

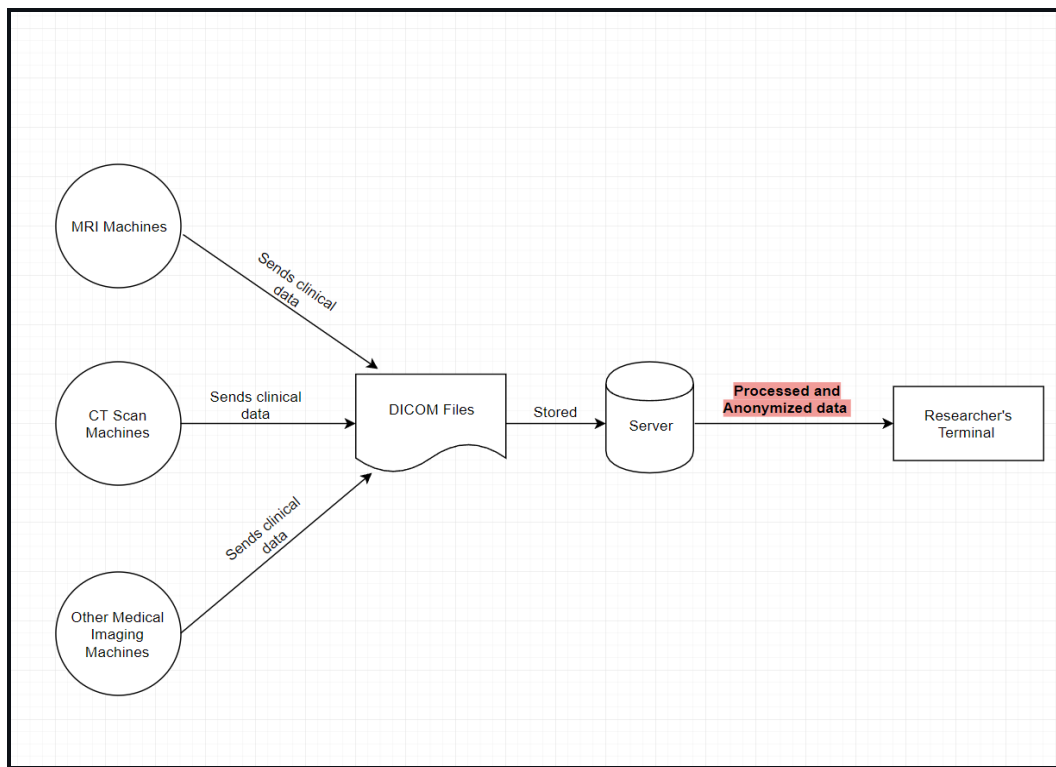
A high-level description (about one paragraph) of the software you are building and who the target user groups are.

Medical research of the modern world has become more and more reliant on data as well as processing it on larger scales. AI has become an invaluable asset to the medical research field. One major concern that many researchers face is lack of access to clinical data. This project introduces a pipeline that streamlines the process of secure access to medical data for researchers within the BC medical system. The project seeks to anonymize real patient data from clinical studies, and streamlines researchers' access to the data.

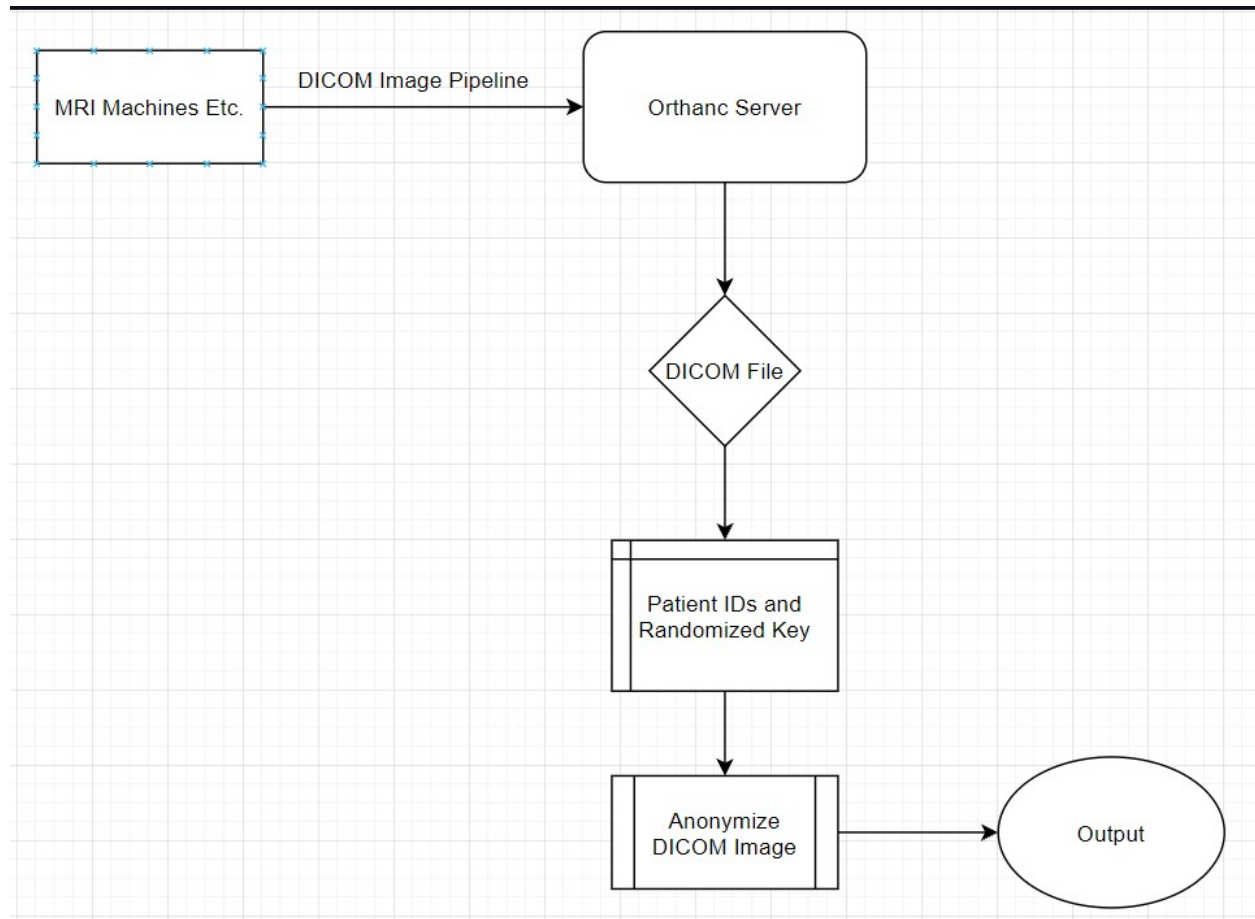
The intended audience is for BC researchers who require access to processed and anonymized medical data. If approached with the right steps toward implementation, we believe that the pipeline could also be useful toward researchers in other fields and regions who require access to clinical data.

An overview of your system architecture presented as a data flow diagram (DFD), at both levels 0 and level 1

Level 0 DFD: Medical imaging machines such as MRI or CT machines, will send an instance of patient data along with a copy of their medical image in the form of a DICOM file to a server. The data will be processed and anonymized before presented to researchers based on their query. Our purpose and goal is to de-identify and anonymize the dicom file. Our hope is that the plug in can be widely used by researchers.



Level 1 DFD



A detailed list of functional requirements for each milestone.

- ❖ Milestone 1
 - Complete Data Flow Diagram (DFD) level 0 and level 1
 - Complete Software Requirement Specification Document (SRS)
 - Set up weekly meeting with client
- ❖ Milestone 2
 - Set up Orthanc environment
 - Set up server
 - Determine language (Python or C)
- ❖ Milestone 3
 - Complete all functional requirements

- Complete Peer testing # 1
- ❖ Milestone 4
 - Complete peer testing #1
 - Project deliverance

A detailed list of non-functional requirements and environmental constraints

Non-functional requirements:

- Security
 - DICOM file before de-identification and anonymizing contain personal health information (PHI)
 - Access to anonymized data must be limited and monitored
- Reliability
 - Robust design to prevent breaches on PHI
- Maintainability
 - Admin needs to be able to change the perimeter of which part of DICOM file is to be de-anonymized

Environmental constraints:

- Lack of access to facility server for testing, as it contains PHI

Identify the tech stack you plan to use.

Based on the information given by the client, we believe it is best to use python and various packages that come with it. Since we are dealing with lots of data processing, python contains various packages that will help us such as pandas and numpy.

We've already raised the issue with the client, he will approve either plugin written in C or python.

An explanation of how you will test the developed features of your system

In terms of testing, we would like to see how the plugin would be able to handle mass amounts of data, and connect to BC cancer servers, but that would be up to client discretion as we will be using PHI (Patient Health Information) from real patients.

The most likely case of testing is that we will feed DICOM file containing sample information through the plug in, although this method will reveal either desired information on the DICOM file is being anonymized, it cannot account for the robustness of the plugins when faced with large number of DICOM file being processed.

Conclusion:

Based on the information given the client, we currently only have one functional deliverable, but it is expected that upon the completion of that deliverable, the client will assign more functional deliverables.