Team know how to handle if there are issues with the divisible by three errors and you know my store can't have those images posted on the viewer you know the specific constituent. Over to you. Yeah. Yeah. Perfect. Thank you. All right, Patrick, before we get into it, are you extending the panel on the videos as we go? Yes. OK. Great. Thanks. Anything else?
All right. So dot com web server, it is based off of a project called DICOM Cloud that was open source and was first put into place for the responsible. And when we start with the Maestro project, there was a couple of differences. So technically there are two Maestro or there's DICOM web service and there's two dot com viewers. So the DICOM web server retail is for the sponsor portal, the DICOM web viewer is for the sponsor portal as well. And then I created a repo called Maestro dot com for Maestro and.

viewer is for the sponsor portal as well. And then I created a repo called maestro dot com for Maestro and I was never able to merge everything back, but.
The back on whatever, Maestro.
Is running the latest versions of net.
Forget we went to 7 or not. Let's see.
Yeah, pretty good. So whereas the dot com website I think is or the sponsorportal.com is probably running version of something like that.
No, I guess I'm not, sorry.
That's what I was talking to. Isn't that visual? I was just there's two repos here, useful, responsible, and then all the maestro dot com is in the maestro dot com repo and in there there's two two versions of the viewer, which I'll talk about a little bit later

viewer which I'll talk about A little bit later. And then the dot com website was talking about now.
So start, we'll take a look at the database.
Structure. It's very simple and this is kind of the first step when somebody says oh, there's an issue with AA dot com where I'll go first. So in dev and val, the DICOM database is actually in the app server. Production has its own DB server called prod .com.
So take a look here.
There is really only one, two, three, four tables that matter. This one doesn't matter. One doesn't matter. It's just there because that's that's what they originally had and then it has some sort procedures as well, in particular to the whatever code we'll call these to basically pull data and then there's also some methods to insert

well, in particular to the whatever code we'll call these to basically pull data and then there's also some methods to insert.
To insert image, take insert image, the person's called by Maestro Wind, you know, patient doesn't really matter for us because it's all blinded type of stuff. So we take a quick look, you know, we'll have a patient ID but there's nothing really helps us that's in there. We don't do anything else. The thing that is most important is the dot com web server setting. This is where it has relations, patience. So that's why we do an ID. But when when somebody says oh I want to do an image, it will be the base address of the viewer and then it's always slash the settings. So that's the unique identifier. Then the study ID column here is the BTR, ID. And then study date is actually the the scan date.
That that's part of the dot com tags.

Before I go further, any questions on?
So within a study which is really just a scan, there could be multiple series. So if if I'm looking for particular images and maybe where they just in S3, I'll take the study key, I'll go to the series with that study key and then this could be one or more series. And then the series also has its own unique, I'll tell you the type of modality of the series, 123, something along those lines. And then finally I'll go through the object instance with the series key to find a particular set of images. So this one just has one image associated with series.
Generally do have a lot more, and again also has its own instance

ID. But I was looking for the files this path, the file path, this is where it exists in the S3 part. So since I'm looking at dev, that means it's going to be in the development BTPR, file uploads location in the submissions folder if you would and then it actually gives you the dicom name to the file name as well. Dicom tags is the dicom tags in the file, but it is inserted into the database in. Forget what I think it's, I think it's binary format just so that we don't have to open the file like when the viewer is requested information, just kind of return it quickly.
So when there are problems, I'll kind of usually go die down to this level, pull the pull the files from the files, they'll upload it into a

lower file so differ out, and then just try to view it right. And then there's two test errors. Sometimes there's errors when it's pulling it out of the database, so like you'll get a 500 error from the datum website and then sometimes there's it pulls it correctly, but the viewer is incompatible. So first thing I'll do is I'll offer logs on dot com web server, make sure that it pulls the file correctly. A lot of times it does and there's various reasons for that, but there are postman calls that you can do to to pull this.
It might have wanted to be an issue. In the past there's been problems with, you know, two large files were incompatible.
So the DICOM Web Server API really only has the data structure of calls, right? So we can

calls, right. So we can get a study, get studies right. And one of the things that was done inside recently but within the last year is based off of the user that came in determinants, are they authenticated whether or not it's AAPI call here and then if they have access to the study that you're trying to do and if they don't? If they do, then they'll be giving back the data that they're now.
The studies also does have a post. This is for Jet to upload.
There is a two gigabyte limit. I guess in theory that could be increased, but we haven't reached that issue yet. When?
The icon viewer.

The icon viewer calls to get images. Usually it calls the frame controller to get information. Again you'll see the thing where it will have to use information when you want to this real quick to sidetrack this. A lot of times somebody says oh I can't see an image. Sometimes it is permission based so you have to make sure that you know does the user exist, are they assigned to the appropriate permissions and studies studies. But it does cash. Once it looks it up once, I think it's 10 or 15 minutes but it will call the office authentication API.
With the off zero user information, once it has that checked in, if you can't find it then it's going to look for the user by the email against the camp. Once it has that then it needs to pull basically the all studies information so

the all studies information. So that's what the stamp call is. If it's open do they have all studies? They all studies then if they don't then we need to then go to the study service and get studies that are assigned to that user. Question here in terms of checking is that within the DICOM application the if we were to go down and try to check for that user's permission.
Would we be looking at the DICOM Application Permissions OK? Thanks.
So that basically happened on each of the API calls here. And then when it also calls the API, you know it's going to do the scope exists stuff that all other Apis do. So there's two types of users that come in. It could be an all zero user or it could be you know

coming from Maestro and then it's got the cookie for that. So it handles those two different types of.
All the stuff that is specific to kind of ETR whereas is in the bio research dot AWS project. The main items are the storage service. Storage service is where it goes to the database information. This part that commented out, three that's commented out is when this runs in AWS a role designed to the note that is running us to give access to the S regarding. Previously we provided a access key to get access to the blocker. So if you're debugging this locally you need to have your AW Explorer set up with a profile and then that basically does the same thing as this.

basically does the same thing as this. That's just.
Their kind of historical reasons. So there's two different types of or depending on what API and what you hit.
So if you're just gonna study information and it's just pointing some information, send me back contacts and then start that same thing series and then with instance it, it passes the series information as well as the objects.
Put him in topic that is trampled back and then on each of these you'll see the part where it's going to do with your all studies, then you're just allowed to do it. Otherwise check and make sure that the study codes that to the user also are good on that particular object. If it's not then it will just basically the other big one is the AWS location. This is where it actually pulls the files

AWS location. This is where it actually pulls the files out of S3. So actually it's going to call redicom file.
In there, it pulls the image out, it will copy it to a stream, and then it's going to open the file and then this is what it does for viewing within the Ohif.com viewer. We'll talk about a little bit, but basically it's setting the transfer syntax what's available, then it's returning the uncompressed pixel data for the image. It steps where the frame is, writes it to a stream.
This sometimes the request includes the training number, so that's what this is saying. Hey, get this specific ranking. If it doesn't include that, then it's going to go over all frames and write it to a stream. After it has a stream, sets the screen back to zero so that when it returns it returns all the information

when it returns it returns all the information.
Yeah, so this is just a support method that happens. Basically it's just copying the data out of the buffer onto the stream.
Are you going to like these loggers? Like are you able to look at data dog and kind of usually that's not usually not an issue.
The only thing we used to have an issue with this, I used to do the content like I used to do it that length on the screen here and that throw errors if it was too big a content link doesn't seem to be a problem so now we just check that it's not all that it had OK response and then it reads. But generally pulling pulling the file out at three and reading it into a stream is not not been an issue where you could get into problems is

you could get into problems is inverting it to this uncompressed pixel, but again that generally doesn't choose.
For the other parts of the code they showed us before we'd be logged, yeah? Are you usually able to determine the issue just from the logs?
Sometimes, a lot of times I'll run through points and see where where it breaks and then a lot of times it's what they modified the the data to not be compatible. This code supports if I tell that they can convert it to something else and then I see. Thanks. You're welcome. Now there's one other thing that we have the read entire file. This was created for Jet because here it's just returning the pixel data whereas here is 1 point to return basically the raw file at S3.

I believe we've got to compress the wrong compressed. Oh yeah. If it's not compressed, that means return the whole thing out of S3. I think the Sammy did this or something. I think Jet might.
Do this for exactly but this is very similar again to the the one above where it changes transaction taxes, memory stream will change the OR get the data and then this just kind of saves it to the stream. And then there is upload function which is used by Jet. So if they call the upload end up in the folder and then the key is going to be provided and as well as the streaming of the file. This is just an example of how you put metadata on that object but it's not done currently but it's done for a different process.
And then I'll also attach the postmark collection that I use for the dot com web server to the documentation page as well.

This code to just kind of be able to.
Anyquestionsonthe.com website.
Is this deployed into a Hikas cluster?
Yeah, some of the Dagon web.
Another set, sam. Sam's decently familiar with it. I think you played around with the last person. Yeah, I was gonna say I've stepped through the methods in there once or twice, and it was honestly

pretty easy once I knew to comment out the authentication pieces and what exactly to do to get it set up for the first time.
That's good to hear.
From the beginning, because the person that originally wrote the code obviously stated a lot of stuff. So let's say you go into there's two basics or two basic services, Kaido and Watto. Kaido is more like a query, wado is more the actual image. So we we dive into. Dive into this, you'll see that, oh, this other thing. So then you have to dive into that and then, oh, this is all another thing and then you kind of have to dive into that deal. So it's, it's many layers deep to find out where it's actually running the specific things. So using a stepped into type of things really where it comes out, right. So here I have my record one, because this is where it's going to

actually create the SELECT command against the database to make sure that it pulls that information correctly. So this is just searching for, you know, the images that might be associated with that study. But I also know that 0 service I can just put a break point here but it doesn't get to the break point because it broke somewhere up above. So there's depending on what you're looking for you might need to start at a higher level when it comes in or you might know OK I know go to this area so.
And then the the like a return code is based off of the dot com standard. So there's a DICOM web standard like if you just google it, if this doesn't follow everything exactly, but the methods it does have does follow it. So it is faithful to that to a certain degree, but like I said, it doesn't implement everything because a lot of

like I said, it doesn't implement everything because a lot of things we don't use.
So we don't have like delete functionality. It has very limited update or upload functionality. It's more just get the images out and then to to put the data into the system. We're more calling, it's faster for us trying to utilize this this one service and we're not doing any you know image manipulation you know within the system either you know Jet. Jet kind of does that which has its own process for doing that and it'll save the metadata by the web source.
All right.
Any other questions on that?
Now let's talk about the

Now let's talk about the dot com viewer. So the dot com viewer is based based off of open source project called OHI. The original version is. This one is OHIF version 26 months ago or so. They came out with actually less two months ago. Three months ago they came out with version 3 which replaced version 2, but not all. The functionality of two is in three yet, which is why we haven't switched it. So I started to kind of work on that. For the most part, the viewer is just the default that they have but with certain buttons removed. So I have removed those buttons from version 3 and.

That's what we had planned on doing, letting you choose which viewer is used from micro before you you launch an image

never got there. I actually have a Maestro branch which started that from a long time ago, but again I never completed it at a certain point. Could the Web viewer 3 replaced two? Yes, it just needs to be validated because it hasn't been validated yet. So part of the R/3.

Both are available currently in the dev environment. I did not create the team names in the upper environments yet, so while the code is deployed and seeing that running, you can't access it because there's no endpoint to be able to to go to it.

But at the end point is just for person two like just dot com or you know environment dash dot com, dash viewer dot CSS dot com and then for D three dot com or environment dot com viewer dash V, three dot com ICSS dot com. Now the most important part.

D3.com or environment.com viewer dash V3.com ICSS .com. Now the most important part of the viewer is.
We asked it is Chiada based more or less and the config files to point to the dot com whatsoever are as follows.
Nope, that's not the right one.
So for version two it is here and I'll put this in the chat so we don't get it.
And it always uses default JS, so this is where it points to for def. It's just what it uses for the authentication. So this points to the all

zero client and then there's also 15. I guess it's playing with it, but there's also a just like you know the portals and all the other portals. So this is a JavaScript component that we had to add in as well and this is the integration for it. Now a problem that occurs when merging is that this file it's the same name file for all of them. So basically you'll want to always when you merge from one branch to the next, you'll always want to skip this file.
Right, if you add something into the configuration that needs to be changed, then you're going to have to ignore the file and then do a just a simple pull directly for this one file for that environment. So let's say you did an upgrade and you forgot to ignore the file. Basically no images would ever come back because it's not going to be pointing to the correct instance

Just update this file, then obviously we rebuild it and then it would work.
For uh, the dot com you were three. It's a little bit slightly different, but similar idea. I don't know if it's cool.
This is very similar in terms of the setup, I just kind of added it a little bit differently.
So this is for the OITC, which you can see is slightly slightly different things here and then. The data source is figured a little bit down below, slightly different the endpoints are still seeing here and then there's hockey for it. Now this can this code can be runable. It is a little tricky.
You need to have yarn installed, but let's let's get us dry man.

And I have AI. Do have a web page or sorry conference page on how to build this. I will link it in the documentation as well. First it's. All.
This is going to take a while. Sorry while we're doing that.
With that one.
Nope, that's the homepage.
Yes, but I'll link to this page. I'll tell you how to build and then some of the changes that we did about the success to.
I'll be change that. I'll make sure these are up to date.
It's just going to take takes about 5 minutes or so, but the end

product is.
Where's my go to?
Then I just need to copy the ID.
No, I don't know if any of these studies are good, so we just give this a random try. Hopefully that works.
All right. Yeah, this is going to make you sign up. So 11 trick I do on this. So this sign in is the external or all zero only if you go directly to it. So if you have an external user, it's a great way to test, but normally you're already logged in whatever product like that, right. Let's say I was doing, yeah.

If you're logged into internally.
Go back to that, it will take the copy, so it will apply the settings there. So like here there's an error with the image, right? Error loading image. So that means it got the image but it didn't like what was returned. Like I said, I just picked one randomly. So that could be the case, right? So what I would do, I'll look here. This is where it has one here that means framework one. You can look here. This is the call it made against the dot com web server. So this is our study instance ID. This is our series instance ID. This is our instance ID and then it's pulling frame one

instance ID. And then it's pulling frame 1 and if there was a response, it doesn't really say one way or another again.
This is data that's probably very old. So yeah, that probably doesn't probably doesn't work sometimes, you know?
Way back when they would upload the same instance ID in here. That also screws up the dot com, whatever. They can't have two of the same instance.
Some more thing.
Try one more. If it doesn't work, we'll actually go into the data portal and load one.
Let's go.

Yeah.
Anybody know when you were studying that might have stopped with?
I just think these are good.
I can't remember if mine's endeavor in there, but you can try this one called SAM Study.
Right. I think I was in there.
I also might not have uploaded the right hierarchies for it to work

OK. Sorry. All right.
Cardiac stuff.
All right. Hopefully we'll be like here.
All right. Yeah.
It doesn't really matter, basically.
Each thing shown here is a series. Series two. There's two inches in that series. These are the standard.
Options allowed. What's not allowed is to download, right? So that would be something that would be removed if you want to see the dot com tags and you can see that here.
If we do.
I forget this is three.

I'll be the exact exact.
* But it's a little bit different. So instead of just having slash the ID, it's like it takes it as a parameter. Kind of like this. It might not be fewer, I forget exactly.
So we're starting to do, we'll start off the back on tour now, right now pointing to the web server.
What do you call it online?
EKF. If we had wanted to debug locally on both, then I wouldn't need to update the file to point to local host in order to do that.
One thing that's a little bit odd that I never figured out whether changed for the background web server is that I think it runs locally

over HTTP so HTTPS that could be changed. I just never got around to it. So when I passed my collection you'll see that this is very resourceful.
When it runs so it takes a while. Also when it builds in it takes a lot but we had to up the resources in order to do this. So this this is AA list of images that are pulled out of the database. In order to see that you have to have the show study list as true. So I think in dev is that true but like in production this would be false so you wouldn't be able to search for images how many images are in that instance. So we can just try to click and try it out. OK. So here's here's what it is. So it's viewer studied since Uids like I just missed

one.
S.
I have to. I don't know how to.
Check that.
Again, it's going to have some more things where if these images are older than it could have an issue. This is saying that the DICOM tag for patient age isn't set correctly in the image, so that's why it's not it's not showing up there.
Used to work, really, so load from my extra data that will load here, but because there's a lot of nonsense, sometimes it's harder to find out what one of those is.
So if you want to troubleshoot, would you like go down to your

local or no?
And then they'll give you basically just ask what is the study instance, UID. And then you can try the first things I do, try to go to a main account like that, if you haven't, probably the same behavior, exit itself and then I'll inspect the logs of the DICOM, whatever and see what it's for. About other times you'll need to, you know, just do F12.
Everybody's are having here, right? So there's some basically this return in all.
Access to AWS validation? Is that a problem like for this?
You can still get those things you don't need. You know I'm not doing anything with the AWS console. No like secrets need to be
obtained from.
But yeah, that's kind of how it how it goes with obviously the Canon will be issues, but a lot of times it is you know, a problem with the data type of thing, what will happen theoretically because it's Java basically 0 there's a vulnerability with the thing. So at that point it's best to go to the you know dot org and then oh, they have any release or something like that and then get down the latest code into the repo which will fix whatever vulnerability and I'll come back to USI have to do it like once a year or so.
How happens? So I'll just go to the.
I would say to somebody.
Yeah, so it's just this Github. Oh, you can see this is the. There's

I have added a lot recently. They tell you about the renting strategy, release strategy and stuff like that. What are the requirements YARN 117. So that's that's some other things that sometimes we'll have to do these things out of date and go into the trial formation. They they supply the Docker file for the, I just modified it a little bit.
So making sure that 0 it's using the right person. To note though, this is the version 2 viewer, so that was 16. Originally it was working so I had a great picture built that removed some of the vulnerabilities for.
Verse three. You know, this one's building off of sixteen by fifteen

Verse 3 you know this one's building off of 16 by 15. All this is over 3018 plus. So they did upgrade. So when you pull down the code then that's when you need to check for OK it's it's version 18. I can't just use the file that I have here. It needs to be modified a little bit just so that it builds quickly and then you'll see things like the yarn install yarn from build similar similar to the.
To get it through.
But yeah, that's pretty much it. Any any questions?
Yes, please. And one last question, what about the harmony where the people were working on previously

the people were working on previously? That is shelved right now. Nobody ever worked on, nobody had anything on it. There was there was talks, but nobody had done anything on it.
So it's probably, probably dead, especially with Jesse gone now