Design Document

High-level Description:

The project aims at creating an automated x-ray image pipeline, to provide mammography scans to experts to analyze using AI for cancer prediction of the breast tissue.

Current process and problem:

Currently, a female patient attends a clinic to receive a mammogram. After an x-ray scan is complete an image file is created called a DICOM file. DICOM files contain the actual image as well as patient metadata. Patient metadata includes identifiable patient information such as, name, sex, birth date, patient id, clinic id, etc, and contains non-identifiable information such as time of the scan, type of machine for scan, image view, etc.

The image is sent to a medical server via a secure medical image protocol called DIMSE. Those images are then stuck on that physical server in the clinic. Dr. Rajapakshe then has to physically drive to the clinic (approximately every 6 months), remove the hard drive, download the data onto a research server, delete the old DICOM files from the hard drive, and finally, drive the empty hard drive back to the clinic.

Once the files are on the research server, they are anonymized. The purpose is to open these images to scientists to analyze, but these DICOM files must not contain any Patient Health Information (PHI). There are then two servers. One contains the anonymized DICOM images and the other contains only patient metadata, which is kept confidential. The anonymized DICOM images are then released for research.

Solution:

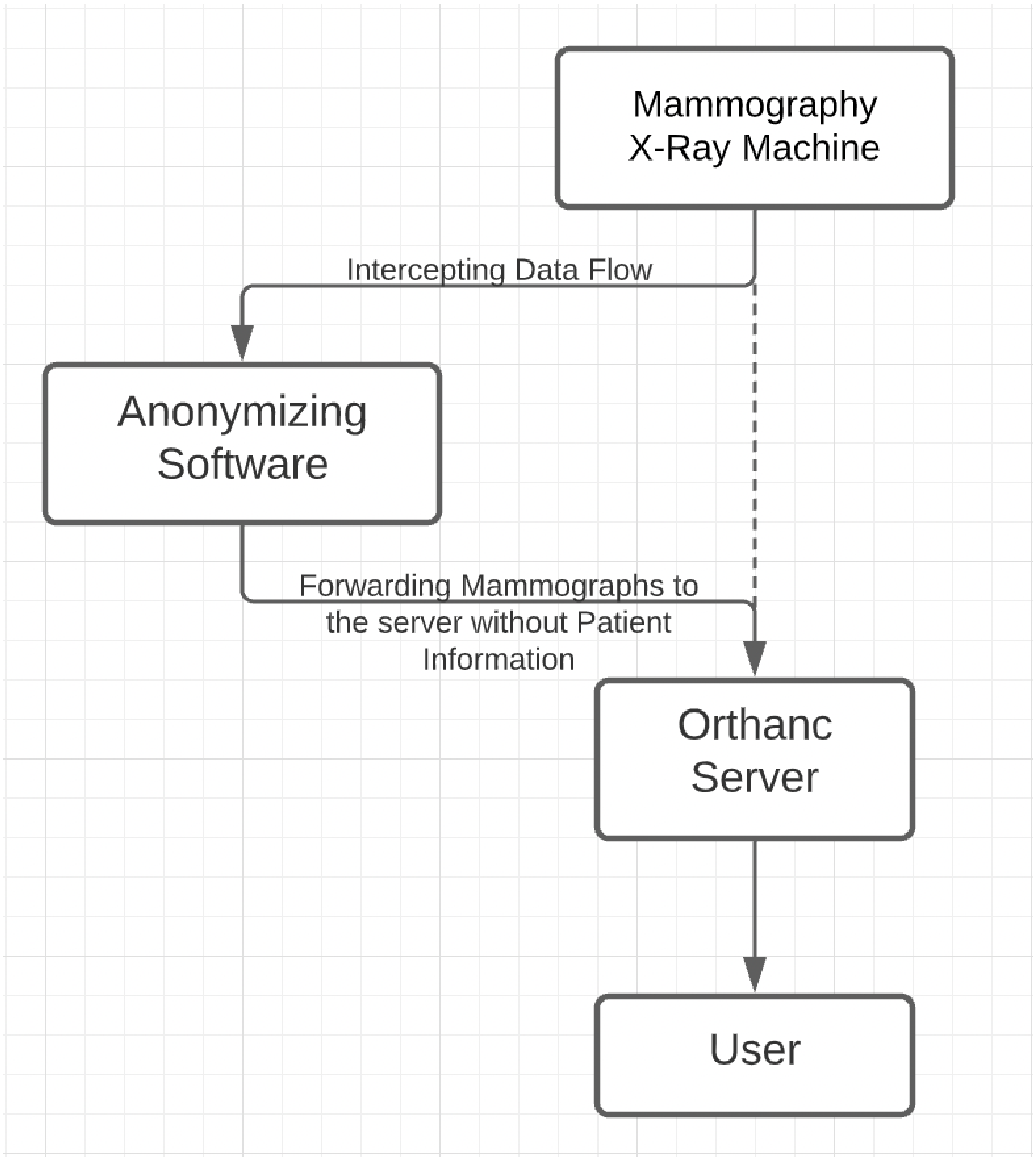
The proposed process will set up two new servers using open-source software called Orthanc. Once the mammography scan is complete the DICOM file will be sent to the first Orthanc server (via DIMSE protocol) which will store the confidential metadata, anonymize the DICOM file, and create a random new ID to connect the patient health information to the anonymized DICOM file. The anonymized DICOM file is then sent to another Orthanc server. This server has a GUI (Graphical User Interface) which will be used by researchers to download the x-ray scans and perform analysis.

This project will save Dr. Rajapakshe time, and allow for a faster release of new anonymized DICOM files. Instead of having to wait 6 months for new scans, the scans will be available to research almost as soon as the scan itself is complete.

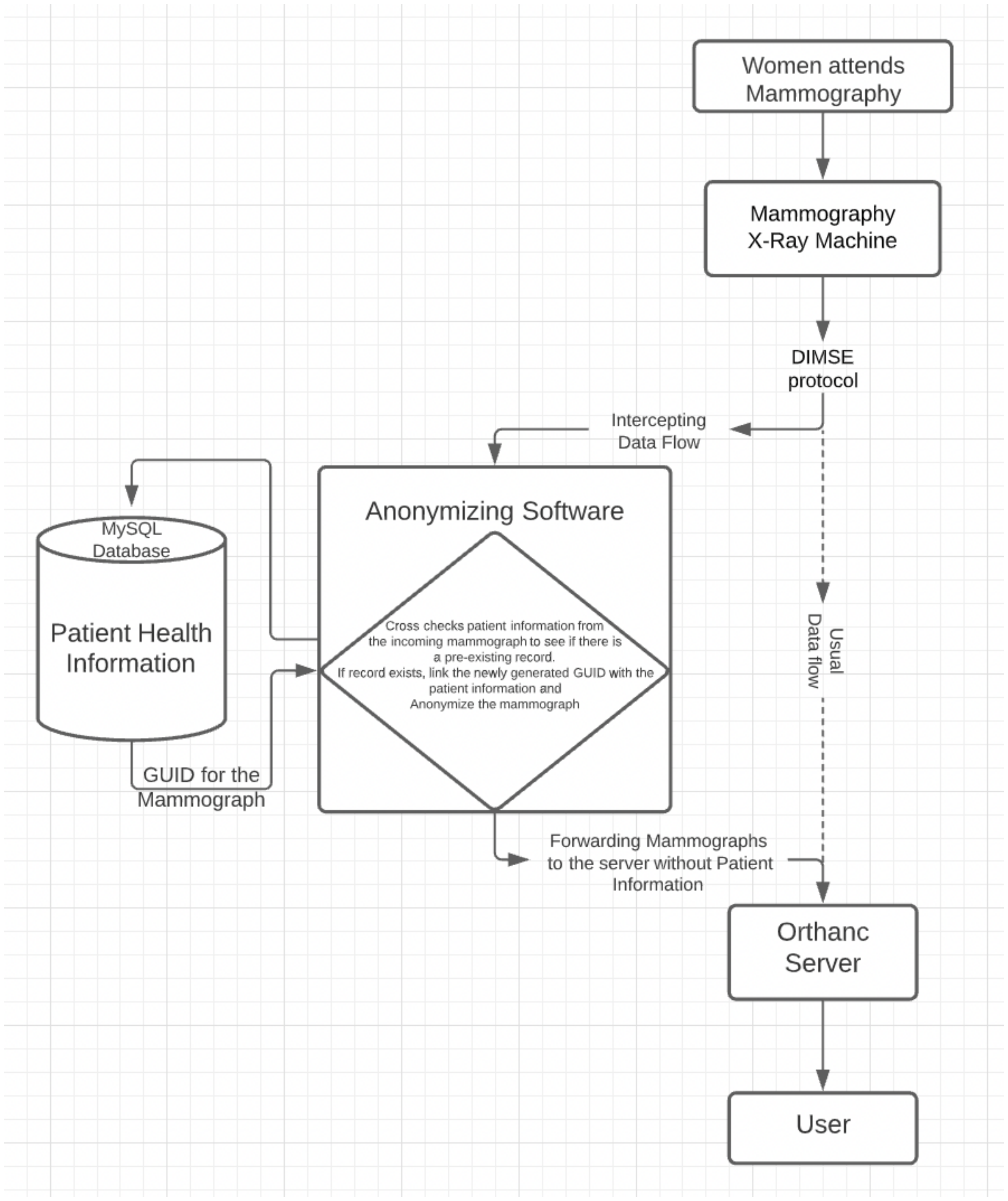
Development Plan :

System Architecture Overview:

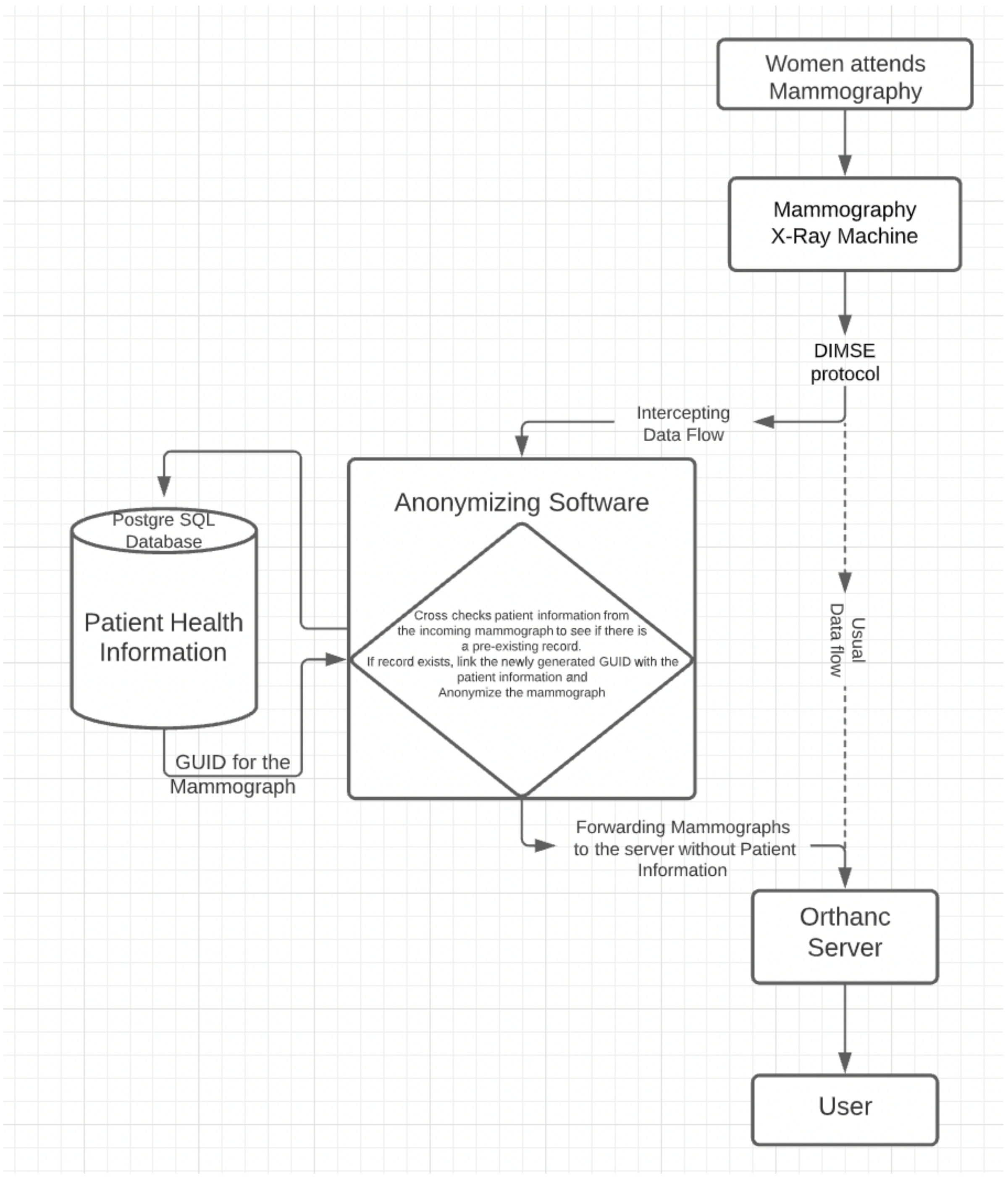
Level 0 DFD


Mammography X-Ray Machine
Intercepting Data Flow
Anonymizing Software
Forwarding Mammographs to the server without Patient Information
Orthanc Server
User

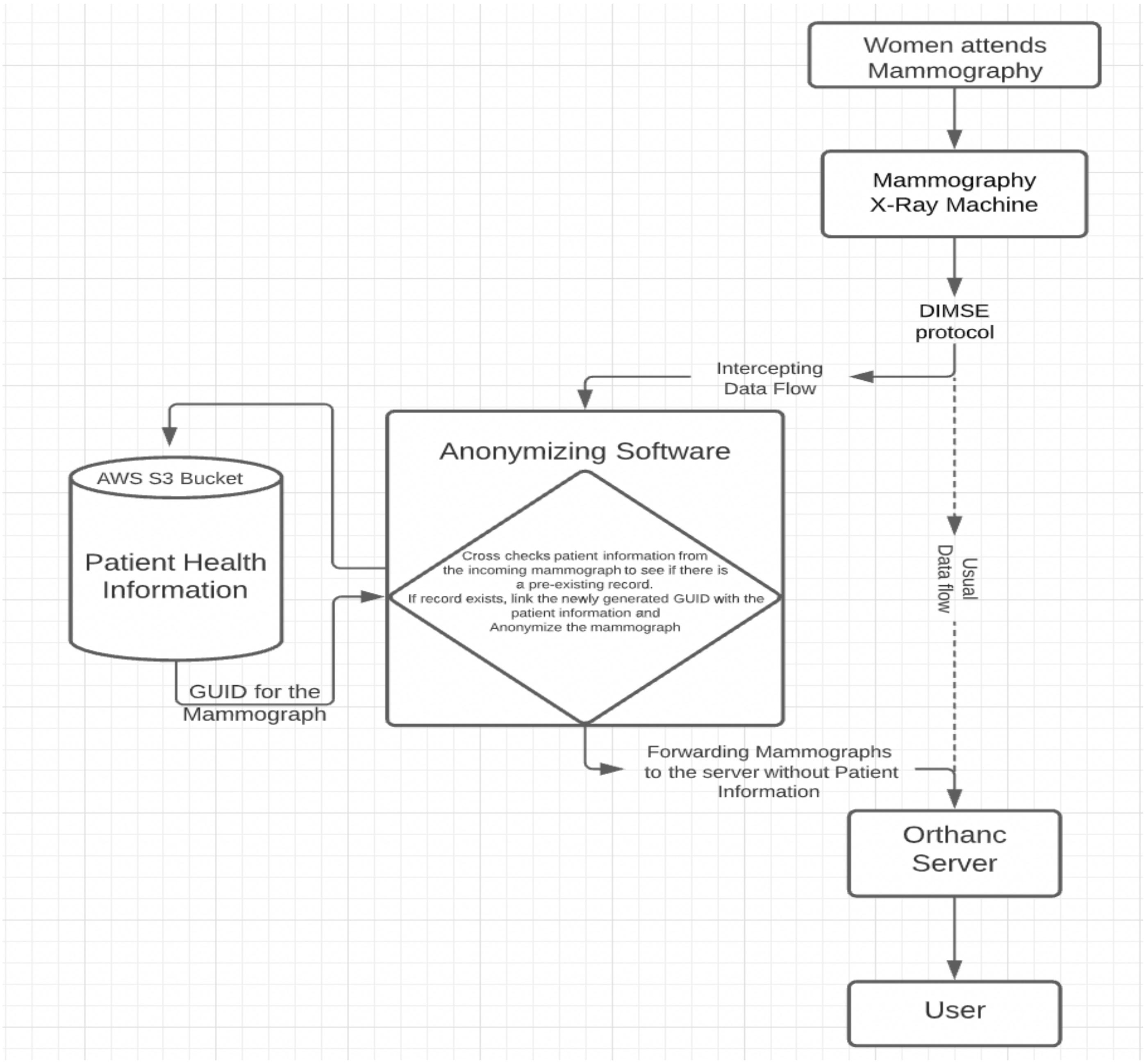
Level1 DFD with MySQL


Women attends Mammography
Mammography X-Ray Machine
DIMSE protocol
Intercepting Data Flow
Usual Data flow
Anonymizing Software
Cross checks patient information from the incoming mammograph to see if there is a pre-existing record.
If record exists, link the newly generated GUID with the patient information and Anonymize the mammograph
MySQL Database
Patient Health Information
GUID for the Mammograph
Forwarding Mammographs to the server without Patient Information
Orthanc Server
User

Level1 DFD with PostgreSQL


Women attends Mammography
Mammography X-Ray Machine
DIMSE protocol
Intercepting Data Flow
Usual Data flow
Anonymizing Software
Cross checks patient information from the incoming mammograph to see if there is a pre-existing record.
If record exists, link the newly generated GUID with the patient information and Anonymize the mammograph
Postgre SQL Database
Patient Health Information
GUID for the Mammograph
Forwarding Mammographs to the server without Patient Information
Orthanc Server
User

Level1 DFD with AWS S3 Bucket


Women attends Mammography
Mammography X-Ray Machine
DIMSE protocol
Intercepting Data Flow
Usual Data flow
Anonymizing Software
Cross checks patient information from the incoming mammograph to see if there is a pre-existing record. If record exists, link the newly generated GUID with the patient information and Anonymize the mammograph
AWS S3 Bucket
Patient Health Information
GUID for the Mammograph
Forwarding Mammographs to the server without Patient Information
Orthanc Server
User

Functional Requirements:

Functional Requirement of:	Functional Requirement
Orthanc server with x-ray	The first Orthanc server should handle requests from the x-ray machine via DIMSE protocol. When a woman gets mammography the x-ray machine will send the DICOM file to the Orthanc server.
Orthanc Server 1	The Orthanc server will then store the DICOM file PHI (Patient Health Information), and anonymize the DICOM files. The anonymized DICOM file and the PHI will have a random GUID linking them. The Anonymized DICOM file will then be sent to the next Orthanc server.
GUID	The GUID is created as a crosswalk to be able to link anonymized DICOM files to their respective patient in the PHI database on Orthanc server 1.
Orthanc Server 2	The Server will hold receive anonymized DICOM files and reveal them to the researchers. The researchers will be able to download anonymized DICOM files for AI training.

Functional requirements for each Milestone:

Peer Testing 1:

- User Sends DICOM file from one Orthanc server (Acting as x-ray machine) to another using DIMSE protocol.
- Orthanc Server 1 Receives the DICOM image and anonymizes it.
- The server sends an anonymized DICOM file to Orthanc Server 2
- Orthanc Server 2 is accessed by peers who then download the DICOM file.

Peer Testing 2:

- An Orthanc server (acting as x-ray) sends DICOM files to Orthanc server 1. Orthanc server 1 partitions the PHI and anonymized data.
- Orthanc server 1 creates a GUID as a crosswalk.
- Orthanc server 1 send anonymized DICOM files to Orthanc server 2.
- Peers download the anonymized data.

Final Product:

- X-ray is configured to send DICOM files after scan to Orthanc server 1. Orthanc Server 1 catalogs and cross-references similar data to make sure the patient is unique
- DICOM files get anonymized and sent to Orthanc server 2.
- Orthanc server 1 store PHI on a relational database.

Non-functional Requirements and Environmental Constraints

Security:

- The most important non-functional requirement is security. The security of the PHI is very important

Database Read Speed:

- It is important for our database to have the fastest readability because we will be constantly cross-referencing data to see if a patient with the wrong id is already in the database.
- We must also read quickly to determine if data is part of a new study.

Tech Stack

Programming language:

Python:

- As we all know that maintaining datasets, storing the correct information and transferring the correct information about a patient is really important in the medical imaging field and python provides us with a massive advantage due to its vast collection of libraries.
  In reference to this project, Python offers libraries like Pydicom and plugin packages for Orthanc Server

C#:

- C# is used in creating the Orthanc server. It will be simpler to use this language.

Database:

There are two main options for storing patient metadata and DICOM images. We could use a relational or non-relational database.

Approach 1: Relational database MySQL

Using MySQL would be our definite choice and we could use Microsoft Azure to make it remotely accessible for every single admin user.
Even though MySQL comes with a lot of restrictions and licensing issues I feel like the fastest and the safest way to secure the amount of sensitive data that the application will be using MySQL would be the best choice.

Data Storing Decision Table:

MySQL	PostgreSQL
Relational **database**	Object-relational **database**
Does not allow for table inheritance and function overloading	Allows for table inheritance and function overloading
Does not support AS many number of data types.	supports a number of advanced data types
Has licensing issues	Truly open-source
Faster under heavier workloads	Slower under heavier workloads
Does not offer materialized views	Offers materialized views

Testing

Our testing will have two phases:

Unit Tests

System Tests