TMI Group 2:

1. (Daniil): Great definitions, I felt like you did a great job providing background.

2. (Daniil): You mention Orthanc is a DICOM server? This is partially what it is but would it be better to describe this as a PACS server? As PACS is more encompassing then just saying DICOM server?

3. (Daniil): In addition to feedback point 2, it might make sense to add the definition of what a PACS server is.

4. (Daniil): Adding to this, there is a discrepancy in your presentation,

- Brain calls Orthanc is a PACS
- Raouf calls it a DICOM server

This is a nit but thought you guys should know.

5. (Daniil): The architecture looks great but there is a part that is confusing to me:

- Why is there an arrow from anonymize data to find matching records?
  - Why would you be trying to match records with an anonymized file?
- When I see the diagram I assume anything that passes through the anonymize data part I assume it to be anonymized. In this case it is confusing to me because where are you storing PHI (Patient Health Information)? Because the arrow goes directly from anonymize data to database. Therefore I assume no PHI is hitting the database.
- It is unclear from the diagram if you have two Orthanc servers. If they are the same why are they two different shapes?
- Ok as I look at the diagram more I see the PHI can flow to the file system.
- Shouldn't there be an arrow going back to Orthanc? How does a researcher download the anonymized files? Because the arrow only goes one way.

6. (Daniil): I like the decision for c++

7. (Daniil): Great test strategy! Once comment, I feel like you could add some edge cases that we should be thinking about regarding system tests. As we already know some system tests that we will be running example:

- What happens if we have the same patient id but different PHI?
- What happens if same patient ID same PHI except for a small typo?
- what happens if same PHI but different patient ID?

(Daniil): Overall really nicely done good job!

TMI Group 0 Feedback:

1. (Daniil): Love the background music! Also nice background for slides, we have the same one :)

2. (Daniil): You call Orthanc a DICOM server wouldn't a better term be that it is a PACS server? This is a nit but from my understanding, PACS server is more encompassing term then DICOM server.

3. (Daniil): You mention briefly what we are trying to replace but you only state that it is old and outdated. Just because something is old does not mean it should be replaced. My feedback here is you should a more explanation of the existing problem and what we are trying to fix. From watching the slide it is unclear why your new system is better than the old one.

4. (Daniil): I have a nit for your DFD:

The machines send the DICOM files already, what I would do is instead of having the arrows say “Sends clinical data” I would say “Sends DICOM files” then delete the “DICOM files” box and point the arrows directly to the server.

5. (Daniil): It is unclear from your DFD level 1 where you will be storing the PHI (Patient Health Information).

I see you are showing the anonymization process but what about the cross-referencing process. You touch on it in the presentation but it is not showing in the DFD.

6. (Daniil): Have you considered the loss of speed that you might get from using python? Yes python has many libraries but as it hides implementation of those libraries don’t you think you will not be able to maximize speed of your database performance? Isn’t Orthanc already optimized for C++? Pandas and Numpy are also slower because they hide the details and are written in C don’t you think it would be better to stick with C/C++ as it will be able to maximize the cross-referencing you will be doing of the DICOM files? Thos are big files and there will be a lot of them, also deletion speed of records is going to be very important, take a look at Orthanc/books/scalability to see how deletion needs to be optimized.

Overall very well done, but I would like to hear more of a defense for why you are choosing python :)

MIM Team 1:

1. (Daniil): Great problem description.

2. How will the matching be done? Is there a database involved in this process? Where are the files stored?

3. I understand the minimum to send is two but what is the maximum? Can I send 100 files to be checked? Can I send a whole study? What is the cap and why does it exist? How much will the speed of your system be effected by sending a large number of DICOM files to be matched?

4. I am guessing this is done with DICOM images right? It is sort of unclear in this case because then you are not actually comparing images but PHI (Patient Health Information), maybe you should be more explicit about this?

5. You show that the data is sent in the diagram but some questions are unclear:

- sent from where, from a PACS?

- And how is it sen, Via DICOM protocol?

6. Maybe this is a misunderstanding on my end but why are you using AI? If all you have to do is check if there is a match why do you need AI? Because unless you plan on actually modifying the actual PHI all you need is a relational database with a really high read speed and a comparison software? I could see using AI if you will modify PHI but I don’t think our health care system will allow this because it must be done usually by a medical professional.

7. Speaking of databases, if you have a lot of records coming through (thousands) how will one script handle this? What kind of database will you use relational or non-relational? What kind of availability are you guys looking at? How much downtime is required to adjust the script, or will it be self-adjusting? Can you run multiple comparisons in parallel? Have you thought about parallelizing this?

Great presentation, I know you only have 10 min but I feel like there could have been more details regarding the mamograms, what they are, and how you are checking them. I feel like someone who does not know about DICOM would be confused about what exactly you are comparing.

MIM Team 0

1. I feel like you glossed over important details such as, what is a DICOM file?

2. What is in a mammogram file? You weren’t very explicit where these fields are located or how the file is structured or what a DICOM file is which is very important in order to understand how exactly you will be comparing fields.

3. I am assuming you are comparing PHI (Patient Health Information) could you be more clear about this? What exact fields do you assume will usually be off? Many of those fields are not manual entry, therefore you should be expecting certain fields to be off more times than others? For example: it is unlikely the date and imaging modality fields will be wrong. I think you should be specific about this.

4. What is the max throughput here? Will one script be able to handle thousands of records? Can you run them in parallel? Why or why not?

5. What kind of database do you plan to use and what kind of availability do expect? Will it be relational or non-relational?

6. If you only plan on checking matches with PHI why not discard the image altogether and store it in a separate database with a cross-walk GUID? That way you have a smaller data file to make comparisons increasing the speed of the comparisons.

Great presentation overall!

AGMEETING Team 2:

1. It was unclear what kind of database will you be using, will it be relational or non-relational and why will you go that route?

2. I am guessing your system needs to have high availability? You should be more explicit about this. How will it handles records? How much downtime is too much downtime?\

3. How many records do you plan on having, thousands, millions? Do you plan on partitioning you database? Will you utilize distributed computing, what about cloud computing? Why or why not?

AGMEETING Team 1:

1. How do you plan on managing availability? What is the maximum amount of downtime you allow?

2. What kind of database will you be using? Relational or non-relational?