the bus are consumed one by one by the job executor, which 1) gets the relevant patient, study or series associated with the SeriesInstanceUID 2) generates a Slurm script containing the appropriate parameters for job execution. The data and the Slurm script are sent to compute resources and the job executor notifies the job dispatcher that a new job was submitted. Figure 2 illustrates the process.

In addition to jobs triggered by new images received on the platform, it is also possible to launch job manually, either from the command line or through a GUI developed with Streamlit.

3.4 Deployment environment

Components of PARADIM were deployed on our institutional OpenShift (RedHat) platform, using a DevOps-type continuous development and release approach orchestrated by GitLab. Files in YAML format define the different Kubernetes resources to deploy as well as the required resources. It also includes the number of replicas to deploy to ensure high availability, as well as the network configuration, such as the ports to open. All applications can be scaled except the RabbitMQ component to avoid concurrent access problems. Resources can be quickly redeployed or terminated using the GitLab task scheduler. Probes were developed to periodically enquire the state of applications. In case of failure, notifications are sent by email using the SysDig tool provided by our institution. Sensitive variables and information, such as database access, are managed with the GitLab Secrets Manager and dynamically inserted into configuration files during deployment.
Deployed instances, called pods in the Kubernetes nomenclature, are accessible inside the platform through services, and from outside through routes. For example, Orthanc has no external route, and Kheops accesses it through a service. However, Kheops is open to the external network (and users) through routes. The platform lives inside the institutional network, accessible either from campus or via a VPN connection. In all cases, connections are made with the TLS security protocol, with institutional certificates.

4 Results

This section presents use cases of PARADIM, for which the platform was instrumental in automating tasks or in capturing expensive information such as image annotations.

4.1 Annotation pipeline

One of the first use of PARADIM was the capture of image annotations by medical specialists. This information can be considered expensive, in terms of time and expertise, and it is crucial to preserve it on the long term, along with contextual details (e.g. who, when, how). Annotations often act as ground truths in AI project and in this regard they deserve robust methods for preservation and documentation. The current annotation pipeline in PARADIM is based on the use of 3D Slicer [10] and its QuantitativeReporting plugin [11]. 3D Slicer is used to fetch data from the platform (through authenticated DICOMweb transfers), to capture the name of the operator (stored in the ContentCreatorName DICOM attribute), to perform lesion contouring tasks, and to encode everything in DICOM objects containing as much metadata as possible (e.g. dates, software versions). This pipeline was successfully used at our institution to create lesion segmentations on CT for a cohort of approximately 2000 patients in lung cancer. It allows us to keep track of who made the annotations, when, and in which context. Considering that machines are poised to increasingly perform this type of task in the future, keeping a record on the origin of information is of paramount importance. For the particular case of lung lesion contouring in CT, capturing the annotator identity led us to unveil discrepancies in contouring methods used by different individuals – an important information that can have an impact on results generated.

4.2 Automated tasks

AI models are rapidly becoming commodities that can be consumed from various outlets (e.g. HuggingFace, MONAI), and integrated into complex pipelines. PARADIM facilitates this process and provides functionalities to automate it. As a proof of concept, the TotalSegmentator model [12] was integrated in a pipeline triggered each time a CT study of the thorax is received on the platform. This automatically created DICOM RTSTRUCT objects in the study, with rich metadata (e.g. execution log, software version). A database keeps track of objects created by a particular model, so that subsequent calls will not generate calculations unless the model version (in the container registry) is different. This MLOps/DataOps approach allows for large scale computational endeavours and continuous monitoring of model performance.

4.3 Monte Carlo calculations

Another use case, different from AI-based tasks, is the automatic recalculation of dose maps for brachytherapy cases [13]. The code was provided in a Docker container, stored in the registry, converted to an Apptainer format, and made available in the job dispatcher database. Each RTPLAN modality delivered on the platform triggers a calculation using this container (which also requires RTSTRUCT and CT images to generate results). The entire recalculation pipeline can be re-triggered manually on demand. This kind of automation streamlines the production of results and fosters the transition from research to the real world, through robustness and reproducibility. From a user perspective, the complexity related to storage and computing resources dispatching is masked.

5 Discussion

PARADIM fills a gap at the interface of AI and medical imaging, where data and digital infrastructures were historically not given the attention they deserve. Several improvements are planned for the future, including a fully web-based annotation pipeline (3D Sclicer client currently required), an interface with tools to extract semantic information from radiology reports and better support for whole-slide imaging in digital pathology. Scaling resources on-demand to external providers (cloud bursting) is also planned to deal with large computational tasks.

6 Conclusion

The PARADIM platform was designed to store, manage, enrich and responsibly access medical images in a data science context. It relies on the DICOM standard for maximal interoperability, and facilitates the execution of complex pipelines. It is used by several research teams for various tasks, and keeps track of as much information as possible on data operations, aligned in this regard with the FAIR principles of research data management.

References

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, et al. "The FAIR Guiding Principles for scientific data management and stewardship". en. *Scientific Data* 3.1 (Mar. 2016). Number: 1 Publisher: Nature Publishing Group, p. 160018. DOI: 10.1038/sdata.2016.18.

[2] L. Mullie, J. Afilalo, P. Archambault, et al. "CODA: an open-source platform for federated analysis and machine learning on distributed healthcare data". *Journal of the American Medical Informatics Association* (Dec. 2023), ocad235. DOI: 10.1093/jamia/ocad235.

[3] S. Jodogne. "The Orthanc Ecosystem for Medical Imaging". en. *J. Digit. Imaging* 31.3 (May 2018), pp. 341–352.

[4] P. Després. *Harnessing the potential of data in clinical PACS with an open-source DICOM server*. eng. Publisher: Zenodo. Sept. 2019. DOI: 10.5281/zenodo.3450560.

[5] *OsiriX Foundation*. en.

[6] S. M. Moore, D. R. Maffitt, K. E. Smith, et al. "De-identification of Medical Images with Retention of Scientific Research Value". eng. *Radiographics: A Review Publication of the Radiological Society of North America, Inc* 35.3 (2015), pp. 727–735. DOI: 10.1148/rg.2015140244.

[7] E. Ziegler, T. Urban, D. Brown, et al. "Open Health Imaging Foundation Viewer: An Extensible Open-Source Framework for Building Web-Based Imaging Applications to Support Cancer Research". *JCO Clinical Cancer Informatics* 4 (Apr. 2020), p. CCI.19.00131. DOI: 10.1200/CCI.19.00131.

[8] G. Couture, Y. Lemaréchal, and P. Després. *PyOrthanc*. Sept. 2022. DOI: 10.5281/zenodo.7086219.

[9] Y. Lemaréchal, R. Lefol, P.-L. Asselin, et al. *PyGRPM: a Medical physics library containing many utilty functions*. Aug. 2023. DOI: 10.5281/zenodo.8250064.

[10] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, et al. "3D Slicer as an image computing platform for the Quantitative Imaging Network". en. *Magn. Reson. Imaging* 30.9 (Nov. 2012), pp. 1323–1341.

[11] A. Fedorov, D. Clunie, E. Ulrich, et al. "DICOM for quantitative imaging biomarker development: a standards based approach to sharing clinical data and structured PET/CT analysis results in head and neck cancer research". en. *PeerJ* 4 (May 2016), e2057.

[12] J. Wasserthal, H.-C. Breit, M. T. Meyer, et al. "TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images". *Radiology: Artificial Intelligence* 5.5 (Sept. 2023). Publisher: Radiological Society of North America, e230024. DOI: 10.1148/ryai.230024.

[13] S. Ouellet, Y. Lemaréchal, F. Berumen-Murillo, et al. "A Monte Carlo dose recalculation pipeline for durable datasets: an I-125 LDR prostate brachytherapy use case". en. *Physics in Medicine & Biology* 68.23 (Nov. 2023). Publisher: IOP Publishing, p. 235001. DOI: 10.1088/1361-6560/ad058b.